12-10-2018 2:00 PM

Tension Analysis in Survivor Interviews: A Computational Approach

Jumayel Islam
The University of Western Ontario

Supervisor
Mercer, Robert E.
The University of Western Ontario Co-Supervisor
Xiao, Lu
Syracuse University

Graduate Program in Computer Science
A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

Abstract

Tension is an emotional experience that can occur in different contexts. This phenomenon can originate from a conflict of interest or uneasiness during an interview. In some contexts, such experiences are associated with negative emotions such as fear or distress. People tend to adopt different hedging strategies in such situations to avoid criticism or evade questions.

In this thesis, we analyze several survivor interview transcripts to determine different characteristics that play crucial roles during tension situation. We discuss key components of tension experiences and propose a natural language processing model which can effectively combine these components to identify tension points in text-based oral history interviews. We validate the efficacy of our model and its components with experimentation on some standard datasets. The model provides a framework that can be used in future research on tension phenomena in oral history interviews.

Keywords: emotion recognition, hedge detection, oral history, interview transcripts, social discourse, reticence, tension

Acknowledgements

I would like to express my sincere gratitude to my thesis supervisors Dr. Robert E. Mercer and Dr. Lu Xiao for their continuous support and guidance throughout the entire time of my Masters study. I am really thankful to them for their patience, encouragement and always steering me in the right direction whenever they thought I needed it. Furthermore, I would like to thank Dr. Steven High for his useful comments, remarks and engagement through the learning process of this Masters thesis. I am gratefully indebted to his valuable comments on this thesis. I would also like to thank his group at the Centre for Oral History and Digital Storytelling for providing us with the transcribed and translated interviews.

I would like to thank the Department of Computer Science of Western University for its support towards my studies and providing me with scholarship. I would also like to thank Concordia University and Dr. Steven High for providing me with the financial support for my research.

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

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Oral history is the systematic collection of living people's testimony about their own experiences [55]. It is a strategy for conducting historical research through recorded interviews between a storyteller with individual experience of historically significant events and a well-informed questioner, with the objective of adding to the historical record [70]. It plays a significant role for historians to understand the experience shared by the people from their past and analyze it effectively. One of the crucial benefits of oral history interviews is that they can shed light on important issues that might not have been present in previous historical narratives.

Oral history interviews can be free form where the interviewer allows the narrator to speak freely with the recorder turned on. Narrators do not usually have to worry about being interrupted with questions from the interviewer in such a form of interview [91]. On the other hand, some interviews are well-structured. Such interviews have the interviewer asking a set of questions to the narrator in order to get as much information as possible. Often in such interviews, though the interviewers have a specific set of guidelines for asking questions, they might need to improvise in order to capture certain information at certain times through out the interview. Sometimes, the narrators don't feel able to speak freely on certain matters causing a conflict of interest, i.e., tension between the interviewer and the narrator [43]. Tension is the feeling that is produced in a situation when people are anxious and do not trust each other, and when there is a possibility of sudden violence or conflict [44]. It is a situation or condition of hostility, suspense, or uneasiness. In psychology, tension is defined as an emotion of physical and psychological strain joined by discomfort, unease, and pressure to look for alleviation via talking or acting [61]. Tension is also defined as a state of latent hostility or opposition between individuals or groups [4].

Oral history interviews involve complex social interactions and different factors highly influence the interview situation such as complexity of human lives, age, intelligence, personal quality, etc. [11]. Both the interviewer and the interviewee contribute to these components

and often it generates a situation which can be difficult for both parties, causing unpredictable emotional responses by the interviewee which changes the interview dynamics. Understanding these factors and the turning points in an interview is important for the interviewers to understand the process of interviewing and also for self-awareness [11]. Misztal [48] suggests that emotions lead directly to the past and bring the past somatically and vividly into the present. This motivates our interest in analyzing the emotional aspect of the interview, how it affects the internal dynamics and it's impact on causing tension between the narrator and the interviewer.

Layman (2009) [43] showed how reticence can also cause conversational shifts by interviewees which often limits responses on certain issues. It is a common strategy adopted by the narrators to avoid either outright refusal to respond or full disclosure. For example, the use of discourse markers such as *"not really"*, *"not that I remember"* or *"well, anyway"* in responses shows how reticence can be influential in an interview. This phenomenon indicates tension points in an interview and gives interviewers an idea that the conversational flow has somewhat been disrupted. Layman (2009) [43] also showed how certain topics can lead the narrators to employ such strategies in order to avoid answering certain questions. More often such responses are filled with reticence and are either short or dismissive. Subjects which address individual trauma, regardless of whether torment or dread or humiliation, are probably going to incite hesitant reactions from narrators. This leads to interviewers' judgement whether to press the interviewee when it becomes evident that the narrator is reluctant to speak on certain matters. Thus, it imposes the necessity of analyzing such phenomenon and building tools that can automate the process of identifying tension points in interviews giving the interviewers much more flexibility to understand and control the flow of the interview.

Ponterotto (2018) [66] showed how hedging is used in conversations to deal with controversial issues. Hedging refers to the technique of adding fuzziness to the propositional content by a speaker. It is a common hesitation strategy adopted by the narrators in oral history interviews. It gives narrators a moment to think and organize their thoughts to plan a successful response. For example, the usage of *"I think ..."*, *"Well, ..."* in interviews give narrators the authority to shape their narratives. Thus, it is an important element to explore in order to identify tension points in such interviews.

In this thesis, our final objective is to detect tension in oral history interviews such as survivor interview transcripts. To achieve this, we identify lexical variations in narratives, recognize emotions which may affect the interview dynamics, and analyze hedge strategies. We apply computational approaches to automate this process of identifying the tension phenomenon in survivor interviews.

The rest of the thesis is structured as follows: In addition to the discussion above, Chapter 2 reviews the related literature on detecting tension or similar phenomena in interview transcripts.

Literature related to emotion detection and hedging is reviewed in the appropriate chapters. Chapter 3 describes the overall architecture of our system. Chapter 3 also briefly describes different modules that have been used in this study. Chapter 4 describes our proposed model for emotion recognition and gives a comprehensive evaluation of the model on some standard datasets. Chapter 5 describes in detail different hedging strategies and our proposed approach for identifying hedging and discourse moves in survivor interviews. In Chapter 6, we provide our experimental results on two survivor interviews and give a thorough analysis of our system. We also discuss different aspects of our architecture, its advantages and its shortcomings in this chapter. Lastly, in Chapter 7, we give a summary of the research work that has been done for this thesis. We also give direction for further work that can be done in this field.

Chapter 2

Literature Review

This chapter reviews some of the studies that have been done in recent years to analyze interview dynamics and how different factors affect the interview flow, causing tension between the interviewer and the interviewee. We also discuss some of the computational approaches that have been developed to identify such a phenomenon. We provide detailed analysis of past literature concerned with the various components of our architecture in the later chapters: emotion detection in Chapter 4 and hedging and discourse markers in Chapter 5.

One of the important factors in interviews that shows signs of tension between the interviewer and interviewee is hedging. Ponterotto (2018) [66] discussed different hedging strategies that have been employed by Barack Obama, the former president of the United States, in political interviews affecting interview dynamics by changing the flow of conversation. They presented an in-depth analysis of the president's responses by identifying hedging-related discursive strategies. In their work, they proposed four crucial discourse moves. "Reformulating the interviewer's question"; "Expanding the scope of the original question sequence"; "Switching the time frame of the question context" and "Recasting the question reference from specific to general terms". They also discussed hesitation strategies, such as, pauses and repairs *(yes, no)*, restarts *(I won't ... I won't say)* and discourse markers *(anyhow, anyway, I mean)*.

Pedro Martín (2013) [14] discussed four common hedging strategies: Indetermination, Camouflage, Subjectivization and Depersonalization. We provide brief details about these strategies motivated by the description found in [5]. Strategy of Indetermination includes the usage of various epistemic modality, for example, epistemic verbs *(assume, suspect, think)*, epistemic adverbs *(presumably, probably, possibly)*, epistemic adjectives *(apparent, unsure, probably)*, modal verbs *(might, could)* and approximators *(usually, generally)*. Usage of such epistemic modality provides vagueness and uncertainty in the interviewee's response. Strategy of Camouflage includes the use of different adverbs, e.g., *generally speaking, actually*. This method acts as a lexical device to avoid a negative reaction by the interviewer. Strategy of

Subjectivization is activated by the usage of first person pronouns followed by verbs of cognition, for example, *"I think", "I feel"*. These expressions have been given the term "Shield" in [67]. This technique allows the interviewees to express their opinion freely in certain events causing the interviewers to oblige and listen. Strategy of Depersonalization includes the use of impersonal pronouns or constructs, for example, *"we", "you", "people"*. This allows the interviewees to hide themselves behind a non-identified subject.

Though interview dynamics have been studied thoroughly in the past [11, 43, 48, 66, 91], there are very little work that have been done to apply computational approaches to automate the process of detecting tension in interviews. Burnap et al. (2015) [13] performed conversational analysis and used different text mining rules to identify spikes in tension in social media. They showed how lexicons of abusive or expletive terms can identify high levels of tension separated from low levels. Their proposed tension detection engine relies solely on the lexicons and membership categorization analysis (MCA) [72]. They showed that their proposed model consistently outperforms several machine learning approaches and sentiment analysis tools.

Buechel et al. (2018) [12] provided the first publicly available dataset for text-based distress and empathy prediction. Distress is a negative affective state that people feel when they are upset about something. Distress is closely related to tension, the main focus of this thesis. Buechel et al. (2018) [12] considered the problem of distress and empathy prediction as a regression problem. They used a Feed-forward Neural Network with word embeddings from Fast-Text as their inputs and a Convolutional Neural Network model with one convolutional layer with three different filter sizes. They argue that CNN models are able to capture semantic effects from the word order. They showed that CNN performs particularly well in detecting distress compared with detecting empathy from text.

In this chapter, we discussed different factors that affect the interview dynamics. We also gave brief details about the studies that have been done to identify tension-like phenomena in text. We provide a detailed literature review of the different components of our architecture in their respective chapters. There has been very little work done in detecting tension from interview collections. In this thesis, we introduce a novel computational approach to detect tension in survivor interviews.

Chapter 3

Overall Architecture

This chapter gives an overview of the architecture of our overall system. We briefly present the components of our framework. We also discuss our proposed algorithms for tackling the problem at hand. More detailed discussions of the main components are provided in Chapters 4 and 5.

3.1 Overview

The two core components of our proposed architecture for detecting tension in interview transcripts are: the Emotion Recognition Module and the Hedge Detection Module. We give a brief overview of these components in this chapter and we give a detailed analysis of these components in their respective chapters. We also talk about other important features (booster words, markers, etc.) which we found to be useful during our analysis in this chapter. We provide pseudo-code for our tension detection algorithm combining all these components.

3.2 Emotion Recognition

The first component in our system is the emotion recognition module. As we discussed earlier, emotion plays a vital role in determining tension situations in survivor interviews. The following excerpt from an oral history interview[1] shows how emotion can dominate a tension situation.

> **Interviewer:** Even if they do - but the Hutus can pray in their Hutu churches and the Tutsis in the Tutsi churches.

[1] In all transcript excerpts, the questions by the interviewer will be indicated by "Interviewer", and the interviewee's response by "Narrator". The interview transcripts can be found at `http://livingarchivesvivantes.org/`.

> **Narrator:** Exactly. You see, it's odd, no, I'm not going to waste my time praying in these circumstances because it's completely - it's hogwash.
> **Interviewer:** Tell me -
> **Narrator:** But what is even more serious is that there are Canadians, especially Quebecers, who stand behind the factions and are even more extremist than we are!
> **Interviewer:** Indeed.
> **Narrator:** It's strange!

As we can see by the responses, the narrator felt *anger* at that moment of time, thus creating a tension situation between the interviewer and narrator. Tension can also arise at the very beginning of an interview but there might not be any visible indicators of it until some later point. Sometimes the interviewers keep pushing which makes the interviewee really uncomfortable changing the dynamics of the situation. As a result two opposing forces arise in such situation which are sometimes so strong that we can detect it, which, we refer to as tension. Interviewees usually show different emotions as discomfort in such scenarios. Jurek et al. (2015) [36] also showed how negative sentiment or emotion can lead to tension.

In our studies, we have found negative emotions such as *anger, fear, disgust, sadness* to be particularly important in tension situations, thus, we incorporate these emotional categories in our algorithm. We provide pseudo-code for emotion recognition in Algorithm 1. We provide a detailed analysis of this emotion recognition component in Chapter 4 where we discuss our proposed model and do a comprehensive performance evaluation of the model.

3.3 Hedge Detection

The second component of our proposed architecture is the hedge detection module. We have discussed previously how interviewees employ different hedging strategies in tension situations to change the flow of conversation or to show their reluctance in responding to certain topics. The following excerpt from an interview shows a scenario like this:

> **Interviewer:** Do you think that one day you may want to meet them?
> **Narrator:** It's a good question. I don't know. We think we'd meet them, but meeting someone who says "No, I didn't do that," yet we know they have, it's also - in any case, I am reluctant. I am not so sure.

In the above example, as we can see, the use of phrases like *"I don't know"*, *"We think"* shows how hedging contributes to the conversational dynamics, building a tension like environment

among the interviewer and the interviewee. Hedging is presented in more detail in Chapter 5 where we also discuss our proposed approach for detecting hedges in interview transcripts, along with a performance analysis of our approach.

3.4 Tension Detection

Our tension detection algorithm is composed of the Emotion Recognition Module and the Hedge Detection Module. In addition, we have additional features which we have found to be useful in our studies. We provide details about these features below.

3.4.1 Booster Words

Boosting, using terms such as *absolutely*, *clearly* and *obviously*, is a communicative strategy for expressing a firm commitment to statements. Holmes (1984) [32] provides an early definition of boosting. According to him, "Boosting involves expressing degrees of commitment or seriousness of intention". It allows speakers to express their proposition with confidence and shows their level of commitment to statements. It also restricts the negotiating space available to the hearer.

Boosting plays a vital role in creating conversational solidarity [32] and in constructing an authoritative persona in interviews [31]. The following example shows a use of *"completely"* as a booster word during an interview.

> **Interviewer:** Before returning to I don't know where after that?
> **Narrator:** Do you know what happened to me afterwards? Oblivion, amnesia. I can't tell you how long we stayed in Kabuga, I can't tell you when we left for Kabuga and on what date we left Kabuga - no. When I think about it, it's like a single day that is being drawn out. I **completely** lost the notion of time. That may have helped me survive, but my capacity to remember was blocked.

In this thesis, we have compiled a lexicon of booster words from different online resources. This lexicon can be found in Appendix A. Interestingly, if booster words are preceded by negated words such as *"not", "without"*, it can act as hedging. For example, "I'm still **not sure** if I would go back, I don't know what it would be like." Here, *"sure"* is a booster word. However, since it is preceded by a negation word *"not"*, it changes the meaning completely. We handle this kind of situation in our proposed algorithm.

3.4.2 Markers

The use of markers (e.g., laugh, silence, sigh) in such interviews is noteworthy. These have various functions. Sometimes markers like "laughter" indicate invitations to the interviewer to ask the next question. At other times it represents hesitation or nervous deflection. In our thesis, we have compiled a lexicon (Appendix D) of such markers/cues and used them in our tension detection algorithm. The example below shows a use of the marker "laugh":

> **Interviewer:** What would you like Canadians to know about you?
> **Narrator:** I don't know [laughs]. ... It's a question of mutual acceptance. Sometimes we can be suspicious, thinking that Canadians haven't lived through such major events, but ... they also have things to share ... I think it's a question of ... opening up to the experience of others...

3.4.3 Asking a Question Back

When the interviewee poses a question back, asking for clarification, that is also a sign that interviewee is trying to negotiate, a good marker for identifying tension points. In our tension detection algorithm, we use this as a potential criterion. The example below demonstrates a situation like this:

> **Interviewer:** So going back now to the period of 1994, during the genocide - you saw it coming, but how did you live through that time?
> **Narrator:** How do you mean?

3.4.4 Outliers

Interview dynamics provide examples of a narrator giving unusually long or short answers for a specific question type. During our lab discussions we felt that this type of dynamic could be a sign of tension, so we added this as one of the criteria in our tension detection algorithm. We find statistics (mean and standard deviation) based on question type over the whole interview transcript which allows us to determine if a response by the interviewer is unusually long or short in comparison with their other responses for the same question type. We consider *wh-question*, *how*, *yes-no* and *mixed* (mix of other question types) as the prime question types. We find the mean (Equation 3.1) and standard deviation (Equation 3.2) for each type.

$$\mu_{q_t} \leftarrow \frac{1}{N(q_t)} \sum_{i=1}^{N(q_t)} \{w_i(q_t)\} \tag{3.1}$$

$$\sigma_{q_t} \leftarrow \sqrt{\frac{\sum_{i=1}^{N(q_t)}(w_i(q_t) - \mu(q_t))^2}{N(q_t) - 1}} \tag{3.2}$$

Here, μ_{q_t} indicates the mean for the question type q_t, σ_{q_t} indicates the standard deviation for the question type q_t, $w_i(q_t)$ indicates the total number of words in excerpt e_i belonging to q_t and $\mathtt{N}(q_t)$ indicates the total number of excerpts belonging to each q_t. We consider a response to be an outlier, thus a possible point for tension, if it falls below $3\sigma_{q_t}$ or is above $3\sigma_{q_t}$.

3.4.5 Proposed Algorithm

Here, we provide the pseudo-code for tension detection and its components:

Algorithm 1 Emotion Detection algorithm

```
function isNegativeEmotion(s)
    Predict emotion using MC-CNN model
    if Predicted Emotion ∈ {anger, fear, disgust, sadness} then
        return True
    else
        return False
    end if
end function
```

Algorithm 2 Tension Detection algorithm

```
function TensionDetection()
    Excerpts(E) ← List of narrator's responses

    Markers (M) ← List of evasions markers and cues
    Single Excerpt(e) ← List of sentences in each response

    q_t ← Question type (wh-question/how/yes-no/mixed)

    w_i(q_t) ← Total number of words in excerpt e_i belonging to q_t

    N(q_t) ← Total number of excerpts belonging to each q_t

    Mean, μ_{q_t} ← 1/N(q_t) Σ_{i=1}^{N(q_t)} {w_i(q_t)}

    Standard Deviation, σ_{q_t} ← sqrt( Σ_{i=1}^{N(q_t)} (w_i(q_t) − μ(q_t))^2 / (N(q_t) − 1) )

    for each excerpt e in E do
        w ← Total number of words in excerpt e
```

```
        q ← Question asked by the Interviewer
        nSentences ← First n sentences in e
        isNegativeEmotion, e_neg = False
        isHedgedSentence, h_s = False
        isBoosting, b_s = False
        markerPresent, m_p = False
        isQuestion, q_s = False
        isOutlier, o_r = False
        for each sentence s in nSentences do
            if isNegativeEmotion(s) is True then
                e_neg = True
            end if
            if isHedgedSentence(s) is True then
                h_s = True
            end if
            if isBoosting(s) is True then
                b_s = True
            end if
        end for
        for A in M do
            if A in e then
                m_p = True
            end if
        end for
        if nSentences[0] is a Question then
            q_s = True
        end if
        if w > μ_{q_t} + 3 * σ_{q_t} or w < μ_{q_t} − 3 * σ_{q_t} then
            o_r = True
        end if
        if (e_neg and h_s) or (h_s and b_s) or (h_s and m_p) or q_s or o_r then
            mark excerpt as Tension
        else
            mark excerpt as No Tension
        end if
    end for
end function
function isBoosting(s)
    Boosters(B) ← List of booster words
    for b in B do
        if b in B and b is not preceded by not or without then
            return True
        end if
    end for
    return False
end function
```

3.5 Summary

In this chapter, we have provided an overview of our tension detection architecture. We have also briefly discussed some of its core components. Emotion plays a significant part in the tension situation in an interview. Hedging or reticence largely contributes to it, as well. Hedges and boosters draw attention to the fact that statements don't just communicate ideas, but also the speaker's attitude to them [30]. Lastly, we have provided our tension detection algorithm along with pseudo-code for the core components of our architecture. Details of these components is provided in the next chapters.

Chapter 4

Multi-channel CNN Model for Emotion and Sentiment Recognition

This chapter discusses about one of the core components of our tension detection architecture, namely emotion recognition. We provide brief details of the recent works that have been done in this field of research. We also discuss in details about our proposed model for recognizing emotion from text. We give an in-depth analysis of our model by evaluating performance on some standard datasets.

4.1 Introduction

Emotion plays a significant role in oral history interviews. Misztal [48] suggests that emotions lead directly to the past and bring the past somatically and vividly into the present causing a shift in interview dynamics. Jurek et al. (2015) [36] also showed how negative sentiment or emotion can lead to tension. Emotion recognition in computational linguistics is the process of identifying discrete emotion expressed by humans in text. In the past decade, emotion detection in text has become widely popular due to its vast applications in marketing, psychology, political science, etc. The evolution of different social media sites and blogs has paved the way for researchers to analyze huge volume of opinionated text. As a result, it has caught attention of lot of researchers of related fields. Emotion analysis can be viewed as a natural evolution of sentiment analysis. While sentiment analysis deals with polarity of texts (positive, negative or neutral) and the intensity of it, emotion mining deals with identifying human emotion expressed via text. However, this field still has a long way to go before matching the success and ubiquity of sentiment analysis. Identifying discrete human emotions can be useful in lots of applications such as analyzing interview dynamics, political campaigns, etc.

There is often a misconception about sentiments and emotions as these subjectivity terms have been used interchangeably [56]. Munezero et al. (2014) [56] differentiate these two terms along with other subjectivity terms and provide the computational linguistics community with clear concepts for effective analysis of text. While sentiment classification tasks deal with the polarity of a given text (whether a piece of text expresses positive, negative or neutral sentiment) and the intensity of the sentiment expressed, emotion mining tasks naturally deal with human emotions which in some end purposes are more desirable [69][19][54].

There are four approaches that have been widely used in emotion detection studies. They are keyword based, learning based, hybrid based and rule-based approaches. Keyword based approach usually depends on some sort of emotion lexicons. This approach is fairly easy to implement as it depends on identifying emotion keywords in text mostly. But this approach has some major limitations because of its complete reliance on emotion lexicons. For example, "I passed the test" and "Hooray! I passed the test", both should imply the same emotion "happiness", but keyword based approaches might fail to predict the emotion for the first sentence as it lacks emotion keyword in it. Moreover, keywords can be multiple and vague and they can possess different meaning according to its usage and context. Learning based approaches make use of trained model on large annotated datasets. Such models, in general, use emotion keywords as features. These approaches include traditional machine learning and deep learning based techniques. One of the main advantages of such techniques is that they can adapt to domain changes very easily by learning new features from the given training set. Although learning-based approach can automatically determine the probabilities between features and emotions, it still has limitations as it also depends on keywords as features to certain extent. Hybrid approaches consist of a combination of keyword-based implementation and learning-based implementation. It is relatively more popular than keyword-based approach or learning-based approach alone as it can achieve higher accuracy from training a combination of classifiers and adding knowledge-rich linguistic information from dictionaries and thesauri [6][7]. Rule based approaches consist of a predefined set of rules or ontologies for the purpose of detecting emotion. However, success of this approach can be highly domain dependent and requirement of rules or ontologies can be expensive.

There are a number of emotion theories available which suggest different sets of basic emotions. Interestingly, *joy, sadness, anger, fear* and *surprise* are common to all of the models. To the best of our knowledge, the model suggested by Ekman (1999) [22] is the most broadly used emotion model. In this study, we use Ekman's basic emotions together with other sets of emotions [65][78].

Deep learning models have been very successful in recent years when applied on text-related tasks. However, such models require large amount of data on which it can be trained

on. But in emotion detection related studies, the datasets that are available are very small. As a result, we focused on social media because of the fact that it generates huge volume of text data every moment and the data collection process is also very simple and straight-forward. The huge number of collected text can be very beneficial for deep learning models. In this thesis, we put our efforts on building such a model which can be effectively used in identifying emotions in interview transcripts.

4.2 Related Work

The advent of micro-blogging sites has paved the way for researchers to collect and analyze huge volumes of data in recent years. Twitter, being one of the leading social networking sites worldwide [57], allows its users to express their states of mind via short messages which are called tweets. Detecting emotion and sentiment from noisy twitter data is really challenging due to its nature. Tweets tend to be short in length and have a diverse vocabulary making them harder to analyze due to the limited contextual information they contain. We are interested in tackling these two tasks with a novel use of a single neural network architecture.

In early textual emotion mining and sentiment analysis research, the usefulness of using external lexicons along with predefined rules has been demonstrated. Aman and Szpakowicz (2007) [6] introduced a task for annotating sentences with different emotion categories and its intensities. They showed the usefulness of using lexical resources to identify emotion-related words using two machine learning algorithms - Naive Bayes and Support Vector Machine (SVM). However, lexical coverage of these resources may be limited, given the informal nature of online discourse. A sentence may not have any emotion-bearing word at all. They [7] also used *Roget's Thesaurus* along with *WordNet-Affect* for fine-grained emotion prediction from blog data. They utilized two different types of features - corpus based features and features derived from two emotion lexicons. They have reported superior performance with their experiments when combining both of the lexicons with corpus-based unigram features on data collected from blogs. Agrawal and An (2012) [3] proposed a novel unsupervised context-based approach for detecting emotion from text. Their proposed method computes an emotion vector for each potential affect-bearing word based on the semantic relatedness between words and various emotion concepts. The results of evaluations show that their technique yields more accurate results than other recent unsupervised approaches and comparable results to those of some supervised methods. One of the weaknesses of their approach is that the semantic relatedness scores depend on the text corpus from which they are derived. Neviarouskaya et al. (2007) [59] proposed a rule-based system called "Affect Analysis Model" which can handle informal texts in particular. They built a database of abbreviations, emoticons, affect

words, etc., in which each entry is labeled with an emotion and its intensity. Bandhakavi et al. (2017) [9] proposed a unigram mixture model (UMM) to create a domain-specific lexicon which performs better in extracting features than Point-wise Mutual Information and supervised Latent Dirichlet Allocation methods. Thelwall et al. (2010) [90] proposed an algorithm, *SentiStrength*, which utilizes a dictionary of sentiment words associated with strength measures to deal with short informal texts from social media. Gilbert and Eric (2014) [27] proposed *VADER*, a rule-based model for sentiment analysis. They built a lexicon which is specially attuned to microblog-like contexts and their model outperforms individual human raters.

More recently, deep learning models have proven to be very successful when applied on various text-related tasks. Kim (2014) [38] showed the effectiveness of a simple CNN model that leverages pre-trained word vectors for a sentence classification task. Kalchbrenner et al. (2014) [37] proposed a dynamic CNN model which utilizes a dynamic k-max pooling mechanism. Their model is able to generate a feature graph which captures a variety of word relations. They showed the efficacy of their model by achieving high performances on binary and multi-class sentiment classification tasks without any feature engineering. dos Santos et al. (2014) [20] proposed a deep CNN model that utilizes both character and word-level information allowing them to achieve state-of-the-art performance on both binary and fine-grained multi-class sentiment classification for one of the twitter datasets. Tai et al. (2015) [84] proposed a Tree-LSTM model which can capture syntactic properties in text. Their model performs particularly well on the sentiment classification task. Wang et al. (2016) [95] proposed a regional CNN-LSTM model for dimensional sentiment analysis. Their proposed model computes *valence-arousal* ratings from texts and outperforms several regression-based methods. Felbo et al. (2017) [24] proposed a bi-directional LSTM model with attention and show that their model can learn better representations when distant supervision is expanded to a set of noisy labels. Abdul-Mageed and Ungar (2017) [1] also used distant supervision to build a large twitter dataset and proposed a Gated Recurrent Neural Network model for fine-grained emotion detection.

4.3 Proposed Model

We represent the architecture of our model in Fig. 4.1. In this thesis, we discuss a novel use of a multi-channel Convolutional Neural Network model. A Convolutional Neural Network (CNN) is a type of artificial neural network used primarily in image recognition and processing, but in the last few years it has been widely used in natural language processing (NLP) tasks as well. Due to its huge success in related NLP tasks, we also make use of a CNN model in our study. Our proposed model consists of an embedding layer with two channels, a convolution

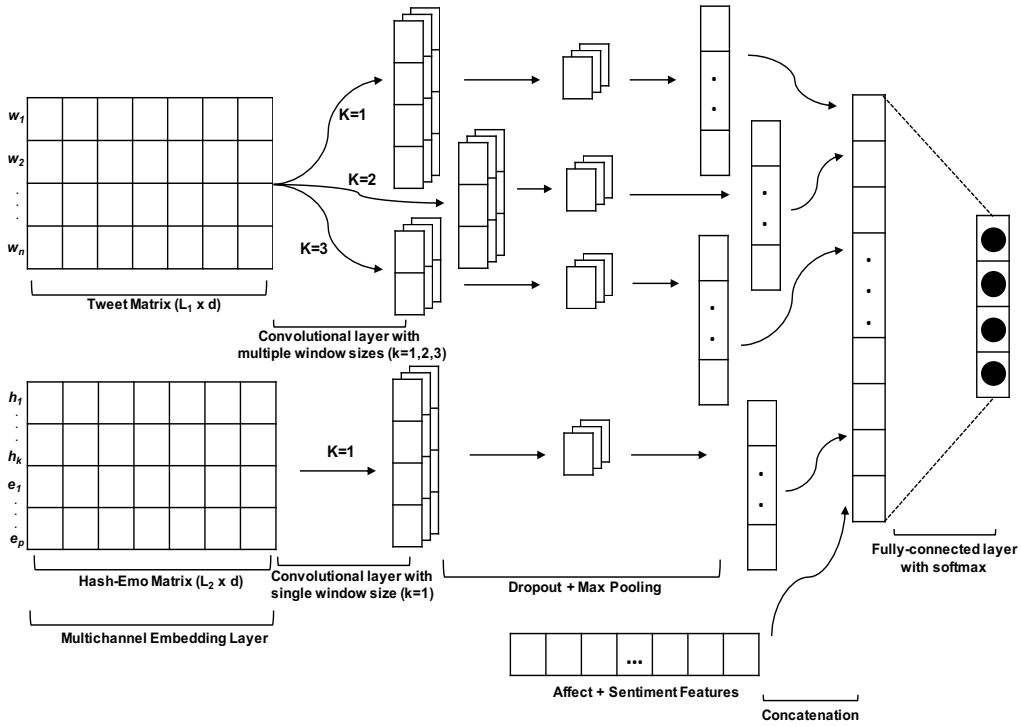

Figure 4.1: Overview of the MC-CNN model

layer with different kernel sizes and multiple filters, a dropout layer for regularization, a max pooling layer, multiple hidden layers and a softmax layer. In the following subsections, we describe each of these layers in detail.

4.3.1 Embedding Layer

In this layer, we have two embedding matrices, called the Tweet Matrix and the Hash-Emo Matrix, passed through two different channels of our convolutional neural network. The first matrix represents a particular tweet. Each tweet t_i consists of a sequence of tokens $w_1, w_2, \ldots, w_{n_i}$. A full description of what tokens are is given in Section 4.4.2.1. L_1 is the maximum tweet length. The height of the Tweet Matrix is L_1. Short tweets are padded using zero padding.

In the Tweet Matrix, every word is represented as a d-dimensional word vector. Since tweets are usually noisy, short in length, and have different kinds of features other than text, it's useful to have a word embedding specially trained on a large amount of Tweet data [87]. Previous research [16][80] has shown the usefulness of using pre-trained word vectors to improve the performance of various models. As a result, in our experiments, we have used the publicly available pre-trained `GloVe`[1] word vectors for Twitter by Pennington et al. [63]. `GloVe` is an

[1]https://nlp.stanford.edu/projects/glove/

unsupervised learning algorithm for obtaining vector representations for words, which is called a word embedding. The word vectors are trained on 27 billion word tokens in an unsupervised manner. A word embedding such as this is capable of capturing the context of a word in a sentence, semantic and syntactic similarity, relationships with other words, etc.

In this layer, we also pass another matrix called the Hash-Emo Matrix through a different channel in our network. This matrix is composed of three different sets of features: hashtags, emoticons and emojis. These are considered as distinguishable traits to showcase one's mood [102]. People like to use hashtags to express their emotional state through various micro-blogging sites (e.g., Twitter) [68]. Also graphical emoticons or emojis can convey strong emotion or sentiment. So for each tweet t_i, we extract hashtags $h_1, h_2, \ldots, h_{k_i}$ and emoticons/emojis $e_1, e_2, \ldots, e_{p_i}$. We concatenate the hashtags and emoticons/emojis vectors for each tweet t_i to get the Hash-Emo Matrix. We introduce a hyper-parameter L_2 as a threshold on the height of the Hash-Emo Matrix. Tweets with the number of hash-emo features less than L_2 are padded with zero while tweets with more hash-emo features than L_2 are truncated. We use word vectors from `GloVe` with dimension d for hashtags words. In the case that no word vector is found for a particular word we randomly initialize it. We also do random initialization of word vectors for emoticons. For emojis, we first map it to something descriptive (to be discussed in more detail in Section 4.4.2) and then generate random word vectors. These word vectors are tuned during the training phase.

4.3.2 Convolutional Layer

In this layer, we apply m filters of varying window sizes (k) over the Tweet Matrix from the embedding layer as seen in Fig. 4.1. Here, window size (k) refers to the number of adjacent word vectors in the Tweet Matrix that are filtered together (when $k > 1$). Then we slide our filter down and do the same for the rest of the word vectors. Let $w_i \in \mathbb{R}^d$ be the d-dimensional word vector corresponding to the i-th word in a tweet. Also let $w_{i:i+j}$ denote the concatenation of word vectors $w_i, w_{i+1}, \ldots, w_{i+j}$ and $F \in \mathbb{R}^{k \times d}$ denote the filter matrix. Thus a feature f_i is generated by:

$$f_i = F \otimes w_{i:i+k-1} + b \tag{4.1}$$

where b is a bias term and $\otimes$ represents the convolution action (a sum over element-wise multiplications). At this stage, we apply a nonlinear activation function such as *ReLU* [58] before passing it through the dropout layer. We use multiple filters with the same window size in order to learn complementary features from the same window. Different window sizes (k) allow us to extract active local k-gram features.

For the Hash-Emo Matrix, we apply m filters to each hash-emo vector to generate local unigram features in different scales before passing it to the next layer.

4.3.3 Pooling Layer

In this layer, we employ a max-over pooling operation [16] on the output from the previous layer for each channel in order to extract the most salient features. In this way, for each filter, we get the maximum value. So we get features equal to the number of filters in this stage. We chose max pooling instead of other pooling schemes because [101] showed that max pooling consistently performs better than other pooling strategies for various sentence classification tasks.

4.3.4 Hidden Layers

We concatenate all the feature vectors from the previous layer. In addition, we concatenate additional sentiment and affect feature vectors (which are described in detail in Section 4.4.2) as well which forms a large feature vector. This is then passed through a number of hidden layers. A non-linear activation function (i.e., ReLU [58]) is applied in each layer before the vector is finally passed through the output layer. We tried a different activation function (tanh) as well, but ReLU worked the best for us.

4.3.5 Output Layer

This is a fully connected layer which maps the inputs to a number of outputs corresponding to the number of classes we have. For multi-class classification task, we use softmax as the activation function and categorical cross-entropy as the loss function. The output of the softmax function is equivalent to a categorical probability distribution which generally indicates the probability that any of the classes are true. Mathematically, the softmax function is shown below:

$$P(y = j|z) = \frac{e^{z^T w_j}}{\sum_{k=1}^{K} e^{z^T w_k}} \tag{4.2}$$

where z is a vector of the inputs to the output layer and K represents the number of classes. For binary classification task, we use sigmoid as the activation function and binary cross-entropy as our loss function. Finally, the classification result can be obtained by:

$$\hat{y} = \arg\max_{j} P(y = j|z) \tag{4.3}$$

Emotion	Dataset			
	BTD	TEC	CBET	SE
joy	409,983	8,240	10,691	3,011
sadness	351,963	3,830	8,623	2,905
anger	311,851	1,555	9,023	3,091
love	175,077	–	9,398	–
thankfulness	80,291	–	8,544	–
fear	76,580	2,816	9,021	3,627
surprise	14,141	3,849	8,552	–
guilt	–	–	8,540	–
disgust	–	761	8,545	–
Total	1,419,886	21,051	80,937	12,634

Table 4.1: Basic statistics of the emotion datasets.

Dataset	#Tweets	#Positive	#Negative	#Neutral
STS-Gold	2,034	632	1,402	–
STS-Test	498	182	177	139
SS-Twitter	4,242	1,252	1,037	1,953

Table 4.2: Basic statistics of the sentiment datasets.

4.4 Experiments

In this section, we describe in detail the datasets and experimental procedures used in our study.

4.4.1 Datasets

We used a number of emotion and sentiment datasets for our experiments. A description of each dataset is given below:

BTD. Big Twitter Data is an emotion-labeled Twitter dataset provided by Wang et al. (2012) [96]. The dataset had been automatically annotated based on the seven emotion category seed words [78] being a hashtag and the quality of the data was verified by two annotators as described in [96]. We were only able to retrieve a portion of the original dataset as many tweets were either removed or not available at the time we fetched the data using the Twitter API. We applied the heuristics from [96] to remove any hashtags from the tweets which belong to the list of emotion seed words.

TEC. Twitter Emotion Corpus has been published by Saif (2012) [51] for research purposes. About 21,000 tweets were collected based on hashtags corresponding to Ekman's [22] six basic emotions. The dataset has been used in related research works [8][52][77].

CBET. The Cleaned Balanced Emotional Tweet dataset is provided by Shahraki et al. (2017) [77]. To the best of our knowledge, this is one of the largest publically available balanced datasets for twitter emotion detection research. The dataset contains 80,937 tweets with nine emotion categories including Ekman's six basic emotions.

SE. The SemEval-2018 Task 1 - Affect dataset was provided by Mohammad et al. (2018) [50]. The SemEval task was to estimate the intensity of a given tweet and its corresponding emotion. However, in this study, we utilize the labeled dataset only to classify the tweets into four emotion categories. We have used the training, development and test sets provided in this dataset in our experiments.

STS-Gold. This dataset was constructed by Saif et al. (2013) [73] for Twitter sentiment analysis. The dataset contains a total of 2,034 tweets labeled (positive/negative) by three annotators. This dataset has been extensively used in several works for model evaluation [40][74][75].

STS. The Stanford Twitter Sentiment dataset was introduced by Go et al. (2009) [28]. It consists of a training set and a test set. The training set contains around 1.6 million tweets, whereas the test set contains 498 tweets. The training set was built automatically based on several emoticons as potential identifiers of sentiment. However, the test set was manually annotated and heavily used for model evaluation in related research [34, 20, 28]. We perform two experiments on the dataset. One with all three labels (positive/negative/neutral) to compare the performance of different variants of our model and the other one with two labels (positive/negative) to make comparison with related works [34][20][28].

SS-Twitter. The Sentiment Strength Twitter dataset has been constructed by Thelwall et al. (2012) [89] to evaluate SentiStrength. The tweets were manually labeled by multiple persons. Each tweet is assigned a number between 1 and 5 for both positive and negative sentiments. Here, 1 represents weak sentiment strength and 5 represents strong sentiment strength. We followed the heuristics used by [73] to obtain a single sentiment label for each tweet, giving us a total of $4,242$ positive, negative and neutral tweets. The transformed dataset has been used in existing literature [73][100][28].

We provide basic statistics of the emotion labeled and sentiment labeled datasets used in our experiments in Table 4.1 and Table 4.2.

4.4.2 Experimental Setup

4.4.2.1 Data Cleaning

Twitter data is unstructured and highly informal [99] and thus it requires a great deal of effort to make it suitable for any model. NLTK [10] provides a regular-expression based tokenizer for

Twitter, TweetTokenizer, which preserves user mentions, hashtags, urls, emoticons and emojis in particular. Tokenization is the process of splitting a sequence of strings into elements such as words, keywords, punctuation marks, symbols and other elements called tokens. TweetTokenizer also reduces the length of repeated characters to three (i.e. "Haaaaaapy" will become "Haaapy"). In our experiments with Twitter data, we utilized the TweetTokenizer to tokenize tweets.

To accommodate the pretrained word vectors from [64], we pre-processed each tweet in a number of ways. We lowercased all the letters in the tweet. User mentions have been replaced with <user> token (i.e. @username1 will become <user>). In addition, we also removed urls from the tweets as urls do not provide any emotional value. We also normalized certain negative words (e.g., "won't" will become "will not"). Using slang words is a very common practice in social media. We compiled a list of the most common slang words[2] and replaced all of the occurrences with their full form (e.g., "nvm" will become "never mind"). Usage of certain punctuation is often crucial in social media posts as it helps the user to emphasize certain things. We found that two punctuation symbols (! and ?) are common among social media users to express certain emotional states. We kept these symbols in our text and normalized the repetitions (e.g., "!!!" will become "! <repeat>")

The use of emojis and emoticons has increased significantly with the advent of various social media sites. Emoticons (e.g., :-D) are essentially a combination of punctuation marks, letters and numbers used to create pictorial icons which generally display an emotion or sentiment. On the other hand, emojis are pictographs of faces, objects and symbols. The primary purpose of using emojis and emoticons is to convey certain emotions and sentiments [21]. One advantage of using the TweetTokenizer is that it gives us emoticons and emojis as tokens. Though we use the emoticons as is in our experiment, we utilize a python library called "emoji" to get descriptive details about the pictorial image.

In our experiments, we removed stop-words from the tweets and replaced numbers occurring in the tweets with the token <number>. We also stripped off "#" symbols from all the hashtags within the tweets (e.g., "#depressed" will become "depressed"). We only kept tokens with more than one character.

4.4.2.2 Input Features

Along with word embeddings, we used additional affect and sentiment features in our network. In our experiments, we used a feature vector V_f where each value in the vector corresponds to a particular lexical feature ranging between $[0, 1]$. We utilized a number of publicly available

[2]https://slangit.com/terms/social_media

lexicons which are described briefly below to construct the vector.

Warriner et al. (2013) [97] provides a lexicon consisting of 14 thousand English lemmas with valence, arousal and dominance scores. Three components of emotion are scored for each word between 1 and 9 in this lexicon. We calculate the average score for each component across all tokens in a tweet and normalize them in the range [0, 1]. Gibert (2014) [27] provides a list of lexical features along with their associated sentiment intensity measures. We utilize this lexicon to calculate the average of positive, negative, and neutral scores over all the tokens in a tweet.

In addition, we used the *NRC Emotion Lexicon* provided by Mohammad and Turney (2013) [53] which consists of a list of unigrams and their association with one of the emotion categories (anger, anticipation, disgust, fear, joy, sadness, surprise, trust). We use the percentage of tokens belonging to each emotion category as features. We also used the *NRC Affect Intensity Lexicon* provided by Mohammad and Bravo-Marquez (2017) [49] and *NRC Hashtag Emotion Lexicon* provided by Mohammad and Kiritchenko (2015) [52] which contain real-valued fine-grained word-emotion association scores for words and hashtag words respectively.

We combined two lexicons *MPQA* and *BingLiu* provided by Wilson et al. (2005) [98] and Hu and Liu (2004) [33], respectively, and used them to calculate the percentage of positive and negative tokens belonging to each tweet.

We also used *AFINN* [60] which contains a list of English words rated for valence with an integer between −5 (negative) to +5 (positive). We first normalized the scores in the range [0,1] and then calculated the average of this score over all the tokens in a tweet. Lastly, we detect the presence of consecutive exclamation (!) and question marks (?) in a tweet and use them as boolean features.

Hyper-parameter	**Ranges**	**Selected**
Embedding dimension	50/100/200	100
Number of filters	64/128/256	128
Kernel sizes	1/2/3/4/5	1/2/3
Batch size	16/30/50	16
Epochs	10/20	10
Dropout rate	0.1/0.2/0.5	0.5
Learning rate	0.015/0.001/0.01	0.001

Table 4.3: Ranges of different hyper-parameters searched during tuning and the final configurations selected for our experiments

4.4.2.3 Network Parameters and Training

Zhang and Wallace (2017) [101] performed a sensitivity analysis on various parameters of a one-layer CNN model and showed how tuning the parameters can affect the performance of a model. Inspired by the work done by [101], we also searched for the optimal parameter configurations in our network. Table 4.3 shows different hyper-parameter configurations that we tried and the final configuration that was used in our model. The final configuration was based on both performance and training time. The embedding dimension has been set to 100 for both of the channels of our network as it worked best for us among other dimensions. We also experimented with a different number of filters and varying kernel sizes during our experiments. The combination of kernel sizes, ($k = 1, 2, 3$) in the first channel and $k = 1$ in the second channel worked the best for us. We also experimented with various batch sizes and the performance of the model remained reasonably constant, though the training time varied significantly. We used the Adam optimizer [39] and the back-propagation [71] algorithm for training our model. Keras 2.2.0 was used for implementing our model under the Linux environment.

4.4.2.4 Regularization

In order to reduce overfitting, it is a common practice to employ regularization strategies in CNNs. In our experiments, we used dropout regularization [81] for both of the channels after the convolutional layer as seen in Fig 4.1. We experimented with three different dropout rates as seen in Table 4.3 and also with no dropout at all. The model works better when we apply dropouts after the convolutional layer.

4.5 Performance

In this section, we describe the results obtained through our experimentation. We use precision, recall, F1-score and accuracy as our evaluation metrics. These metrics are defined as:

$$\text{precision} = \frac{tp}{tp + fp} \tag{4.4}$$

$$\text{recall} = \frac{tp}{tp + fn} \tag{4.5}$$

$$\text{F1-score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{4.6}$$

Emotion	Dataset											
	BTD			TEC			CBET			SemEval		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
joy	68.4	77.4	72.6	67.4	77.1	71.8	58.1	56.1	57.1	78.5	70.1	74.1
sadness	72.7	74.5	73.6	48.8	53.7	50.9	38.0	43.3	40.5	62.6	41.0	49.6
anger	74.7	79.1	76.8	34.5	23.8	27.7	49.3	52.1	50.7	59.7	63.6	61.6
love	57.0	46.4	51.1	–	–	–	65.4	53.3	58.7	–	–	–
thankfulness	63.2	55.3	59.0	–	–	–	66.1	68.0	67.0	–	–	–
fear	57.6	38.3	46.0	61.5	57.2	58.6	70.3	69.6	70.0	51.6	71.9	60.1
surprise	88.1	16.1	27.1	55.9	50.2	52.5	51.0	55.3	53.0	–	–	–
guilt	–	–	–	–	–	–	53.8	49.6	51.6	–	–	–
disgust	–	–	–	67.4	77.1	71.8	59.3	61.0	60.2	–	–	–
Avg.	**68.9**	**55.3**	**58.0**	**55.9**	**56.5**	**55.6**	**56.8**	**56.5**	**56.5**	**63.1**	**61.7**	**61.3**

Table 4.4: Results (in %) of our model (MC-CNN) for four emotion-labeled datasets.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{4.7}$$

Here, tp represents true positive, fp represents false positive, tn represents true negative and fn represents false negative. We evaluated our model on four emotion labeled datasets. Table 4.4 shows the results for each emotion category for all of the datasets. For the BTD dataset, we trained our model with $1,136,305$ tweets, while we used $140,979$ and $142,602$ tweets as development and test data respectively. We used the same training, development and test sets as [96] except that our retrieved dataset contains fewer samples. We achieved relatively high F1-scores of **72.6%**, **73.6%** and **76.8%** for *joy, sadness* and *anger*, respectively, whereas for *surprise* we get a low F1-score of **27.1%**. This is probably due to the imbalanced nature of the dataset as can be seen in Table 4.1. The number of samples for *joy, sadness* and *anger* is much higher than for *surprise*. Our model achieves an accuracy of **69.2%**, whereas Wang et al. (2012) [96] reported an accuracy of **65.6%** when trained on a much larger dataset. We can not make direct comparison with [96] since we were not able to retrieve the full test set due to the unavailability of some tweets at the time of fetching data from Twitter. For the TEC dataset, we evaluated our model with 10-fold cross validation. Mohammad (2012) [51] reported an F1-score of **49.9%** with SVM, whereas our model achieves an F1-score of **55.6%**. For the CBET dataset, we used 80% of the data as the training set and the remaining 20% as the test set. We get an average F1-score of **56.5%**. We also used 10-fold cross-validation for the SemEval dataset and achieved an F1-score of **61.3%**.

Table 4.5 shows the performance of our model with 10-fold cross-validation on different sentiment datasets with two classes (positive and negative). For the STS-Gold dataset, our

Datasets	Methods	Positive			Negative			Average			Accuracy
		P	R	F1	P	R	F1	P	R	F1	
STS-Gold	A	70.5	74.1	72.2	88.0	86.0	87.0	79.3	88.0	79.6	82.3
	B	–	–	–	–	–	–	79.5	77.9	78.6	82.1
	C	–	–	–	–	–	–	–	–	77.5	80.3
	D	75.4	74.9	75.1	90.2	90.3	90.2	82.8	82.6	82.7	86.0
	Ours	87.9	82.0	84.5	92.1	94.6	93.3	90.0	88.3	88.9	**90.7**
STS-Test	D	88.0	89.5	88.7	87.2	85.4	86.3	87.6	87.4	87.5	87.6
	E	–	–	–	–	–	–	–	–	–	86.4
	F	–	–	–	–	–	–	–	–	–	83.0
	Ours	90.2	91.2	90.5	91.3	89.3	89.9	90.8	90.3	90.2	**90.3**
SS-Twitter	G	–	–	–	–	–	–	67.8	52.7	59.3	61.9
	F	–	–	76.6	–	–	69.2	–	–	72.9	73.4
	Ours	81.3	84.7	82.0	72.5	72.7	72.2	76.9	78.7	77.1	**79.3**

Table 4.5: Results (in %) of our model (MC-CNN) from 10-fold cross-validation compared against other methods for sentiment labeled datasets (2-class). Bold text indicates the best performance in a column. **A:** Thelwall-Lexicon (Updated + Expanded) [75]. **B:** SentiStrength [40]. **C:** SentiCircle with Pivot [74]. **D:** Deep Convolutional Neural Network [34]. **E:** Character to Sentence Convolutional Neural Network (CharSCNN) [20]. **F:** Maximum Entropy [73]. **G:** Globally Convergence based Quantum Language Model + Quantum Relative Entropy [100].

model achieves an accuracy of **90.7%** whereas the previous best accuracy (**86.0%**) was reported by Jianqiang et al. (2018) [34] with a deep CNN model. Our model achieves the best accuracy (**90.3%**) for the STS-Test dataset as well, while the previous best (**87.6%**) was reported in [34]. Dos Santos et al. (2014) [20] also experimented with the same dataset with their Character to Sentence CNN model (CharSCNN) and they reported an accuracy of **86.4%**. Lastly, for the SS-Twitter dataset, our model achieves an accuracy of **79.3%** whereas Zhang et al. (2018) [100] and Saif et al. (2013) [73] reported an accuracy of **61.9%** and **73.4%**, respectively.

Tables 4.6, 4.7 and 4.8 show the performance of three variants of our model on the emotion labeled datasets with all available emotion categories for each of the emotion datasets and the sentiment labeled datasets with all the available sentiment labels for each of the sentiment datasets. The first variant is a basic CNN model without hash-emo embedding or any additional features. The second variant includes the hash-emo embedding, while the last variant combines additional lexical features as well. It can be observed that when we introduce the second channel with hash-emo embedding, we get a significant increase in accuracy for most of the datasets. We can see in Table 4.6 that, for STS-Test and SS-Twitter datasets, we get better F1-scores for all three sentiment labels when we include the hash-emo embedding along with

Datasets	Methods	Positive			Negative			Neutral		
		P	R	F1	P	R	F1	P	R	F1
STS-Test	CNN	73.6	84.1	76.8	73.5	74.2	72.8	74.2	64.3	65.8
	MC-CNN†	63.1	83.4	70.3	76.5	70.4	72.8	71.6	53.9	60.8
	MC-CNN†‡	**80.3**	**83.8**	**81.4**	**87.5**	**81.2**	**83.5**	**79.2**	**77.9**	**77.7**
SS-Twitter	CNN	48.9	51.7	49.3	43.9	53.7	47.4	67.8	**66.8**	64.3
	MC-CNN†	61.6	62.0	61.4	55.3	65.5	59.7	71.6	62.7	66.4
	MC-CNN†‡	**65.1**	**65.4**	**64.0**	**56.2**	**65.7**	**60.0**	**72.2**	62.7	**66.7**

Table 4.6: Results (in %) of three variants of our model from 10-fold cross-validation for sentiment labeled datasets (3-class). Bold text indicates the best performance in a column.† represents the inclusion of Hash-Emo embedding into the network. ‡ represents the inclusion of external features into the network.

Models	Dataset			
	BTD	*TEC*	*CBET*	*SE*
CNN	66.1	54.3	53.8	56.3
MC-CNN†	68.5	57.6	56.1	59.8
MC-CNN†‡	69.2	58.9	56.4	62.0

Table 4.7: Comparison of results (accuracy in %) of three variants of our model. † represents the inclusion of Hash-Emo embedding into the network. ‡ represents the inclusion of external features into the network.

external lexical features. In Table 4.8, for SS-Twitter, we get a **4.1** percentage point increase in accuracy over the base model. Also for STS-Gold, we get a **2.3** percentage point increase. When we include the hash-emo embedding along with additional features in the network, we get an increase of **4.5**, **6.4** and **5.5** percentage points for STS-Gold, STS-Test and SS-Twitter, respectively, over the base model. In the case of the emotion labeled datasets in Table 4.7, inclusion of hash-emo embedding in the network gives us **2.4**, **3.3**, **2.3** and **3.5** percentage points increase in accuracy and the inclusion of additional features as well gives us **3.1**, **4.6**, **2.6** and **5.7** percentage points increase in accuracy for BTD, TEC, CBET and SE datasets, respectively, over the base models.

Table 4.9 and Table 4.10 show the cross-corpus results on the emotion and sentiment labeled datasets used in our experiments. We trained our model with one dataset and evaluated its performance on the other datasets. For cross-corpus evaluation on the emotion labeled datasets, we only consider four basic emotions *anger, fear, joy* and *sadness*, as these four are common to all of them. (It should be noted that in tension situation negative emotions such as *anger, fear* and *sadness* can play a vital role.) For each emotion labeled dataset, we randomly chose $8,000$ tweets as training samples and $2,000$ tweets as testing samples while making sure that the classes remain balanced. As we can see in Table 4.9, our model gives comparatively better

Models	**Dataset**		
	STS-Gold	*STS-Test*	*SS-Twitter*
CNN	86.2	75.1	59.1
MC-CNN†	88.5	70.6	63.2
MC-CNN†‡	90.7	81.5	64.6

Table 4.8: Comparison of results (accuracy in %) of three variants of our model. † represents the inclusion of Hash-Emo embedding into the network. ‡ represents the inclusion of external features into the network.

	BTD	*TEC*	*CBET*	*SE*
BTD	–	42.4	46.7	48.9
TEC	53.2	–	44.2	36.4
CBET	63.2	42.8	–	41.2
SE	68.8	40.1	43.3	–

Table 4.9: Cross-corpus results (Accuracy in %). Rows correspond to the training corpora and columns to the testing.

results when trained on CBET and SE datasets. In Table 4.10, we can observe that, STS-Gold and STS-Test perform comparatively better than the SS-Twitter dataset. Our model achieves an accuracy of **81.1%** and **78.4%** on the STS-Test and SS-Twitter datasets, respectively, when trained on the STS-Gold dataset which has two sentiment classes (positive/negative). Though the STS-Test dataset has relatively fewer positive and negative samples, our model still generalizes well when trained on this dataset. As discussed earlier, this is probably due to the fact that both STS-Gold and STS-Test were labeled with positive-negative sentiment in comparison with the SS-Twitter dataset which was originally labeled with numerical numbers for the sentiment.

It should be noted that our interview datasets do not have hashtags, emoticons or emojis which are found in Twitter data. Hence, during training a model with Twitter data for detecting emotion in our interview transcripts, we only use the first channel in our proposed model for emotion recognition along with the external feature vector.

	STS-Gold	*STS-Test*	*SS-Twitter*
STS-Gold	–	81.1	78.4
STS-Test	76.9	–	72.6
SS-Twitter	31.1	50.7	–

Table 4.10: Cross-corpus results (Accuracy in %). Rows correspond to the training corpora and columns to the testing.

4.6 Summary

In this chapter, we have discussed our emotion recognition model and its effectiveness on predicting emotion from text. To the best of our knowledge, our model achieves the best accuracies on the three sentiment datasets, and has significant improvement in performance on the four emotion labeled datasets over the basic CNN model. The model performs even better when additional lexical features are introduced into the network. The trained model can be incorporated with our tension detection architecture to identify emotion in interviewee's responses.

Chapter 5

A Lexicon-based Approach for Identifying Hedges

This chapter discusses in detail hedging in conversation management and recent computational approaches which have been undertaken for detecting such a phenomenon. We propose a new algorithm for detecting sentence-level hedges utilizing two manually constructed lexicons of hedge words and discourse markers. We also describe the data that we used along with the annotation procedure. We conclude the chapter by comparing our proposed algorithm with some standard machine learning algorithms and show the effectiveness of our approach.

5.1 Introduction

The concept of *hedging* was first introduced by Lakoff [41]. He defined *"hedges"* as words which a speaker uses to add fuzziness or uncertainty in the propositional content of a sentence. According to Silva et al. (2001) [18], hedging can be viewed as a speaker's attitude towards a claim and towards their audience and thus can have a huge influence on conversational dynamics. Hedging can be as simple as saying *"maybe"*, *"almost"*, or *"somewhat"* in ordinary discourse. This phenomenon is widely used in conversations where speakers show their lack of commitment to what they communicate. For example, the sentence *"I **assume** he was involved in it."* shows how the usage of the hedge word *"assume"* can weaken the propositional content *"he was involved in it"*. Prince et al. [67] categorized hedges into two distinguishable categories: propositional hedges and relational hedges. Propositional hedges exhibit fuzziness within the propositional content, whereas, relational hedges exhibit fuzziness between the proposition and the speaker. In Example 1, the hedge term *"think"* is used as a relational hedge, while in Example 2, the hedge term *"sort of"* is used as a propositional hedge.

(1) I **think** it came naturally to me, I was learning quickly.

(2) There was a **sort of** madness to it too.

The concept of hedging is particularly important in oral history interviews where it can be treated as a crucial element of discourse function. This phenomenon is adopted by the interviewees when they try to avoid criticism or evade questions [17]. This can affect the conversational dynamics to such an extent that sometimes two opposite forces come into play causing tension between the interviewer and the interviewee. An example of such a question/answer pattern in a survivor interview is given below:

> **Interviewer:** You came to Canada to study, you finished your master's degree, you are now a doctoral student. What is your message to - I was going to say to the survivors, but I'm thinking especially of the young survivors who've been through almost the same situations or worse - what is your message to the young survivors?
>
> **Narrator:** I don't know if I **would** address the young survivors specifically, since everyone is dealing with this experience in their own way, so I don't want to assume the role of a counselor. But from my personal experience, some of which is shared by others, it is more important to convey a message ... to the young survivors' circle of friends and family. And of course the survivors ... I **think** that every survivor's story must be heard in the singularity of experience that it recounts.

The above example shows hedging in the narrator's response. The use of hedge terms *"would"* and *"think"* demonstrates the instability in their narrative. This is frequent when there is a disjuncture between the interviewer and the narrator. Often interviewees insist on individualizing their narrative, because they either don't feel authorized to speak for the group, or a realization that their story is theirs.

Hedge words that are composed of multiple words are simply called multi-word hedges. In some cases, some words alone might not demonstrate the effect of hedging unless used with certain other words. For example, the sentence *"**In my view**, this attitude produced through social discourse can also change things within families."* shows how the multi-word hedge can be used during conversations. The phrase *"in my view"* acts as an important indicator of hedging here. The speaker didn't want to take responsibility and talk on behalf of a larger group when giving his/her proposition. Rather he/she individualized his/her opinion with the use of a multi-word hedge. The words *"in"*, *"my"* and *"view"* though can not show any hedging when used independently.

Often, as a substitute for hedge words, discourse markers are used during oral history interviews. A discourse marker can be an utterance or a word or a phrase (such as oh, like, well,

and you know) that either direct or redirect the flow of conversation without adding any significant meaning to the discourse [76]. The example below shows an example of the discourse marker *"well"*. Jucker et al. (1993) [35] showed how it has a profound effect on conversational dynamics.

> **Interviewer:** I'd like to hear your thoughts about that first, your point of view on these different ways of approaching reconciliation, and if there really should be a reconciliation - how do you see that? We will talk about forgiveness after.
>
> **Narrator:** **Well**, for me, reconciliation necessarily involves all the actors: the victims as well as the perpetrators of evil. Each actor has a role to play...

In conversation, discourse markers such as *"well"* have various functions [66]. We can use *"well"* to show a slight change in topic or when what we want to say is not quite what is expected or as a pause filler in the face of an interactive difficulty. For example, *"I think..., **well**, I've never compared them, it's a bit of a difficult question, but I think it's different."* We can also use *"well"* when we want to change what we have just said, or say something in a different way. Such an example is given below:

> **Interviewer:** And in those moments of panic when you were at the convent, was everyone concerned by what was going on? Is it..., was there... Was there solidarity between all of you who were staying with the nuns? You were all in this together, how did it...
>
> **Narrator:** No, life continued. **Well**, I can't know everything that was going on in the communes, but we knew that the Tutsis were targeted, that it was about the Tutsis...

Discourse markers are more common in informal speech compared with writing. Simon (2005) [79] showed various functions of discourse markers. Discourse markers can be used to shift a topic either completely or partially. It can also be used as a filler or delaying tactic or to preface a reaction or response. The following example shows how the discourse marker *"you know"* affects the dynamics of the ongoing conversation.

> **Interviewer:** ... Can we say - has there been - could there be harmony between the Tutsis, harmony between the Hutus? Do Tutsis and Hutus manage to communicate in an authentic, deep and sincere way?
>
> **Narrator:** It's very difficult because, **you know**, the problem - an identity problem is very difficult to delineate. And I can't say either that complete harmony reigns between the different groups, the different identities that we have, Hutus and Tutsis, etc. It's not 100 percent harmony either.

5.2 Related Work

In this section, we describe different computational approaches that have been undertaken in recent years to identify hedging in text.

Though the term *hedging* was first introduced by Lakoff (1975) [41] in the 1970s, it gained popularity among the NLP community about two decades ago. One of the earliest works was by Light et al. (2004) [45]. They constructed a dictionary of hedge cues to identify speculative (hedged) sentences in MEDLINE abstracts. They also used a Support Vector Machine (SVM) as a classifier to determine speculative sentences in the abstracts. However, their constructed list of hedge terms is biased towards the bio-medical domain and needs to be refined for it to work well in other domains, as well.

Medlock and Briscoe (2007) [47] treated the problem of determining speculative sentences as a classification task. They used single words as features to build their classifier using a set of semi-automatically collected training examples. Szarvas (2008) [82] used the same dataset as [47], but in their experiments, they used bigrams and trigrams as features. They trained a maximum entropy model classifier by providing binary data about whether single features occurred in the given context or not.

Ganter and Strube (2009) [25] proposed a hedge detection system based on word frequency measures and syntactic patterns by using weasel word information in the readily available Wikipedia corpus. Their proposed approach can be easily extended to other languages as well which makes it more robust. Özgür and Radev (2009) [62] built a supervised machine learning model using a diverse set of features. They used keyword based features, positional information of the keywords as well as contextual information surrounding the keywords. They also used syntactic structures of the sentence to determine the scope of the hedge cues. Agarwal and Yu (2010) [2] used a conditional random field (CRF) algorithm to train models in order to identify hedge cue phrases and their scopes in the bio-medical domain. They worked on the BioScope corpus [83] and their proposed model performed very well on biological literature and clinical notes.

The problem of detecting hedges in natural language texts was addressed in the CoNLL 2010 shared task [23] and it allowed the computational linguistics community to dig deeper into the task of detecting hedges in sentences. The shared task was divided into two sub-tasks. The goal of one of the sub-tasks was to identify sentences containing uncertainty, while the other sub-task focused on identifying the scope of hedge cues. Several teams participated in the shared task and proposed different techniques to tackle the problems [26][86][93][94]. Georgescul (2010) [26] used a Support Vector Machine (SVM) classifier based on a Gaussian Radial Basis Kernel Function (RBF) for the same task and tuned the hyper-parameters accord-

ing to the domain and achieved the best F1-score for one of the datasets. Tang et al. (2010) [86] used a combination of a Conditional Random Field (CRF) model and a Large Margin-based model to identify hedged sentences. Velldal et al. (2010) [93] used a maximum entropy (MaxEnt) classifier using syntactic and surface-oriented features to identify hedge cues. Vlachos and Craven (2010) [94] used syntactic dependencies and proposed a logistic regression model to identify speculative sentences.

More recently, Uliski et al. (2018) [92] proposed a set of manually constructed rules which allowed them to identify hedged sentences in forum posts in an unsupervised manner. Theil et al. (2018) [88] expanded a lexicon of uncertainty trigger words utilizing domain specific word-embedding models. They used term frequency (TF) and term frequency-inverse document frequency (TF-IDF) for feature representation and experimented with several machine learning models. Their expanded lexicon significantly boosts the performance of detecting uncertain sentences in the financial domain.

This section describes the approaches that have been undertaken for the purpose of hedging detection. Most of the approaches either are keyword based or keywords have been used as features in different machine learning models. The problem of identifying hedged or speculative sentences has been highly studied in the CoNLL 2010 shared tasks. Though most of the participants have used different traditional machine learning classifiers for this binary classification task, the lexicons that have been used, are highly domain dependent and further studies need to be done in other domains.

5.3 Lexicons

In this section, we describe two manually constructed lexicons that have been used in this study. One for hedge words and the other one for discourse markers. We have compiled our lexicons using various online resources. We have also included a number of discourse markers and hedge words in our lexicons from [85]. We provide the lexicons in Appendix B and Appendix C.

We compiled 76 potential hedge words. Since epistemic words highly contribute to hedging, we included different epistemic verbs *(assume, think, suspect)*, adverbs *(arguably, presumably, allegedly)*, adjectives *(probable, unsure, unclear)* and modal verbs *(might, maybe)* in our lexicon for hedge words. With epistemic modality, a speaker's level of confidence on his/her proposition can be determined. We also included various approximators such as *(generally, usually)* in the lexicon.

As discussed earlier, discourse markers can also be very useful in detecting possible tension points in interviews. These markers have a variety of functions. We give a few examples of

such markers with their probable functions during conversations:

- Making an unexpected contrast: *even though; despite the fact that*
- Making a contrast between two separate things, people, ideas, etc.: *anyway; however; rather*
- Clarifying and re-stating: *in other words; in a sense; I mean*
- Dismissal of previous discourse: *anyhow*
- Showing the attitude of what you are saying: *I think; I feel; in my understanding*
- Preparing for something unwelcome: *I'm afraid; honestly*
- To change topic or return to the topic: *well, anyway*
- Indicating a difference of opinion: *yes, but*
- Conversation management: *I mean; you know; so to speak; more or less*
- Indicating agreement with a negative idea: *Yes, no, I know*

5.4 Rules for Disambiguating Hedge Terms

In this section, we discuss several hedge terms which are commonly used in interviews but which have other non-hedge senses. We give rules that we employed to disambiguate the most frequent ambiguous hedge terms in our corpus. In order to identify the terms in the corpus we make use of lemmatization. It is to be noted that we did not cover all the hedge terms from our hedge lexicon and the set of rules can be expanded in the future to cover other ambiguous hedge terms as well. Our rules are an extension and modification of the set of rules proposed by [92]. What follows is a detailed analysis of the rules used in our study with examples derived from our survivor interview datasets. We used Stanford CoreNLP (version 3.9.1) [46] for dependency parsing in our study.

- **Feel/Suggest/Believe/Consider/Doubt/Guess/Presume/Hope**
  Rule: If token *t* is (i) a *root* word, (ii) has the part-of-speech *VB** and (iii) has an *nsubj* (nominal subject) dependency with the dependent token being a first person pronoun *(i, we)*, *t* is a hedge, otherwise, it is a non-hedge.

  Hedge: I **hope** to, someday, but no, I haven't reached it yet. (Fig. 5.1)
  Non-hedge: A message of **hope** and daring to shed light on everything we see. (Fig. 5.2)

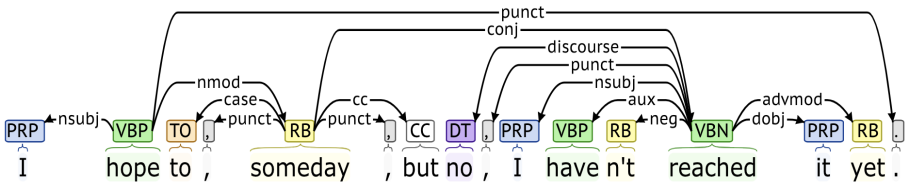

Figure 5.1: Dependency tree for the example *"I hope to, someday, but no, I haven't reached it yet."*

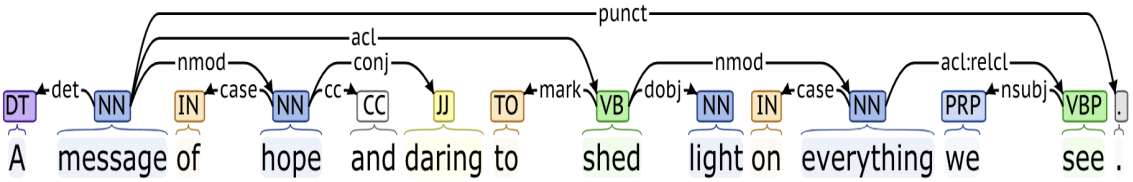

Figure 5.2: Dependency tree for the example *"A message of hope and daring to shed light on everything we see."*

- **Think**

  Rule: If token *t* is followed by a token with part-of-speech *IN*, *t* is a non-hedge, otherwise, it is a hedge.

  Hedge: I **think** it's a little odd. (Fig. 5.3)

  Non-hedge: I **think** about this all the time. (Fig. 5.4)

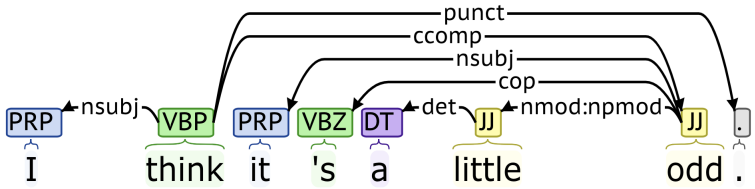

Figure 5.3: Dependency tree for the example *"I think it's a little odd."*

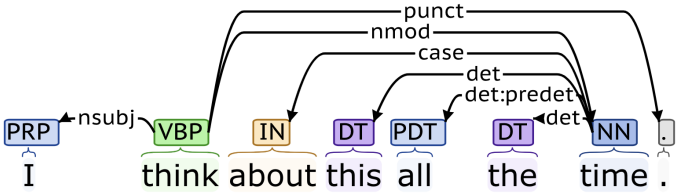

Figure 5.4: Dependency tree for the example *"I think about this all the time."*

- **Assume:**

  Rule: If token *t* has a *ccomp* (clausal complement) dependent, *t* is a hedge, otherwise, it is a non-hedge.

Hedge: I **assume** he was involved in it. (Fig. 5.5)

Non-hedge: He wants to **assume** the role of a counselor. (Fig. 5.6)

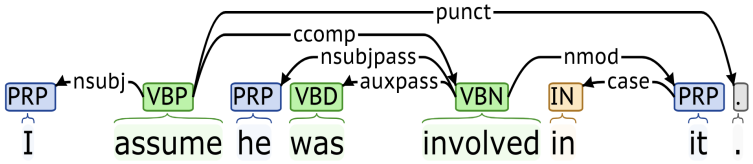

Figure 5.5: Dependency tree for the example *"I assume he was involved in it."*

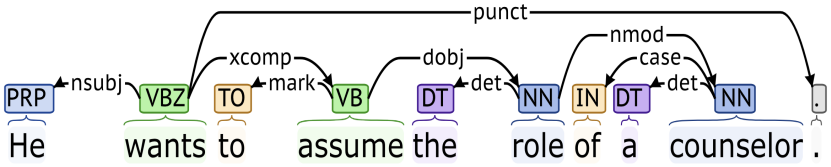

Figure 5.6: Dependency tree for the example *"He wants to assume the role of a counselor."*

- **Appear:**

  Rule: If token *t* has a *ccomp* (clausal complement) or *xcomp* (open clausal complement) dependent, *t* is a hedge, otherwise, it is a non-hedge.

  Hedge: The problem **appeared** to be more serious than we thought. (Fig. 5.7)

  Non-hedge: A man suddenly **appeared** in the doorway. (Fig. 5.8)

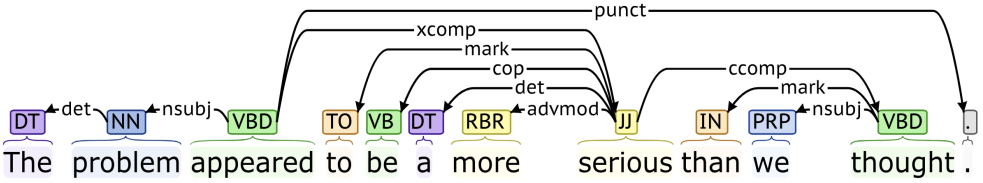

Figure 5.7: Dependency tree for the example *"The problem appeared to be more serious than we thought."*

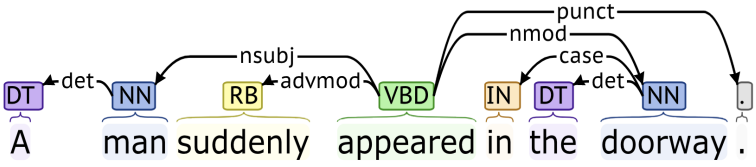

Figure 5.8: Dependency tree for the example *"A man suddenly appeared in the doorway."*

- **Suppose:**

  Rule: If token *t* has an *xcomp* (open clausal complement) dependent *d* and *d* has a mark dependent *to*, *t* is a non-hedge, otherwise, it is a hedge.

Hedge: I **suppose** they were present here during the genocide. (Fig. 5.9)

Non-hedge: I'm not **supposed** to go back there again. (Fig. 5.10)

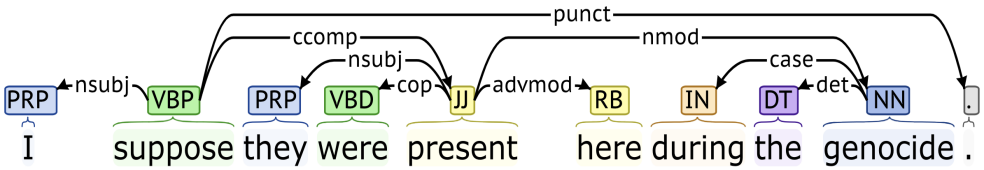

Figure 5.9: Dependency tree for the example *"I suppose they were present here during the genocide."*

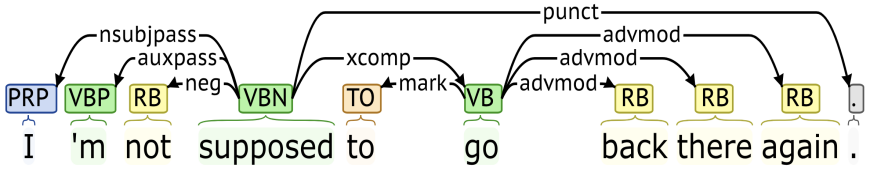

Figure 5.10: Dependency tree for the example *"I'm not supposed to go back there again."*

- **Tend:**

  Rule: If token *t* has an *xcomp* (open clausal complement) dependent, *t* is a hedge, otherwise, it is a non-hedge.

  Hedge: We **tend** to never forget. (Fig. 5.11)

  Non-hedge: All political institutions **tended** toward despotism. (Fig. 5.12)

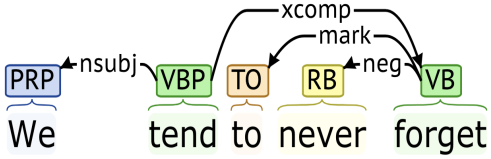

Figure 5.11: Dependency tree for the example *"We tend to never forget"*

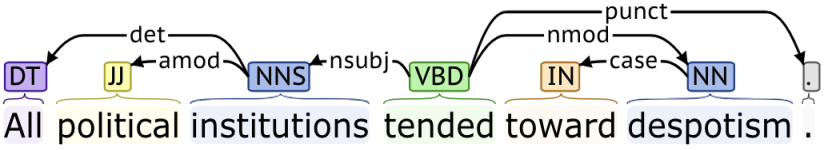

Figure 5.12: Dependency tree for the example *"All political institutions tended toward despotism."*

- **Should:**

  Rule: If token *t* has relation *aux* with its head *h* and *h* has dependent *have*, *t* is a non-hedge, otherwise, it is a hedge.

Hedge: Perhaps I **should** be asking how we should all consider each other. (Fig. 5.13)
Non-hedge: He **should** have been more careful. (Fig. 5.14)

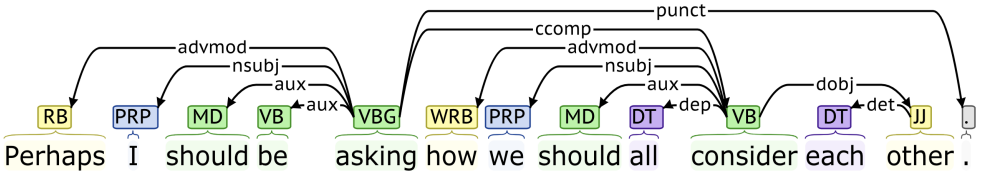

Figure 5.13: Dependency tree for the example *"Perhaps I should be asking how we should all consider each other."*

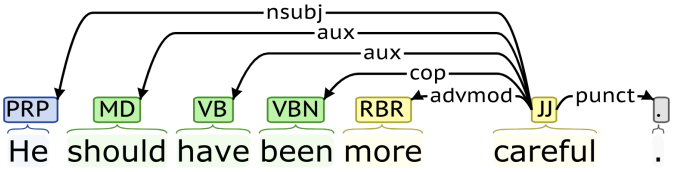

Figure 5.14: Dependency tree for the example *"He should have been more careful."*

- **Likely:**

  Rule: If token *t* has relation amid with its head *h* and *h* has part of speech *N**, *t* is a non-hedge, otherwise, it is a hedge.

  Hedge: We will **likely** visit there once again. (Fig. 5.15)
  Non-hedge: He is a fine, **likely** young man. (Fig. 5.16)

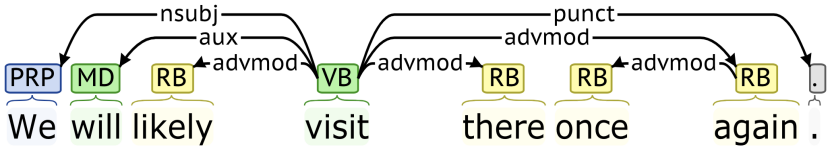

Figure 5.15: Dependency tree for the example *"We will likely visit there once again."*

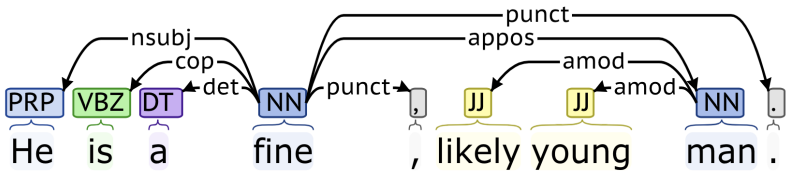

Figure 5.16: Dependency tree for the example *"He is a fine, likely young man."*

- **Rather:**

  Rule: If token *t* is followed by token *than*, *t* is a non-hedge, otherwise, it is a hedge.

  Hedge: She's been behaving **rather** strangely.
  Non-hedge: She seemed in-different **rather** than angry.

5.5 Experiments

In this section, we briefly discuss about our data source and annotation procedure. We also discuss our proposed algorithm for hedge detection and it's performance on the annotated dataset in comparison with other methods.

5.5.1 Data

We collected our data from the living archives of Rwandan exiles and genocide survivors in Canada[1]. This digital repository contains life stories of Rwandan genocide survivors. In this thesis, we worked with the transcribed interviews which are translated into English. We provide statistics of our interview corpora in Table 5.1. For each corpus, we report the number of question-answer pairs, along with the total number of sentences and words in the narrator's responses. We also report the number of times hedge words, discourse markers and booster words are used in the corpus.

Interview	*#Ques-Ans*	*#Sentences*	*#Words*	*#Hedges*	*#Discourse-Markers*	*#Boosters*
1	135	718	17,289	162(0.94%)	92(0.53%)	118(0.68%)
2	74	335	7,430	84(1.13%)	18(0.24%)	40(0.54%)
3	113	470	11,162	112(1.00%)	37(0.33%)	85(0.76%)
4	94	454	11,652	119(1.02%)	29(0.25%)	68(0.53%)
5	127	1,032	19,959	158(0.79%)	31(0.16%)	89(0.45%)
6	65	959	24,208	184(0.76%)	55(0.23%)	174(0.72%)
Total	**608**	**3,968**	**91,700**	**819**(0.89%)	**262**(0.29%)	**574**(0.63%)

Table 5.1: Statistics of our interview datasets

5.5.2 The Annotation Procedure

The task of distinguishing hedged sentences from non-hedged ones is not straightforward. In order to get an understanding of the concept of hedging and also the specifics that need to be kept in mind while annotating, the annotators for this study followed the gold standard set by Farkas et al. (2010) [23] in the CoNLL 2010 shared task. The two human annotators in this study have been given 50 random samples from the CoNLL 2010 shared task dataset which have been annotated as certain or uncertain. One of the advantages of this dataset is that it specifies hedge cues or phrases and their scope in the uncertain sentences, allowing one to get an understanding of the hedging concept and its usage.

[1] http://livingarchivesvivantes.org/

Annotator 1 \ Annotator 2	*yes*	*no*
yes	60	13
no	8	119

Table 5.2: Summary of annotations by two human annotators. Rows correspond to annotator 1's response and columns correspond to annotator 2's response. "yes" represents that annotator thinks there is hedging and "no" represents otherwise.

One of the human annotators for this study, the author of this thesis, annotated 3000 sentences from complete interviewee responses from the interview corpora. Due to time and resource constraints, the other annotator has been given only 200 sentences from the 3000 sentences annotated by the first annotator. The second annotator was not given the labels determined by the first annotator. The second annotator, who is an Engineer by profession, annotated these 200 sentences. We provide the summary of the annotation results in Table 5.2.

It is important to measure the agreement level between the human annotators to have reliable annotations. As a result, we used Cohen's kappa shown in Equation 5.1 to evaluate the degree of agreement between the choices made by the two annotators. Cohen's kappa [15] is a statistical coefficient that represents the degree of accuracy and reliability in a statistical classification.

$$k = \frac{p_o - p_e}{1 - p_e} \tag{5.1}$$

where p_o is the relative observed agreement among annotators and p_e is the hypothetical probability of chance agreement. The kappa value we obtained for this study is 0.77, which falls in the "substantial agreement" range according to the interpretation of kappa values provided by Landis and Koch (1977) [42]. Thus, the measurement proves that the 200 sentence annotated corpus is reliable and can be used for performance evaluation. The 179 sentences that the two annotators agreed on were used in the evaluation of the following algorithm.

5.5.3 Proposed Algorithm

We give pseudo-code for our proposed approach in Algorithm 3. In order to find similarity of discourse markers in our lexicon with phrases from sentences, we make use of the Jaccard index shown in Equation 5.2. It measures similarity between finite sample sets.

$$J(X, Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cup Y|} \tag{5.2}$$

Here, $J(X, Y)$ represents Jaccard index for X and Y, where X and Y are sets of words. On the other hand, Jaccard distance measures dissimilarity between sample sets and is complementary to the Jaccard index. We used this measure, shown in Equation 5.3, in our hedge detection algorithm.

$$d_J(X, Y) = 1 - J(X, Y) = \frac{|X \cup Y| - |X \cap Y|}{|X \cup Y|} \tag{5.3}$$

Here, $d_J(X, Y)$ represents Jaccard distance between X and Y, where X and Y are sets of words.

Algorithm 3 Hedge Detection Algorithm

```
function isTrueHedgeTerm(t)
    Predefined rules to disambiguate hedge terms
    if t is true hedge term then
        return True
    else
        return False
    end if
end function
function isHedgedSentence(s)
    Discourse Markers (DM) ← List of discourse markers
    Hedged Words (HG) ← List of Epistemic words
    Phrases (P) ← List of n-grams from s
    status = False
    for A in DM do
        for B in P do
            if 1 - jaccard_distance(A,B) ≥ threshold then
                status = True
            end if
        end for
    end for
    for hedge in HG do
        if hedge in s AND isTrueHedgeTerm(hedge) then
            status = True
        end if
    end for
    if status is True then
        Mark s as Hedged sentence
        return True
    else
        Mark s as NonHedged sentence
        return False
    end if
end function
```

	Hedge			Non-hedge			Avg			*Acc*
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	
SVM	48.7	28.4	32.0	72.0	**89.0**	78.9	60.3	58.7	55.4	68.8
LR	49.9	38.3	40.3	73.9	83.9	77.9	61.9	61.1	59.1	68.7
HD	**70.2**	**98.3**	**81.9**	**98.9**	78.9	**87.8**	**84.5**	**88.6**	**84.8**	**85.4**

Table 5.3: Comparison of results (in %) (2 annotators) of our model against other methods. Bold text indicates the best performance in a column. HD refers to our hedge detection algorithm.

5.5.4 Performance

To measure the effectiveness of our hedge detection algorithm, we utilized the 200 annotated samples mentioned earlier in this chapter. Firstly, we took only those samples from the annotated dataset, where both the annotators agreed on the label, giving us a total of 179 samples (84 hedged and 95 non-hedged sentences). We compared our results against two standard machine learning algorithms: Support Vector Machine (SVM) and Logistic Regression (LR). We performed a 10-fold cross-validation (Stratified) on both classifiers using a bag of n-grams model where we chose unigram, bigram and trigrams as our features. The comparison results are shown in Table 5.3.

From Table 5.3, we can observe that our hedge detection algorithm performs comparatively better than both machine learning classifiers, achieving an accuracy of **85.4%**, whereas SVM and LR with bag-of-words model achieved **68.8%** and **68.7%** respectively. Although SVM achieved a better recall score for the *Non-hedge* label, our hedge detection algorithm achieved a better F1-score for that label.

Secondly, we took the samples annotated by only the first annotator and used them as the ground truth. This way, we had 200 annotated sentences. We performed the same experiments as above and results can be seen in Table 5.4. It can be observed that, our algorithm still

	Hedge			Non-hedge			Avg			*Acc*
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	
SVM	68.9	46.7	52.1	73.9	**84.2**	78.1	71.4	65.5	65.1	70.5
LR	55.2	38.5	43.7	70.5	81.9	75.4	62.9	60.2	59.5	66.1
HD	**69.0**	**94.5**	**79.7**	**96.0**	75.6	**84.6**	**82.5**	**85.0**	**82.1**	**82.5**

Table 5.4: Comparison of results (in %) (1 annotator) of our model against other methods. Bold text indicates the best performance in a column. HD refers to our hedge detection algorithm.

	Hedge			**Non-hedge**			**Avg**		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
SVM	86.2	60.3	70.9	70.2	81.1	75.2	78.2	70.2	73.0
LR	**91.1**	57.1	70.2	73.2	**84.1**	78.3	**82.1**	70.6	74.2
HD	66.1	**91.3**	**76.7**	**93.7**	70.8	**80.6**	79.9	**81.0**	**78.6**

Table 5.5: Comparison of results (in %) (1 annotator) of our model against other methods. Bold text indicates the best performance in a column. HD refers to our hedge detection algorithm.

performs better than the machine learning classifiers, achieving an accuracy of **82.5%** in comparison with the accuracies achieved by SVM and LR with a bag of n-grams model. Finally, we compare the performance of our proposed hedge detection algorithm on the 3000 sentences annotated by annotator 1 against the machine learning algorithms in Table 5.5.

5.6 Summary

Hedging plays an important part in conversational management, thus making it a crucial component for our tension detection architecture. In this chapter, we have discussed our approach for identifying hedges at the sentence-level.

We constructed two lexicons: one for hedge words and one for discourse markers. We also discussed rules to handle ambiguous hedge terms. To measure the effectiveness of our proposed approach, we got two annotators to annotate a portion of our interview corpora. We also discussed the annotation procedure where we have seen that the kappa value is 0.77, which is an acceptable inter-rater agreement measure according to Landis and Koch (1977) [42]. We also compared the performance of our proposed approach against two standard machine learning algorithms and showed the effectiveness of our approach.

Chapter 6

Performance Analysis

In this chapter, we discuss the overall performance of our tension detection architecture. We provide experimental results on the two annotated interview transcripts that are available to us. We also discuss briefly the results that we obtained during our experimentation. Finally, we discuss some shortcomings of our proposed approach and provide ideas for improvement.

The data used for our experiments consists of two annotated interview transcripts. It should be understood that although this is not a lot of data, the data is not readily available. The audio of the interview must first be transcribed, and since it is either in the Rwandan language or French, it must then be translated into English. Then it must be analyzed to find the points of tension. This last step has been performed by a group of oral history interview experts discussing the interviews and coming to a collective agreement on what constitutes a point of tension. We accept this decision making as providing us with a rigorous gold standard that is different than the normal individual annotations done by individual annotators. The downside of this process is that it is time consuming and so this explains the small amount of data that we have for the following experiment.

6.1 Experimental Results

We experimented on the two available transcribed interview transcripts which have been annotated for tension by a group people having expertise in the oral history interview domain. Table 6.1 shows the results obtained on the two transcripts with our approach described in the previous three chapters. We refer the readers to Chapter 3 for the pseudocode of our proposed algorithm.

We observe high recall scores of **85.7%** and **75.0%** for the two transcripts, respectively. However, the precision scores are very low. We can see from Table 6.1 that we have a high number of false positives, which means our approach identifies such tension points in the tran-

	TP	*FP*	*TN*	*FN*	*Prec.*	*Rec.*
	Berthe Interview					
Tension	6	76	46	1	7.3%	85.7%
	Yvette Interview					
Tension	3	56	34	1	5.1%	75.0%

Table 6.1: Results of our tension detection algorithm.

scripts which were not marked previously by the human judges. It will be interesting to evaluate the findings of our model with the human judges, which might reduce the number of false positives, increasing the overall precision scores. So, our proposed approach can be used as a filtering tool as well for the human experts.

Gratch et al. (2014) [29] provides a corpus that can be used to build models for identifying psychological distress in interviews. We used this dataset as a surrogate for tension detection. We collected 40 such interviews (20 interviews with distress and 20 without distress). We evaluated our proposed approach with this collection of interviews. An interview is deemed to be one that displays distress if it contains at least a certain number of points of tension. We achieved a precision score of **46.7%** and a recall of **74.2%** for identifying distress when using a threshold of 5 (which means we identify a particular interview transcript as distress if there are at least 5 tension points in that transcript). We experimented with different thresholds between 1 to 10 and the above mentioned threshold gave us the best precision and recall scores.

6.2 Discussion

Our proposed approach achieves a good recall score overall. To achieve the first purpose, the algorithm needs some improvement as the number of false positives is high. We have analyzed a few examples from our dataset which were mislabelled. We discuss them here.

One of the problems that must be tackled by any description of discourse markers is their poly-functionality which means it is very important to distinguish the usage of different markers. Although we tried to disambiguate a number of hedge terms in this thesis, we still need a better understanding of some discourse markers. One of the issues with our approach is its inability to differentiate different discourse functions of the marker *"well"*. *"well"* has various functions. It has been well studied over the years by many researchers [66],[35]. It appears in seemingly different contexts. According to Jucker (1993), *"well"* can be used as a marker of insufficiency, as a face-threat mitigator, as a frame, or as a delay device. The discourse marker *"well"* has several homonyms, for instance a manner adverb *(He sings well)*, a degree word

(He knows the concept perfectly well), a noun *(Everyone digs their own well)*, and a verb *(Tears well in her eyes)*. So it is very important to understand its functions and make decisions upon that. For example, these responses by narrators were marked incorrectly as containing tension by our proposed approach:

(1) "I remember really well my maternal grandparents."

(2) "Uh, not at all. Some received job offers even before they left here, and they are feeling well there, they are doing well"

(3) "Well, for sure Sam was someone who - in terms of sports, he was very athletic and I wanted to be better than him."

So it is necessary to have rules that can disambiguate different markers as well. Moreover, we trained our model for recognizing emotion from text using Twitter data. We would achieve better results if we could train our emotion recognition model using training data from the oral history interview domain. To accomplish this we would need a sufficient amount of annotated data to serve that purpose and this data will not be available in the foreseeable future.

We had two purposes when setting out to determine points of tension in oral history interviews: 1) to locate points of tension, 2) to filter the interview to reduce the number of transcribed text that human experts would need to look at when looking for points of tension in the interviews. We have been reasonable successful in the second purpose. It should be understood that the first purpose is difficult, even for human experts.

6.3 Summary

In this chapter, we have discussed our experimental findings on two of our interview transcripts. Our proposed approach achieves a high recall score, allowing it to be used as a filtering tool for the human experts. We also discussed the results obtained on a distress corpus.

Lastly, we discussed some of the shortcomings of our approach with examples. We also provided ideas for improvement. We expect a boost in performance once these issues are tackled.

Chapter 7

Conclusion and Future Work

This chapter gives concluding remarks for the research work that has been done for this thesis. It also discusses possible future improvements in this research field.

7.1 Conclusion

In this thesis, we first discussed in detail the interview dynamics and how different factors affect this phenomenon. We also talked about survivor interviews and why it is important to analyze a narrator's responses in order to identify tension situations.

We provided in-depth analysis of emotion which plays a significant part in conversational management. Emotions can be observed in survivor interviews and most of the time, negative emotion tends to be the cause of a tension situation. We showed with our experiments that the CNN models can be very effective in determining emotion from text. We showed how external features can be incorporated in a CNN network and how a separate channel in a CNN model can be effective in classification. We also provided cross-corpus comparisons to show the effectiveness of our model.

Next, we presented a discussion about hedging and boosting in speakers' narratives. These two phenomena are crucial in tension detection studies and show a speaker's attitude during a conversation. We proposed a simple lexicon-based approach for hedge detection and showed its effectiveness when compared with other machine learning algorithms. We discussed our annotation studies and our obtained kappa value which verified the reliability of the two-person annotation.

We showed how to integrate these components into our tension detection architecture by providing pseudo-code for each component. Our proposed algorithm gives a very good recall score for the oral history interview dataset that has been used in this thesis. However, the precision score can be improved. It also shows decent performance when applied to a distress

interview corpus. Our proposed architecture can be used as a filtering tool because of its high recall score which can help researchers in this field of work. Since, there has been very little work done in this research field, we believe our research findings can be helpful in the future continuation of this research.

7.2 Future Work

It is important to have gold standard datasets in order to evaluate a model and to train machine learning models. Plans to annotate more interview datasets by people with domain expertise are underway. It is crucial to have a good understanding of tension phenomenon in order to better analyze such data. The domain experts at the Centre for Oral History and Digital Storytelling at Concordia University will be performing this annotation.

We plan to train our proposed neural network model for emotion detection using data from our interview corpus. To achieve this, we plan to get our interview datasets annotated with different emotion categories. We also plan to get more data annotated for hedging so that we can do a better evaluation of our hedge detection algorithm. Better methods for determining when a poly-functional word is a discourse marker can be incorporated. Finally, we would like to analyze our interview dataset more closely to find semantic patterns in narrators' responses when a tension situation arises.

Bibliography

[1] Muhammad Abdul-Mageed and Lyle Ungar. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 718–728, 2017.

[2] Shashank Agarwal and Hong Yu. Detecting hedge cues and their scope in biomedical text with conditional random fields. *Journal of biomedical informatics*, 43(6):953–961, 2010.

[3] Ameeta Agrawal and Aijun An. Unsupervised emotion detection from text using semantic and syntactic relations. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 346–353. IEEE Computer Society, 2012.

[4] John J Ahn. *Exile as forced migrations: A sociological, literary, and theological approach on the displacement and resettlement of the Southern Kingdom of Judah*, volume 417. Walter de Gruyter, 2010.

[5] Rosa Alonso Alonso, María Alonso Alonso, and Laura Torrado Mariñas. Hedging: An exploratory study of pragmatic transfer in nonnative english readers' rhetorical preferences. *Ibérica: Revista de la Asociación Europea de Lenguas para Fines Específicos (AELFE)*, (23):47–64, 2012.

[6] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*, pages 196–205. Springer, 2007.

[7] Saima Aman and Stan Szpakowicz. Using roget's thesaurus for fine-grained emotion recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[8] Alexandra Balahur. Sentiment analysis in social media texts. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 120–128, 2013.

[9] Anil Bandhakavi, Nirmalie Wiratunga, Deepak Padmanabhan, and Stewart Massie. Lexicon based feature extraction for emotion text classification. *Pattern recognition letters*, 93:133–142, 2017.

[10] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[11] Joanna Bornat. Remembering and reworking emotions: The reanalysis of emotion in an interview. *Oral History*, 38(2):43–52, 2010.

[12] Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and Joao Sedoc. Modeling empathy and distress in reaction to news stories. *arXiv preprint arXiv:1808.10399*, 2018.

[13] Pete Burnap, Omer F Rana, Nick Avis, Matthew Williams, William Housley, Adam Edwards, Jeffrey Morgan, and Luke Sloan. Detecting tension in online communities with computational twitter analysis. *Technological Forecasting and Social Change*, 95:96–108, 2015.

[14] Pedro Martín Butragueño. The pragmatic rhetorical strategy of hedging in academic writing. *VIAL, Vigo international journal of applied linguistics*, pages 57–72, 2003.

[15] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[16] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[17] David Crystal. On keeping one's hedges in order. *English Today*, 4(3):46–47, 1988.

[18] MIR De Figueiredol Silva. Teaching academic reading: Some initial findings from a session on hedging. In *Postgraduate Conference of the University of Edinburgh*, 2001.

[19] Bart Desmet and VéRonique Hoste. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358, 2013.

[20] Cicero dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[21] Eli Dresner and Susan C Herring. Functions of the nonverbal in cmc: Emoticons and illocutionary force. *Communication theory*, 20(3):249–268, 2010.

[22] Paul Ekman. Basic emotions. *Handbook of cognition and emotion*, pages 45–60, 1999.

[23] Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. The conll-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, pages 1–12. Association for Computational Linguistics, 2010.

[24] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625. Association for Computational Linguistics, 2017.

[25] Viola Ganter and Michael Strube. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 173–176. Association for Computational Linguistics, 2009.

[26] Maria Georgescul. A hedgehop over a max-margin framework using hedge cues. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, pages 26–31. Association for Computational Linguistics, 2010.

[27] CJ Hutto Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*, 2014.

[28] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.

[29] Jonathan Gratch, Ron Artstein, Gale M Lucas, Giota Stratou, Stefan Scherer, Angela Nazarian, Rachel Wood, Jill Boberg, David DeVault, Stacy Marsella, et al. The distress analysis interview corpus of human and computer interviews. In *LREC*, pages 3123–3128. Citeseer, 2014.

[30] MAK Halliday. 3 language as social semiotic. *Language and Literacy in Social Practice: A Reader*, page 23, 1993.

[31] Agnes Weiyun He. Exploring modality in institutional interactions: Cases from academic counselling encounters. *Text-Interdisciplinary Journal for the Study of Discourse*, 13(4):503–528, 1993.

[32] Janet Holmes. Modifying illocutionary force. *Journal of pragmatics*, 8(3):345–365, 1984.

[33] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[34] Zhao Jianqiang, Gui Xiaolin, and Zhang Xuejun. Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, 6:23253–23260, 2018.

[35] Andreas H Jucker. The discourse marker well: A relevance-theoretical account. *Journal of pragmatics*, 19(5):435–452, 1993.

[36] Anna Jurek, Maurice D Mulvenna, and Yaxin Bi. Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4(1):9, 2015.

[37] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665. Association for Computational Linguistics, 2014.

[38] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.

[39] Diederik P Kingma and Jimmy Lei Ba. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*, 2014.

[40] Akrivi Krouska, Christos Troussas, and Maria Virvou. Comparative evaluation of algorithms for sentiment analysis over social networking services. *J Univers Comput Sci*, 23(8):755–768, 2017.

[41] George Lakoff. Hedges: A study in meaning criteria and the logic of fuzzy concepts. In *contemporary Research in Philosophical Logic and Linguistic semantics*, pages 221–271. Springer, 1975.

[42] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[43] Lenore Layman. Reticence in oral history interviews. *The Oral History Review*, 36(2):207–230, 2009.

[44] Moritz Lehne and Stefan Koelsch. Toward a general psychological model of tension and suspense. *Frontiers in Psychology*, 6:79, 2015.

[45] Marc Light, Xin Ying Qiu, and Padmini Srinivasan. The language of bioscience: Facts, speculations, and statements in between. In *HLT-NAACL 2004 Workshop: Linking Biological Literature, Ontologies and Databases*, 2004.

[46] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[47] Ben Medlock and Ted Briscoe. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 992–999, 2007.

[48] Barbara Misztal. *Theories of social remembering*. McGraw-Hill Education (UK), 2003.

[49] Saif Mohammad and Felipe Bravo-Marquez. Emotion intensities in tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 65–77. Association for Computational Linguistics, 2017.

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

[51] Saif M Mohammad. # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics, 2012.

[52] Saif M Mohammad and Svetlana Kiritchenko. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326, 2015.

[53] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[54] Saif M Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499, 2015.

[55] Judith Moyer et al. Step-by-step guide to oral history. 2011.

[56] Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111, 2014.

[57] Dhiraj Murthy. *Twitter*. Polity Press, 2018.

[58] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[59] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Textual affect sensing for sociable and expressive online communication. In *International Conference on Affective Computing and Intelligent Interaction*, pages 218–229. Springer, 2007.

[60] Finn Årup Nielsen. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Heraklion, Crete, Greece, May 30, 2011*, pages 93–98, 2011.

[61] Nugent and Pam. Tension. `https://psychologydictionary.org/tension/`. Accessed: November 9, 2018.

[62] Arzucan Özgür and Dragomir R Radev. Detecting speculations and their scopes in scientific text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1398–1407. Association for Computational Linguistics, 2009.

[63] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

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

[65] Robert Plutchik. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219, 1984.

[66] Diane Ponterotto. Hedging in political interviewing. *Pragmatics and Society*, 9(2):175–207, 2018.

[67] Ellen F Prince, Joel Frader, Charles Bosk, et al. On hedging in physician-physician discourse. *Linguistics and the Professions*, 8(1):83–97, 1982.

[68] Ashequl Qadir and Ellen Riloff. Learning emotion indicators from tweets: Hashtags, hashtag patterns, and phrases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1209. Association for Computational Linguistics, 2014.

[69] Fuji Ren and Changqin Quan. Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing. *Information Technology and Management*, 13(4):321–332, 2012.

[70] Irene Reti. What is oral history. `https://guides.library.ucsc.edu/oralhist`. Accessed: November 9, 2018.

[71] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[72] Harvey Sacks and Gail Jefferson. Lectures on conversation. 1995.

[73] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. *first ESSEM workshop*, 2013.

[74] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *European Semantic Web Conference*, pages 83–98. Springer, 2014.

[75] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Adapting sentiment lexicons using contextual semantics for sentiment analysis of twitter. In *European Semantic Web Conference*, pages 54–63. Springer, 2014.

[76] Deborah Schiffrin. Discourse markers (studies in interactional sociolinguistics, 5), 1987.

[77] Ameneh Gholipour Shahraki and Osmar R Zaiane. Lexical and learning-based emotion mining from text. In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing*, 2017.

[78] Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'connor. Emotion knowledge: Further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061, 1987.

[79] Muller Simon. Discourse markers in native and non-native english discourse, 2005.

[80] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics, 2011.

[81] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[82] György Szarvas. Hedge classification in biomedical texts with a weakly supervised selection of keywords. *Proceedings of ACL-08: HLT*, pages 281–289, 2008.

[83] György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. The bioscope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45. Association for Computational Linguistics, 2008.

[84] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566. Association for Computational Linguistics, 2015.

[85] Chenhao Tan. Hedge words and discourse markers. `https://chenhaot.com/data/hedges.txt`. Accessed: November 9, 2018.

[86] Buzhou Tang, Xiaolong Wang, Xuan Wang, Bo Yuan, and Shixi Fan. A cascade method for detecting hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, pages 13–17. Association for Computational Linguistics, 2010.

[87] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565. Association for Computational Linguistics, 2014.

[88] Christoph Kilian Theil, Sanja Stajner, and Heiner Stuckenschmidt. Word embeddings-based uncertainty detection in financial disclosures. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 32–37, 2018.

[89] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[90] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[91] Paul Thompson. *The voice of the past: Oral history*. Oxford university press, 2017.

[92] Morgan Ulinski, Seth Benjamin, and Julia Hirschberg. Using hedge detection to improve committed belief tagging. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*, pages 1–5, 2018.

[93] Erik Velldal, Lilja Øvrelid, and Stephan Oepen. Resolving speculation: Maxent cue classification and dependency-based scope rules. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, pages 48–55. Association for Computational Linguistics, 2010.

[94] Andreas Vlachos and Mark Craven. Detecting speculative language using syntactic dependencies and logistic regression. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, pages 18–25. Association for Computational Linguistics, 2010.

[95] Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 225–230, 2016.

[96] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. Harnessing twitter “big data" for automatic emotion identification. In *Privacy, Security, Risk and*

Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom), pages 587–592. IEEE, 2012.

[97] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207, 2013.

[98] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[99] Sunmoo Yoon, Noémie Elhadad, and Suzanne Bakken. A practical approach for content mining of tweets. *American journal of preventive medicine*, 45(1):122–129, 2013.

[100] Yazhou Zhang, Dawei Song, Xiang Li, and Peng Zhang. Unsupervised sentiment analysis of twitter posts using density matrix representation. In *European Conference on Information Retrieval*, pages 316–329. Springer, 2018.

[101] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263. Asian Federation of Natural Language Processing, 2017.

[102] Jichang Zhao, Li Dong, Junjie Wu, and Ke Xu. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1528–1531. ACM, 2012.

Appendix A

Booster Words

BOOSTER WORDS				
clearly	obviously	certainly	fact that	show
actually	must	of course	absolutely	always
apparently	assuredly	categorically	compelling	completely
comprehensively	conclude that	conclusively	confirmed	confirmation
considerabley	consistently	conspicuously	constantly	convincingly
corroboratetion	crediblely	crucially	decisively	definitely
definitively	demonstrate	deservedly	distinctively	doubtlessly
enhanced	entirely	especially	essentially	establish
evidently	exceptionally	exhaustively	extensively	extraordinary
extremely	the fact that	find that	found that	firmly
forcefully	fully	strikingly	successfully	fundamentally
genuinely	great	highlight	highly	impossible
impressively	incontrovertible	indispensablely	inevitabley	in fact
manifestly	markedly	meaningfully	necessarily	never
notabley	noteworthy	noticeabley	outstanding	particularly
perfectly	persuasively	plainly	powerful	precisely
profoundly	prominently	proof	proved	quite
radically	really	reliably	remarkablely	rigorously
safely	securely	self-evident	sizablely	superior
surely	thoroughly	totally	truly	unambiguously
unarguably	unavoidabley	undeniabley	undoubtedly	unequivocally
uniquely	unlimited	unmistakablely	unprecedented	unquestionably
uphold	upheld	vastly	vitally	we know
well-known	indeed	no doubt	prove	honestly
mostly	largely	sure	like i said	
nonetheless	mainly	nevertheless	as i say	

Appendix B

Hedge Words

HEDGE WORDS				
suggest	believe	appear	indicate	assume
seem	consider	doubt	estimate	expect
feel	guess	imagine	speculate	suppose
think	understand	imply	presume	suspect
postulate	reckon	infer	hope	rather
slightly	barely	strictly	presumably	fairly
theoretically	basically	relatively	possibly	preferably
slenderly	scantily	decidedly	arguably	seemingly
occasionally	partially	partly	practically	roughly
virtually	allegedly	presumable	possible	probably
likely	apparent	probable	improbable	unlikely
rarely	improbably	unclearly	unsure	sure
chance	unclear	may	might	
shall	should	can	could	
ought	usually	approximately	maybe	
normally	generally	frequently	would	

Appendix C

Discourse Markers

DISCOURSE MARKERS		
however	on the one hand	on the other hand
nonetheless	though	all the same
at the same time	perhaps	although
in spite of the fact that	regardless of the fact that	as a result
therefore	hence	thus
in other words	in a sense	i mean
i think	i believe	i feel
i suppose	i presume	it is my firm belief
to my mind	in my experience	in my understanding
in our opinion	in our view	in our judgement
in my perspective	from my perspective	we think
we believe	i hope	we hope
i suspect	we suspect	i postulate
we speculate	i am afraid	i'm afraid
honestly	actually	anyway
you know	you see	sort of
more or less	not really	no real instance
well	by the way	well , anyway
the thing is	what i mean is	yes , but
i don't want to	i am not going to	i ain't going to
i will not say	i won't say	i will not mention
i don't know	i really do not know	i really don't know
i can not find the word	i can't find the word	a bit
a few	a little	a whole bunch
and all that	and so forth	and so on

DISCOURSE MARKERS		
for the most part	in a way	in part
partial	possible	pretty
seldom	something or other	to a certain
70looks like	sound like	sounds like
pretty good chance	high probability	very high probability
highly unlikely	little support	indicating
very improbable	seems likely	somewhat likely
indicates that	hopefully	appear to
is not known	isn't known	was not known
remains to be investigated	cannot exclude	can't exclude
can't be excluded	can not be excluded	rather probably
nearly certain	very low chance	tends to
somewhat unlikely	lends strong support	fair chance
non-negligible chance	may well occur	doubtful
very possible	very possibly	highly supportive
can not exclude the possibility	can't exclude the possibility	slight evidence
cannot rule out	can not rule out	can't rule out
almost impossible	not probable	extremely unlikely
implicates	thirty percent chance	some possibility
may represent	fairly unlikely	if not
moderate chance	may be associated	according to chance
must be considered	small doubt	not very probable
implying	not likely	somewhat doubtful
slim chance	fifty percent chance	negligible chance
not fully understood	fighting chance	some chance
chances are	plausible	very probable
very probably	small chances	compelling evidence
close to certain	little chance	presumable
looks as	almost certain	highly likely
rather doubtful	no support for	preferentially
most likely	some doubt	must be left open
highly suggestive	probably not	raise the hypothesis
poor chance	not much chance	quite possible
indication	barely possible	highly suspicious
small probability	moderate probability	most possibly
there is a chance	chances are not great	some indication
conceivable	faintly possible	not clear
very low percent	certain hope	reasonable hope
quite unlikely	low probability	small possibility
small chance	no evidence	inconclusive
certain amount	accordingly	even if

Appendix D

Cues

CUES/HESITATION FILLERS		
push	laughs	silence
smile	smiles	smiling
tears	tear	pause
sigh	sighs	ah
mmm	so ...	but ...

Curriculum Vitae

Name:	Jumayel Islam
Post-Secondary Education and Degrees:	The University of Western Ontario London, ON 2017 - 2018 M.Sc. Khulna University of Engineering & Technology Khulna, Bangladesh 2012 - 2016 B.Sc.
Honours and Awards:	Western Graduate Research Scholarship 2017-2018
Related Work Experience:	Teaching Assistant The University of Western Ontario 2017 - 2018 Research Assistant The University of Western Ontario 2017 - 2018