Gauging online and offline public opinion for social media monitoring

Satapathy, Ranjan

2021

Satapathy, R. (2021). Gauging online and offline public opinion for social media monitoring. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/151548>

<https://hdl.handle.net/10356/151548>

<https://doi.org/10.32657/10356/151548>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:21:15 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

Gauging Online and Offline Public Opinion for Social Media Monitoring

Ranjan Satapathy

SUPERVISOR: Erik Cambria

**SCHOOL OF COMPUTER SCIENCE AND
ENGINEERING**

A Thesis Submitted to the Nanyang Technological University in the
partial fulfillment of the requirements of the Degree of Doctor of
Philosophy

06 May, 2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

06 May, 2021

Date

Ranjan Satapathy

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

06 May, 2021

A handwritten signature in blue ink, appearing to read "Erik Cambria", with a stylized flourish at the end.

Date

Erik Cambria

Authorship Attribution Statement

This thesis contains material from 5 papers published and 1 under review in the following peer-reviewed journals / from papers accepted at conferences in which I am listed as an author.

Chapter 2 is published as **Satapathy, R.**, Cambria, E., Nanetti, A., Hussain, A. “A Review of Shorthand Systems: From Brachygraphy to Microtext and Beyond”. Cogn Comput 12, 778-792 (2020).

The contributions of the co-authors are as follows:

- I reviewed the literature and wrote the review manuscript draft.
- Prof Cambria and Prof Andrea suggested the review area and edited the manuscript drafts.
- Prof Hussain proofread the manuscript.

Chapter 3 is published as **Satapathy, Ranjan**, Iti Chaturvedi, Erik Cambria, Shirley S. Ho, and Jin Cheon Na. “Subjectivity detection in nuclear energy tweets.” Computacion y Sistemas 21, no. 4 (2017): 657-664.

The contributions of the co-authors are as follows:

- I designed the algorithm, conducted experiments and wrote the paper.
- Dr. Iti helped me with conducting experiments.
- Prof Cambria suggested topic, edited and proofread the paper.
- Dr. Shirley S. Ho and Dr. Jin Cheon Na proofread the paper.

Chapter 4 is published as **Satapathy Ranjan**, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. “Phonetic-based microtext normalization for twitter sentiment analysis.” In ICDMW, pp. 407-413. IEEE, 2017.

The contributions of the co-authors are as follows:

- I designed the algorithm, conducted experiments and wrote the paper.
- Prof Cambria suggested topic, edited and proofread the paper.
- Dr. Iti helped me with conducting experiments.
- Claudia helped me with constructing the lexicon.

Chapter 5 is published as **Satapathy, Ranjan**, Aalind Singh, and Erik Cambria. “Phonsenticnet: A cognitive approach to microtext normalization for concept-level sentiment analysis.” In International Conference on Computational Data and Social Networks, pp. 177-188. Springer, Cham, 2019.

The contributions of the co-authors are as follows:

- I designed the models, ran the experiments, and wrote the manuscript.
- Prof Cambria participated in discussion, edited and proofread the manuscript.
- Aalind participated in discussions and ran experiments with me.

Chapter 6 is published as **Satapathy, Ranjan**, Yang Li, Sandro Cavallari, and Erik Cambria. “Seq2seq deep learning models for microtext normalization.” In 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2019.

The contributions of the co-authors are as follows:

- I designed the methodology and conducted the experiments and wrote the paper.
- Prof Cambria participated in discussion, edited and proofread the paper.
- Dr. Yang Li helped me with the experiments.
- Dr. Sandro participated in discussions and edited the manuscript.

06 May, 2021
Date

Ranjan Satapathy

Acknowledgments

First of all, I would like to express my sincere gratitude towards my PhD supervisor Prof. Erik Cambria. For the past four years, he has been continuously supportive and encouraging. Without his patient and insightful guidance, I would not have acquired the knowledge and skills to reach this stage.

This thesis is also a dedication to all those reviewers and Editors who painstakingly do all the review process and help the authors to publish. I would also like to thank all my co-authors. I would like to thank my TAC panel members Prof Quek Hiok Chai, Prof Sun Aixin and Prof Andrea Nannetti for their helpful comments and advice. I would also like to thank Prof Nadia, Prof Chng Eng Siong and Prof Dusit Niyato for providing feedback on my thesis.

In addition, I would like to thank Dr. Manoj, Dr. Iti Chaturvedi and Dr. Yang Li for their supportive and inspiring discussions and assistance during my PhD study. My study would not have been complete without their collaborations.

Furthermore, my PhD journey would not end up as a pleasant and rewarding journey without the friendship and support by Nidhi, Dr. Nisha, Aayush, Dr. Guo Qing, Shubham, Aalind and Anoop. My journey is incomplete without mentioning Dr. Nandan, Dr. Ranjan and Madhushala group: Abhishek, Shashank, Arunima, Sakcham and Namz & all the colleagues at IMI, NTU.

Last but not the least, I would like to express my deep gratitude towards my parents, my sister and Prasanna for all the love and relentless support during the critical stages of my life. This thesis is completely dedicated to all of them.

Contents

Statement of Originality	i
Supervisor Declaration Statement	ii
Authorship Attribution Statement	iii
Acknowledgments	v
List of Figures	x
List of Tables	xii
Abstract	2
1 Introduction	3
1.1 Background	3
1.2 Challenges	7
1.2.1 Online	9
1.2.2 Offline	10
1.3 Contributions	10
1.3.1 Motivation for Microtext Normalization	11
1.3.2 Hybrid method for microtext normalization	12
1.3.3 International Phonetic Alphabet (IPA) based method for mi- crotext normalization	12
1.3.4 Deep Learning method for Microtext Normalization	13
1.4 Organization	13
2 Literature Review	14
2.1 Approaches to Microtext Normalization	15
2.1.1 Syntax based approaches	16
2.1.2 Probability based approach	20
2.1.3 Phonetic based Approaches	26
2.2 Grapheme to Phoneme (G2P) Converter	27
2.3 Corpus for microtext normalization	29
2.3.1 Sentiment Analysis	31

2.4	Application of Microtext Normalization	32
2.4.1	Information Retrieval	32
2.4.2	Text Classification	32
2.4.3	Summarization	33
2.4.4	Information Extraction	33
2.4.5	Sentiment Analysis	33
3	Motivation	35
3.1	Prelude	35
3.1.1	Gaussian Bayesian Networks	36
3.1.2	Deep Neural Networks	37
3.2	Reinforced Deep Convolutional Neural Networks	38
3.2.1	Deep Reinforcement Learning	39
3.2.2	RDCNN Framework	39
3.2.3	Input/Output	41
3.3	Evaluation	43
3.3.1	Data collection	43
3.3.2	Data preprocessing	44
3.3.3	Baselines	45
3.3.4	Parameters	46
3.3.5	Effects on Nuclear Energy tweets	46
3.3.6	Effects on Silk Road tweets	47
3.4	Conclusion	49
3.4.1	Subjective Detection on Nuclear Energy Tweets	49
3.4.2	Subjective Detection on Silk Road Tweets	49
4	Unsupervised (Hybrid) Microtext Normalization Framework	51
4.1	Introduction	51
4.2	Framework for unsupervised microtext normalization	52
4.2.1	Data collection	52
4.2.2	Proposed framework	53
4.2.3	SenticNet API for polarity detection	56
4.3	Proposed framework for chatbot	57
4.3.1	Datasets	58
4.3.2	Machine Learning methods	60
4.4	Evaluation at polarity level	61

4.4.1	Soundex	61
4.4.2	Ratcliff pattern-matching algorithm	64
4.4.3	Polarity Detection	64
4.5	Evaluation at chatbot level	65
4.5.1	Dataset Collection and Annotation	65
4.5.2	Effect on execution time	66
4.5.3	BLEU score	66
4.6	Conclusion	67
5	IPA based Microtext Normalization framework for Concept-level Sentiment Analysis	69
5.1	Introduction	69
5.2	Proposed framework	70
5.2.1	How Epitran works?	70
5.2.2	Classification of microtext	71
5.2.3	Polarity Detection	71
5.3	Datasets	72
5.3.1	NUS SMS Corpus	72
5.3.2	SenticNet	72
5.3.3	Normalized Tweets	73
5.4	Results	75
5.4.1	Classification of microtext	75
5.4.2	Comparison of IPA with Soundex	75
5.4.3	Microtext normalization of concepts using IPA	77
5.4.4	Polarity Detection with SenticNet	77
5.5	Discussion and Future Work	78
6	Deep learning based microtext normalization framework	80
6.1	Introduction	80
6.2	Corpus for English Microtext Normalization (CEMt-Norm)	81
6.2.1	Preparation of Corpus	82
6.3	Deep Learning Model	84
6.3.1	Model Variations	87
6.4	Datasets Used	89
6.4.1	LexNorm2015	90
6.4.2	Microtext Lexicon	90

6.4.3	Carnegie Mellon Pronouncing Dictionary	90
6.4.4	NUS SMS Corpus	91
6.4.5	Normalized Tweets	91
6.4.6	Corpus for English Microtext Normalization (CEMt-Norm) . . .	91
6.5	Results and Discussion	91
6.5.1	Evaluation of the corpus (CEMt-Norm)	92
6.5.2	Effect of Masking in the Deep Learning model	93
6.5.3	Sentiment Analysis	94
6.5.4	Discussion	95
7	Summary and Future Work	97
7.1	Summary of Proposed Method	97
7.2	Current Limitations	99
7.2.1	Hybrid method for microtext normalization	99
7.2.2	Deep Learning method	99
7.2.3	IPA based method	99
7.2.4	Corpus for Microtext Normalization	100
7.3	Future Application	100
7.3.1	Beyond Microtext	102

List of Figures

1.1	Microtext analysis as an interdisciplinary field	5
1.2	NLP curve [1]	8
1.3	Polarity Detection of survey feedback (text).	10
1.4	Research path and motivation	11
2.1	Sentiment analysis is a big suitcase of natural language processing (NLP) problems. Sentiment analysis has long been mistaken for the task of polarity detection. This, is just one of the many NLP problems that needs to be solved to achieve human-like performance in sentiment analysis [2]	15
3.1	Bayes Net structure [3] used in the experiment.	37
3.2	Proposed framework for subjective detection	41
3.3	Structure of RNN.	42
3.4	Subjective Detection on Nuclear Energy Tweets	47
3.5	Subjective Detection on Silk Road Tweets	48
4.1	Proposed Microtext Normalization Model	53
4.2	The evaluation process	54
4.3	Overview of the proposed framework for chatbot	58
4.4	Technical details of microtext module used in the framework	58
4.5	Polarity of actual text and proposed model's text accuracy with the human annotated text	65
4.6	BLEU score for the normalized Tweets data	66
4.7	BLEU score for the NUS SMS data	67
5.1	Architecture of the framework	71
5.2	Visualization of string and phonetic distance	76
5.3	Accuracy for polarity detection	78
6.1	Encoder-Decoder framework.	81

6.2	Polarity count in the corpus	85
6.3	Sentiment Analysis accuracy for the corpus on character-level pre-trained model	93
6.4	Architecture for polarity detection comparison.	94
7.1	Application of microtext in computer vision	101

List of Tables

1.1	Common trends in Microtext	6
1.2	Real time sample messages depicting the use of microtext	7
2.1	Research overview of syntax based approaches to microtext normalization	17
2.2	Research overview of probability based approaches to microtext normalization	21
2.3	Research overview of phonetic based approaches to microtext normalization	28
3.1	Final list of Keywords for Nuclear Energy	44
3.2	Final list of Keywords for Silk Road	44
3.3	Ablation study by different models for classifying sentences as Subjective and Objective with F-measure.	45
3.4	6-gram Features learned by the neurons: Nuclear Energy	47
3.5	Features as learned by the neurons: silk road	48
4.1	Output of the proposed model	53
4.2	Sample of the lexicon based approach	55
4.3	Examples of unconstrained language with emotions associated with it .	58
4.4	10-fold Accuracy on NUS SMS dataset	59
4.5	10-fold Accuracy on Normalized tweets dataset	59
4.6	Sample real time tweets/SMS	59
4.7	Results of Polarity Detection Module	62
5.1	Sample Soundex and IPA Encodings with polarities for SenticNet5 . . .	73
5.2	Redundancy in Soundex encoding	73
5.3	Precision, Recall, F1 and Accuracy for each algorithm on different datasets	74
5.4	Sample sentences with their polarities before and after microtext normalization	75

6.1	Sample corpus showing OOV, IV and it's polarity	84
6.2	Accuracy of encoder-decoder models on multiple dataset with mask and without mask	86
6.3	Statistics of the datasets	89
6.4	Accuracy of classifiers on the corpus	92
6.5	Comparison of polarity detection accuracy for different models	95

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ASR	Automatic Speech Recognition
BLEU	bilingual evaluation understudy
CECS	Casual English Conversion System
CEMt	Corpus for English Microtext Normalization
CMC	Computer Mediated Communication
CMUdict	Mellon Pronouncing Dictionary
CNN	Convolutional Neural Network
DNN	Deep Neural Network
ERR	Error Reduction Rate
G2P	Grapheme to Phoneme
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
HRI	Human Robot Interaction
IPA	International Phonetic Alphabet
IV	In Vocabulary
LCS	Least Common Sequence
LR	Logistic Regression
LSTM	Long Short-Term Memory
MDM	Mays, Damerau, and Mercer
MNB	Multinomial Naive Bayes
MT	Machine Translation
NLP	Natural Language Processing
NIV	not-in-vocabulary
OOV	Out of Vocabulary
OBOR	one belt one road
POS	Part of Speech
RNN	Recurrent Neural Network
SCC	spell checking and correction
Seq2Seq	Sequence to Sequence
SMT	Statistical Machine Translation
SMS	Short Message Service
STM	Syntactic Tree Matching
SVM	Support Vector Machine
TTS	text-to-speech

Abstract

This thesis examines the issue of ill-formed text (microtext) and shows its impact on sentiment analysis. The source of microtext is Twitter, WhatsApp, Facebook and other social media platforms. Words or phrases not in their standard language format like “lol” (laugh out loud), “c u 2nite” (see you tonight) and “plz” (please) are called out of vocabulary (OOV) terms. On the other hand, words or phrases in their standard forms, for example, “tonight”, “love” or “talk to you later” are called In Vocabulary (IV) terms. The motivation behind microtext normalization is its growing presence on both online and offline platforms.

Microtext might appear to be an ill-formed text, which requires find-and-replace methods. Traditionally, in a Natural Language Processing (NLP) context, microtexts such as abbreviations and length shortening or extending are normalized, but phonemic and graphemic variation is often ignored. However, the features of microtext can be traced back to social, cognitive and geographic influences. To understand social media text, it is essential to develop a microtext normalization module for traditional and modern NLP models.

The first section of this thesis focuses on a combination of unsupervised learning methods in 2 different chapters 4 and 5. The first method includes lexicon creation to transform the most frequent microtexts and emoticons on social media and applies it to sentiment analysis. This method shows an accuracy increase of 4% when microtext normalization is applied before sentiment analysis. This method is extended with an application to a pre-trained chatbot to show the improvement of the chatbot’s understanding of microtext (social media slangs). This method shows a mean BLEU score of 0.8 for the SMS and Tweet dataset. The second method incorporates a phonetic based method to transform any microtext into its phonetic equivalent before

normalizing it. This method shows an accuracy improvement of 6% for sentiment analysis on SenticNet.

The second part of the unsupervised learning is in Chapter 5, which introduces the IPA-based method for microtext normalization and sentiment analysis. Epitran was used to transform a word to its International Phonetic Alphabet (IPA) equivalent. The International Phonetic Alphabet (IPA) is an alphabetic system of phonetic notation based primarily on the Latin script. There is a significant improvement in microtext normalization task and sentiment analysis over Soundex. The results also show that there is very little redundancy when words are transformed to phonetics in IPA. The accuracy of polarity detection using this method has an improvement over baseline by 5%.

Chapter 6 discusses a simple sequence-to-sequence based Deep Learning method. As the language is dynamic, it is not easy to maintain the lexicon. Deep Learning alleviates the static lexicon with a probabilistic model to transform microtext into its standard form. The findings suggest that a semiotic class of microtext was easy to transform, whereas the phonetic class of microtext was not handled easily. The proposed model improves the pre-trained sentiment analysis model by 4%. We also introduce a corpus for English microtext normalization. The corpus contains microtext-containing sentences along with their correctly spelt sentences and their correct polarity. This dataset can be used to tackle two problems of NLP, i.e., microtext normalization and sentiment analysis. This corpus consists of 38% positive, 39.9% negative and 22.1% neutral. A pre-trained model shows an accuracy of 46.4 % on OOV text for sentiment analysis, showing considerable room for improvement.

In summary, this thesis introduces several approaches to microtext normalization analysis and focuses on sentiment analysis application. Microtext cannot be thought of as a one-dimensional problem to NLP. It encapsulates many factors like regional (geographical), ethnic (national and racial), and social (class, age, gender, socioeconomic status and education), which cannot be ignored given the growing importance of social media in our daily life.

Chapter 1

Introduction

1.1 Background

In current years, sentiment analysis has grown in popularity for processing social media data on online communities, blogs, e-commerce websites, and microblogging platforms [4]. People write their opinions and comments in the form of text on social media. Writing can be construed as a human art or craft (τέχνη/technē) that in historical times involved skills in marking letters, words, ideograms, or symbols. Different instruments were employed depending on the writing materials derived, either from animal parts, minerals, or vegetal¹ used inscribes. The variety of abbreviation system that exists attests to the different resources available such as writing surface. Human civilizations have been developing the art of writing across various continents and over different periods to communicate and share information. To speed up the writing process and save writing materials, the art of shorthand (brachygraphy) came into existence. The art of social media writing does not make an exception in today's era as well. Fast forward to the early 2000s, brachygraphy re-emerged in the form of microtext to enable quicker communication without compromising semantic clarity. This emergence has been steadily improving the efficiency of our writing skills and decreasing the syntactic resemblance.

Brachygraphy is such a system of writing that uses abbreviations or special characters, which came into existence during 1500 A.D. For example, Ancient Greek and

¹<https://en.wiktionary.org/wiki/vegetal>

Latin scripts developed a series of abbreviation methods that were later shared by Romance languages² and Germanic languages³. Similar to brachygraphy, social media is seeing exponential growth in the usage of texting language. It is a convenient and powerful way to express and exchange information in their semi-structured form. For example, Web users write terms or sentences such as “c u 2morrow” (see you tomorrow), “tgif” (thank God it is Friday) and “abt” (about) which may not be found in Standard English but are widely seen in SMS (Short Message Service) messages, Twitter messages, Facebook updates, blogs, discussion forums and chat logs. In this way, computer-mediated communication (CMC) has generated a slang referred to as “microtext” which differs vastly from well-written text [5]. Microtext can also be referred to as the brachygraphy of modern times. Today, the Web has become the primary channel for communication. Humans write about their lives, interests, give opinions and rate products. In some contexts, users write informally without minding spelling or grammar, even deliberately shortening the words or using slang. Web users (Twitter, WhatsApp, Viber and many more) widely use words or phrases such as “lol” (laugh out loud), “c u 2nite” (see you tonight) and “plz” (please). Throughout the thesis, informal words or phrases like “2nite”, “luv” or “ttyl” are called Out of Vocabulary (OOV) and their standard counterparts, for example, “tonight”, “love” or “talk to you later” are called In Vocabulary (IV). This casual usage of language results in a new form of written text which is very different from the well-written text and is especially prevalent in Short Message Service (SMS), chat rooms and micro-blogs. Such a casual style of writing text is also known as texting language (microtext). The normalizing text means converting it to a more convenient, standard form.

There have been significant efforts to produce natural language processing (NLP) algorithms and tools that comprehensively comprehend well-written text. However, these tools cannot be applied out of the box to analyze microtext, which usually contains misspellings, abbreviations and improper grammar. Microtext became one of the most widespread communication forms among users due to its casual writing

²https://en.wikipedia.org/wiki/Romance_languages

³It includes English, before and after the advent of print

style, and colloquial tone [6], in addition to its exponential growth, which is highly perceptible.

Microtext is also referred as “shorthand writing” or “texting language”, i.e., a branch of NLP that focuses on handling “semi-structured” texts. The tasks discovered in this branch overlap considerably with those in more traditional NLP [2], which includes topic detection, summarization, sentiment analysis and classification, question-answering, and information extraction [7]. Microtext in social media is a form of contemporary brachygraphy. Analysis of microtext falls under the interdisciplinary research area, as shown in Figure 1.1, viz., it utilizes knowledge from different domains. Table 1.1 depicts the common trends used to generate microtext. In the

Figure 1.1: Microtext analysis as an interdisciplinary field

context of social data analytics [8], microtext normalization is a necessary step for the pre-processing text before any NLP task is performed. Some of the microtext’s key features are a highly relaxed way of spelling along with the reliance on emoticons and out-of-vocabulary (OOV) words involving phonetic spelling (e.g., b4 for before), emotional emphasis (e.g., coooooool for cool) and popular acronyms (e.g., “otw” for

“on the way”) [9, 5, 10]. The challenge arises when trying to automatically rectify and replace them with the correct in-vocabulary (IV) words [11, 12]. For example, in “tonite” to “tonight”, “ttyl” to “talk to you later”.

Microtext can be perceived as a simple find-and-replace pre-processing [13] task. However, the wide-ranging diversity of spellings makes this solution impractical (e.g., the spelling of the word “tomorrow” includes *tom*, *tmr*, *2moro*, *2moro*, *2morrow*, *2morro*, *2morrow* to name a few variations). Furthermore, given users’ productivity, novel forms that are not bound to orthographic norms in spelling can emerge. For instance, a sampling of Twitter studied in [11] found over 4 million OOV words where new spellings were always created, both voluntarily and accidentally.

Spelling	Ungrammar	Non-standard words
Typo error	Substitution of shorter words	Use of emotions
Deletion of whitespace between two words.	Deletion of pronouns (especially subject)	Abbreviated words
Use of number in words	–	Features from spoken languages

Table 1.1: Common trends in Microtext

“Microtext Analysis” is an interdisciplinary approach (Figure 1.1), i.e., it utilizes knowledge from different domains. Microtext has become omnipresent in today’s world. The term “microtext” was proposed by U.S. Navy researchers [5] to describe a type of written text document that has three main characteristics:

- (a) it is concise, typically one or two sentences, and sometimes as little as a single word (abbreviations like “hru” for “how are you”);
- (b) it is written informally and unedited for quality and thus may use loose grammar, a conversational tone, vocabulary errors, and uncommon abbreviations and acronyms; and
- (c) it is semi-structured in the NLP sense, in that it includes some metadata such as a timestamp or an author.

The design of lexica and language models (L.M.s) are acknowledged to be essential steps in the growth of a microtext normalization framework and entail making some

linguistic decisions that are most often made without quantitative validation. What is indicative of non-standard text is a non-uniform way of writing the identical word types (e.g. “u” instead of “you” in English). While this might appear as a marginal stylistic variation in English, it is a notable feature of less standardized varieties. It is used in most written contexts (literature, newspapers, private correspondence, official documents), in formal and official spoken contexts (education, parliament speeches) and in communications with foreigners. Microtexts have become omnipresent in today’s world. They are notably found in online chat discussions, online forums, user comments on online materials such as videos, pictures, and news stories, Facebook news feeds and Twitter updates, Internet search queries, and short text messages (SMS) as shown in Table 1.2. As most text classifiers are trained on standard words or phrases, the accuracy of classifiers for any L.M. task gets hurt due to the high dependency on data from the Web. In such scenarios, microtext normalization becomes highly important [2].

I’ll meet u b4 lec then...
Whr r u
Hey r v goin out tmr
So u stayin in d hstl ?
R u going to b done anytime soon ?

Table 1.2: Real time sample messages depicting the use of microtext

Microtext is dialect-dependent, i.e., the same set of characters could have completely different meaning in different languages. For e.g., ‘555’ is negative in the Chinese language (because the number ‘5’ is pronounced as ‘wu’ and ‘wuwuwu’ resembles a crying sound) but positive in Thai (as the number 5 is pronounced as ‘ha’ and three consecutive 5s correspond to the expression ‘hahaha’). Table 1.2 depicts some of the real-time sample messages.

1.2 Challenges

In this section, we have discussed both online and offline usage of microtext. The results in this thesis suggest that texting language is moving towards becoming an everyday language. The texting language(microtext) is syntactically shortened words/phrases

with no loss in semantics is crucial to normalize it. Therefore challenges possessed by microtext online and offline is very vital in detecting and normalizing microtext.

Natural Language Processing (NLP) is a scientific field concerned with endowing computing machines to process, interpret or even generate natural language, both in verbal and written forms. Several tasks have been addressed in NLP, such as syntactic analysis, semantic analysis, and discourse analysis in the written form of communication. Assessing the semantic meaning of a text is particularly demanding, which led the research community to address several specific challenges in NLP [2]. Semantics-based NLP concentrates on the inherent meaning associated with natural language text. Semantics is different from syntax as it steps away from keyword spotting or frequency and dives into the contextual meaning of the text. The jump to the Pragmatics Curve [1] will be necessary once semantics-based research reaches human-level understanding.

Figure 1.2: NLP curve [1]

Challenges in microtext normalization are manifold. We found in the literature that the most crucial aspect in microtext generation is an abbreviation (semiotic

class), phonemic resemblance (phonetic class) and hybrid (both semiotic and phonetic). However, with the increasing usage of social media, microtext is omnipresent. Microtext can be divided into 5 classes based on the features it possesses. The classes are as follows:

- (i) *Clipping* : This class ignores or removes the first few consonants, for example: in “em” instead of “them”.
- (ii) *Phonetic* : Words or concepts are transformed to their phonetic equivalents. For example: “luv” instead of “love”.
- (iii) *Acronym* : Longer phrases are written in short forms for quick typing. For example: “t tyl” in “talk to you later”.
- (iv) *Hybrid* : This is a mixture of phonetic and clipping. For example: “2morro” in place of “tomorrow”.
- (v) *Others* : This class has no specific rule. It generally is used when needed to show extreme emotions, like disappointment in “zoh my god” instead “oh my god”.

The microtext exists both online and offline. In our experiments, we found usage of microtext in both social media (Twitter, Facebook posts, Instagram captions and many more.) and offline (survey feedback form). Below we explain it in details:

1.2.1 Online

To understand the opinion of public on “Nuclear Energy” and “Silk Road”⁴, we developed and applied subjectivity detection techniques to distinguish between subjective and objective tweets. The tweets contain noise like URLs, username, hashtags and shortened form of commonly used words. The shortened form was syntactically noisy but semantically meaningful. The model trained on the MPQA dataset did not perform well. The experiment showcased that a trained model on standard English text

⁴Both topics were among the most opinionated topics in South East Asian countries when the experiments were conducted. Hence, these were chosen for experiments.

cannot work efficiently on Web data containing microtext. The experimental setup is detailed in Chapter 3.

1.2.2 Offline

We conducted a survey for social robot’s importance at the workplace [14]. We used an Aspect Based Sentiment Analysis (ABSA) module on this feedback received from customers for setting up a social robot in an Organization. 2.9% of the text gave no result when ABSA was applied. In-depth analysis suggests the presence of microtext in the feedback. The analysis shows the prominence of microtext usage not limited to online.

Figure 1.3: Polarity Detection of survey feedback (text).

Figure 1.3 shows the polarity of the survey responses received. The 2.9% of text had no output, and 10% were neutral when passed through the sentiment model. Further, the manual analysis revealed that there was microtext usage present in both no outputs and neutral labels by the sentiment model.

1.3 Contributions

To address the issues exposed in Section 1.2 this thesis presents the following pieces of work: each focuses on one aspect of microtext normalization and its application in sentiment analysis. An overview of the research path of this thesis is in Fig 1.4.

Figure 1.4: Research path and motivation

In the research path, we have used different methods, namely Deep Learning, Unsupervised Learning and IPA-based methods, to normalize microtext and further show their application on sentiment analysis and chatbot. Further details are enlisted below:

1.3.1 Motivation for Microtext Normalization

We investigate the issue of the presence of microtext on social media. We construct a Gaussian Bayesian Networks based subjective detection framework. We use Reinforcement Learning, and the framework is trained on the MPQA dataset. And then, we use the pre-trained model on Tweets (containing microtext). The results with Tweets show an importance of a pre-processing module to transform OOV terms to IV terms. This chapter also introduces a trained model on a standard English dataset and shows the issues it encounters when there is no microtext module. This is covered in Chapter 3.

The proposed method was documented and published in:

- (i) Satapathy, Ranjan, Iti Chaturvedi, Erik Cambria, Shirley S. Ho, and Jin Cheon Na. “Subjectivity detection in nuclear energy tweets.” *Computacion y Sistemas* 21, no. 4 (2017): 657-664.

1.3.2 Hybrid method for microtext normalization

We prepared a polarised microtext lexicon (word-based). We built a framework with a microtext lexicon and a rule-based method for microtext normalization in Chapter 4 to solve the semiotic class challenges in microtext. The framework treats each word as tokens passed through the rule-based method first and then lexicon to normalize the token. Finally, we show the application of normalization on sentiment analysis. Overall, we found that there is a definite impact of normalization to the existing sentiment analysis framework.

The proposed method was documented and published in:

- (i) Satapathy Ranjan, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. “Phonetic-based microtext normalization for twitter sentiment analysis.” In ICDMW, pp. 407-413. IEEE, 2017.

1.3.3 International Phonetic Alphabet (IPA) based method for microtext normalization

We investigate the problem of phonetic based microtext normalization by transforming the tokens to phonetic subspace. The result discusses the impact of coupling sub-symbolic (phonetics) with symbolic (machine learning) Artificial Intelligence to transform the out-of-vocabulary concepts into their standard in-vocabulary form. We used IPA based method and compared our results with existing phonetic based microtext methods. We built an automatic lexicon method to tackle the microtext normalization method, which can be used on any currently available polarised lexicon. It will act as a link between any knowledge-based framework which is covered in Chapter 5.

The proposed method was documented and published in:

- (i) Satapathy, Ranjan, Aalind Singh, and Erik Cambria. “Phonsenticnet: A cognitive approach to microtext normalization for concept-level sentiment analysis.” In International Conference on Computational Data and Social Networks, pp. 177-188. Springer, Cham, 2019.

1.3.4 Deep Learning method for Microtext Normalization

We tried to solve both semiotic and phonetic class challenge in microtext. In that regards, we propose a seq2seq based Deep Learning method in Chapter 6 for transforming out-of-vocabulary tokens to in-vocabulary tokens. We also introduce a corpus for microtext normalization. We compare different architectures across five available datasets across domains. The results are astonishing as Deep Learning algorithms learn to solve both phonetic and semiotic class of microtext. We also prepared a corpus for microtext normalization (sentence-based). This corpus is for developing and testing microtext normalization systems that cover phonetic variations, abbreviations, and other new variations found on Twitter. This corpus represents a variety of user-generated content (UGC) on Twitter. The results show that Deep Learning methods learn phonetic class transformation better than semiotic class transformation. We further evaluate how the microtext normalization affects sentiment analysis.

The proposed method was documented and published in:

- (i) Satapathy, Ranjan, Yang Li, Sandro Cavallari, and Erik Cambria. “Seq2seq deep learning models for microtext normalization.” In 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2019.

1.4 Organization

This thesis talks about microtext normalization and its application. The rest of this thesis is structured as follows: Chapter 2 presents a literature review on recent progress in microtext normalization, Chapter 3 introduces the motivation behind the microtext normalization module, Chapter 4 introduces the hybrid method for microtext normalization and also shows an application of microtext normalization to chatbot, Chapter 5 extends the rule-based method by incorporating the IPA features and shows automatic construction of polarised IPA based lexicon, Chapter 6 studies the Deep Learning methods, and its impacts on microtext normalization. It also introduces a new corpus for microtext normalization (CEMt-Norm) which can be used for both microtext normalization and sentiment analysis, and Chapter 7 concludes the thesis and proposes future work.

Chapter 2

Literature Review

¹Social media are a combination of interactive computer-mediated technologies that facilitate current society to share information, ideas, career interests and other forms of expression via virtual communities and networks [15]. In [2], authors refer to the task of sentiment analysis as a suitcase of natural language processing (NLP) problems rather than just polarity detection. The semi-structured form of writing makes it difficult for any NLP task, including sentiment analysis, to detect and analyze text. As traditionally sentiment analysis has been trained on standard text, this thesis focuses on microtext normalization as a plug and play module, which removes the inherent problem of non-standard words or phrases usage.

In this chapter, we will review important studies related to microtext. In particular, we will focus on the approaches to microtext normalization. This chapter discusses the approaches to microtext normalization in Section 2.1 as:

- (a) syntax based approach where syntactic units like characters, words or concepts are taken into account to normalize.
- (b) probability based approach where probabilistic models are taken into account to normalize microtext.
- (c) phonetic based approach where phonemic units are taken into account to normalize microtext.

¹The work in this chapter is published in Satapathy, R., Cambria, E., Nanetti, A., Hussain, A., “A Review of Shorthand Systems: From Brachygraphy to Microtext and Beyond”. *Cogn Comput* 12, 778-792 (2020).

Figure 2.1: Sentiment analysis is a big suitcase of natural language processing (NLP) problems. Sentiment analysis has long been mistaken for the task of polarity detection. This, is just one of the many NLP problems that needs to be solved to achieve human-like performance in sentiment analysis [2]

This thesis also introduces to a novel technique of transforming any polarised lexicon to the IPA subspace. This chapter also covers Grapheme to Phoneme (G2P) Converter which discusses the literature study behind IPA and is described in subsection 2.2. Section 2.3 describes the corpora built for microtext normalization due to the increasing usage of social media.

2.1 Approaches to Microtext Normalization

We have classified microtext normalization into syntax based (syntactic), probability based (probabilistic) and phonetic based approaches. Syntax based approaches include rule based, lexical matching and dictionary based methods. On the other hand, probability based approaches include Machine Translation model, Hidden Markov

Model and letter to phoneme methods. The phonetic based approach, include IPA based methods, Soundex methods and phonetic tree-based methods. Each of the section is detailed below:

2.1.1 Syntax based approaches

There are several elements in text that can have syntactic value such as characters, words, concepts or phrases. This subsection introduces syntax based approaches in microtext normalization. Some authors tackle microtext using lexicon as in [16, 17, 18, 19, 20], others use rule based [21, 22, 23, 24], while some others use generator based algorithm [25], finite state machine [26] and contextual disambiguation [27]. The overview of the syntactic based approaches is shown in Table 2.1.

Authors in [16] find the most probable candidate from database for a particular OOV after applying Levenshtein distance. For example, if a word replacement from the database is available, it becomes the most probable solution for that not-in-vocabulary (NIV). In addition to this, authors also incorporated rules for elongated texts like “gooooooooood” instead of “good”.

The authors in [22] investigated properties of lexical transformations as observed within the context of microtext used on Twitter. The data included was scraped from Twitter API. They followed a two-pronged analytic approach:

- (i) Firstly, they conducted a context-free linguistic analysis of words not usually found in everyday English language.
- (ii) Secondly, authors conduct a more in-depth and contextual word-level analysis by first normalizing the noisy message to recover the most probable standard form of the message, noting down the varieties of changes that were produced, and then examining these changes across different general contextual dimensions based on client and geographic location.

The authors in [21] built a normalization system for text messages to be read by a text-to-speech (TTS) engine. They anonymously collected around 20000 sentences

Algorithm	Dataset	Evaluation Metric	Algorithm Rule Types
[16]	SMS and Twitter	Levenshtein distance	letter shortening + database + lexical matching
[22]	Twitter	Lexical transformation	Rule based + lexicon
[21]	English Gigaword corpus + CMU lexicon	Accuracy	Rule based+Maxent classifier+lexicon
[17]	SMS	Top-n	abbreviation lookup + phonetic algorithm
[23]	Twitter	Average errors	Rule based
[24]	Twitter and SMS	Accuracy	Rule + lexicon+soundex
[25]	Twitter, SMS and call-centre data	Precision and Recall	Generator based algorithm
[18]	Twitter	inter-rater agreement	Graph+lengthening
[26]	North Sàmi and Greenlandic Wikipedia	coverage + word error rate	finite state model with weights
[19]	Twitter and SMS	Sentence level Accuracy	database and a user feedback system
[27]	SMS data	Word Error Rate and BLEU score	morphological analysis+ contextual disambiguation
[10]	Twitter and SMS data	F-Measure and Accuracy	orthographic factor, a phonetic factor, a contextual factor and acronym expansion
[20]	multi-lingual dataset	Error Reduction Rate	Aspell, dictionary and word2vec

Table 2.1: Research overview of syntax based approaches to microtext normalization

from volunteers utilizing a Telit GM862-GPS cell phone modem. The approach involves rules to translate English to texting language and automatic abbreviation generation using a maxent classifier.

The authors in [17] propose a three-level architecture to deal with microtext in QA systems, which is described below:

- (i) authors remove noise present in the SMS tokens. They combined Soundex, Metaphone algorithm with a modified LCS algorithm;
- (ii) authors select a semantically similar set of candidate questions and;

- (iii) they build a syntactic tree matching (STM) and WordNet-based similarity to improve the results.

In [23], authors evaluate the performance of two leading open source spell checkers on data taken from the Twitter and measure the extent to which their accuracy is improved by pre-processing with their system. The system “Casual English Conversion System” (CECS) is designed on the basis that errors and irregular language used in casual English found in social media can be grouped into several distinct categories. As the language is dynamic, the usage of microtext has also significantly changed since [23]. Authors in [24] show an application of sentiment analysis with rules and lexicon in addition to the Soundex algorithm. The results show an increase in accuracy for sentiment analysis task by 4%. In [25], the authors convert raw informal texts into their correct grammatical version using a parser-centric word-to-word normalizations method. The authors tie normalization performance directly to parser performance. The framework allows for transfer to new domains with a minimal amount of data and effort. The dataset includes Twitter, SMS and call-centre data. The evaluation is done with the word error rate and BLEU score.

Authors in [18] introduce an automated method which correlates word lengthening to subjectivity and sentiment. The lexicon is built mainly for Twitter and related social messaging networks. They present a method to identify domain-specific sentiment-bearing and emotion-bearing words. The evaluation is conducted with human annotation. This paper misses out other aspects such as the link between the length and the strength of individual words.

In [26], the authors propose a finite-state spell-checking as an alternative to the conventional string based algorithms. North Sámi, Finnish, English and Greenlandic languages are included in this study. For the English language model, authors use the data from [28] and [29], which is a basic language model based on a frequency weighted wordlist obtained from freely accessible Internet corpora such as Wikipedia²

²<http://en.wikipedia.org>

and project Gutenberg³. The language models for North Sámi, Finnish and Greenlandic are drawn from the open-source repository⁴ of finite-state language models. Improvements to Mays, Damerau, and Mercer (MDM) model were addressed in [30] by incorporating a trigram-based real-word SCC method of MDM. The proposed method performs better than the WordNet-based method of [31].

The framework in [27] is based on morphological analysis and contextual disambiguation developed in the context of an SMS-to-speech synthesis system. The evaluation was performed on the corpus of 30000 French SMSes by ten-fold cross-validation [32].

The authors [19] introduced a lexico-syntactic normalization model for cleaning the noisy texts. The channelized database and a user feedback system are utilized for the normalized system. The syntactic analysis of sentences is based on a bottom-up parser which outputs the structure of the input sentence. Output of the parser is passed to prediction module which outputs if a sentence is valid or not. The model captures the user interaction for improving the model accuracy. The user interaction involved here is to acquire the user input through an interface. The user can add an OOV token and its IV token, if it is not present in the dictionary. The noisy input sentence is tokenized into words and parsed through lexicons (means going through lexicons or scan through the lexicon line by line) to transform it into the standard token, following which the output of the lexicon is combined to form a IV sentence. The lexical approaches are critical because they need to contain different forms of each OOV, which tend to evolve rapidly. The lexicons used in the referred paper are Abbreviation lexicon, Non-noisy lexicon, Grapheme lexicon and Phonetic lexicon, which finally goes to user feedback. The non-noisy database consisting of 47000 words are created based on Standard English dictionary. It is used to check whether a word is actually present or not. The grapheme database consists of 654678 words created based on the permutation algorithms. The grammar database consists of 500 Standard English sentences and their part of speech tags. Maintaining the lexicon is a difficult task, with the increasing usage of social media languages.

³The Project Gutenberg website <http://www.gutenberg.org/>

⁴<http://giellatekno.uit.no>

The authors [10] propose a method which combines four factors, namely an orthographic factor, a phonetic factor, a contextual factor and acronym expansion. Authors formulate each factor as channels. The model is the combination of all the four channels with two variations, namely:

- (i) Generic Channel, and
- (ii) Term Dependent Channel.

The authors evaluated the proposed model on Tweets and SMSes at the word level. MoNoise [20] uses dictionary, Aspell and word embeddings to generate candidates. These candidates are then ranked and evaluated at the word level. The evaluation metric authors introduce Error Reduction Rate (ERR). ERR can be interpreted as accuracy normalized for the number of normalised words in the gold standard.

2.1.2 Probability based approach

The last few years have seen a significant shift in speech and language technology as it is being taken over by Deep Learning approaches. In probability based approaches, methods vary from HMM [33, 46, 49, 39, 54], ranking methods with probability [55] and others have used language models [38, 40, 47, 48, 50, 53, 37]. The overview of probability based approaches is shown in Table 2.2 and the details of each method are enlisted below.

The authors [40] propose a two-stage translation method from OOV to IV at word-level. This framework leverages on phonetic information, where non-standard words are translated to possible pronunciations, which are then mapped to standard words. Their results show that this approach enhances the system’s resistance to detect and transform OOV words into their IV form. Authors show their results on SMS and Twitter messages.

The authors in [34] propose a supervised text normalization model based on learning edit operations, which incorporates features from unlabeled data via character-level neural text embeddings. The text embeddings are generated using a Simple Recurrent Network, and this model claims to improve on state-of-the-art with less

Algorithm	Dataset	Evaluation Metric	Algorithm Rule Types
[33]	Celex(English, Dutch, German) + Brulex (French) + CMUdict (English)	word accuracy	expectation-maximization + HMM
[34]	Twitter	character level	Simple Recurrent Network
[35]	Twitter		Feed Forward Neural Networks
[36]	SMS and Twitter Data	Top-N	CRF + Language Model
[37]	WhatsApp messages	word-level	Encoder-Decoder Model
[38]	Twitter dataset	character level	Seq2Seq model
[39]	CELEX and NETalk	Word accuracy (WA) and Syllable break error rate (SBER)	Positional tags capture and Structural tags for Letter-To-Phoneme
[40]	SMS data and Twitter	Top-n	character-based machine translation approach
[41]	Twitter data	Recall and Precision	Monte Carlo training algorithm
[42]	LexNorm dataset	Precision, Recall and F-score	character and word level seq2seq models
[43]	SMS data	Accuracy	Lexicon with Hidden Markov Model
[44]	Chinese micro-blogging site Sina	Precision and Recall	Naive Bayes classifier and a lexicon-based classifier
[45]	SMS corpus and Twitter	F-score	Lexicon+ word similarity + context support
[46]	Pair of misspelled words+ WSJ Penn Treebank	Coverage, F-score	Hidden Markov Model
[47]	WMT'15	BLEU score	word+character level NMT models
[48]	SMS data	Top-N	character-level MT + a language model
[49]	LexNorm1.1 and LexNorm1.2	precision, recall and F-score	Lexicon + Hidden Markov Model
[50]	University of Aix-en-Provence [51] + Catholic University of Louvain [52]	Word Error Rate	Machine Translation+ASR technique
[39]	CELEX and NETalk	word accuracy and syllable break error rate	SVM Hidden Markov Model
[53]	Twitter data	BLEU and NIST score	Orthographic Normalization+Syntactic Disambiguation+Machine Translation
[54]	Chinese-English MT and NUS SMS data	BLEU Score	Hidden Markov Model + beam search decoder
[55]	Twitter dataset	Precision, Recall, F-score	probability distribution with ranking

Table 2.2: Research overview of probability based approaches to microtext normalization

training data and with no lexical resources. The novelty of the paper lies in utilizing less training data and no lexical resource with simple RNN.

In [33], the authors use HMM to convert a letter to phoneme features. The authors used CMUdict, Celex (English, Dutch and German) and Brulex (French). The experiments were conducted with instance-based classifier [56] for predicting phonemes implemented in the TiMBL package [57]. They utilized the HMM technique for post-processing instance-based learning to provide a sequence prediction.

In [36], the authors introduce a two-phase approach for normalization of abbreviations in informal texts. The first phase uses an abbreviation model to generate possible candidates, and the second stage uses a language model to choose the best candidate.

The authors in [35] participated in W-NUT Lexical Normalization for English Tweets challenge⁵ which combines two augmented feedforward neural networks, a flag that acts as an identifier for the words to be normalized and a normalizer, which takes a single token at a time and outputs a corrected version of that token. Authors show that their system achieved an F1-score of 81.49%trailing the second-place model by only 0.26%.

The work done in [41] is a preliminary text normalization technique which uses pre-neural approaches. The authors characterized the relationship between standard and non-standard tokens by a log-linear model, permitting arbitrary features.

Authors in [42] introduce a social media text normalization hybrid word-character attention-based encoder-decoder model. The proposed character-based component is trained on synthetic adversarial data to capture errors commonly found in the online user-generated text. The authors use two encoder-decoder models: a word-based model and a char-based model. First, the tokens are passed through the word-based Seq2Seq model. In contrast, for transforming tokens not found in the word-level model’s vocabulary, it either backtracks to a secondary character-based Seq2Seq model if its confidence is high or copy the source token. Authors evaluate the model on W-NUT [58] dataset.

⁵<https://noisy-text.github.io/norm-shared-task.html>

The authors in [43] constructed an HMM for every word in the standard language, to represent all possible variations of the corresponding *texting language* according to their associated observed probabilities. The model uses a word-level decoder to transform OOV English SMS text to their standard English forms with an accuracy of 89%. The decoder is used for automatic correction as well as information extraction and retrieval from noisy English documents such as emails, blogs, wikis and chat logs. The structure of the HMM is based on linguistic analysis of the SMS data. The authors evaluated the word level model on 1228 different tokens collected from the SMS corpus that does not exist in the training set. The probabilities of every test sample is computed using the Viterbi algorithm.

In [44], the authors propose a hybrid method for multi-class sentiment analysis of micro-blogs, which combines the model and lexicon-based approach. The authors combined the effect of emoticons and Naïve-Bayes classification to divide micro-blogs into three sentiments – positive, negative and neutral. The authors further divide negative opinion into: angry, sad, disgusted and anxious using sentiment dictionaries. They evaluate their algorithm on a Chinese micro-blogging site called Sina.

The authors in [45] proposed a classifier based approach to detect ill-formed words and generate candidate words based on morphophonemic similarity. The most probable candidate for the word to be replaced is chosen based on word similarity and context. The proposed method does not require any annotations and achieves state-of-the-art performance for an SMS corpus and a novel dataset based on Twitter. The authors do not normalize the abbreviations to their standard form.

The authors trained an HMM in [46] with a mixed trigram model, where each state of the HMM is labelled either with a pair of part of speech (POS) tags with a pair made up of a POS tag and a valid dictionary word. The output symbols from the HMM's states are the words observed in the input sentences. The right words that label some of the states represent words in the confusion set of the observed words, collection of misspelt pairs, and a corpus of correct sentences annotated for POS by both WSJ Penn Treebank and Stanford POS tagger [59]. Precision, recall,

and F-score evaluates the system. The training corpus also contains the following additional texts:

- (i) The New American Bible⁶,
- (ii) The Project Gutenberg Encyclopedia, and;
- (iii) four novels by Mark Twain⁷.

Authors in [47] propose a word-character based hybrid Neural Machine translation (NMT) which handles rare word translation as well. The authors compare purely word-based, purely character-based, and hybrid models in which hybrid model scores the highest BLEU and performs better on rare word dataset [60]. The system achieves a new state-of-the-art result with 20.7 BLEU score on English-Czech translation task.

Encoder-Decoder Model [37] to normalize Swiss German WhatsApp messages has also been proposed recently. A multi-lingual framework is still limited by dataset availability but has a lot of scope for the future. In a similar approach, recently authors [38] proposed a seq2seq based encoder-decoder normalization framework in English. They show the improvement in sentiment analysis model by transforming OOV texts into their IV texts. The improvements in the sentiment analysis task show a requirement of the microtext module for the NLP tasks. Such evaluations are needed for other NLP tasks as well. The work done in [48] describes an SMT based system for expanding abbreviations found in the informal text. The SMT follows a two-phase system. The first phase is trained at the character level, while the second phase is trained with an in-domain language model. In this way, the system learns mappings between character-level “phrases” and is much more robust to new abbreviations than a word-level system.

In [49], the authors propose a syllable-based method for tweet normalization to study the cognitive process of non-standard word generation in social media. The authors segment the non-standard words c_i into syllables $s_{c_i}^1 \cdots s_{c_i}^k$, and for standard

⁶The United States Conference of Catholic Bishops website: <http://www.usccb.org>

⁷The Project Gutenberg website: <http://www.gutenberg.org/>

syllable $s_{w_i}^j$ mapping to non-standard syllable $s_{w_i}^j$. The authors calculate the similarity by combining orthographical and phonetic measures. It combines the HMM channel model with four additional characteristics, namely, combination, syllable level, a priori knowledge and general patterns. The paper’s main contribution is that the proposed normalization system relies on unlabeled samples, thereby making it much easier to adapt the method to handle non-standard words in any period of history.

In [50], the authors present a comparative study of methods aiming at normalizing the orthography of French SMS messages. Authors combine MT system with the ASR system to achieve 11% Word Error Rate on a test set of about 3000 unseen messages. The experiments reported by authors use two corpora. The first one has been collected at the University of Aix-en-Provence [51]; it contains approximately 9700 messages. The second corpus contains about 30000 messages [52] and gathered in Belgium by the Catholic University of Louvain. Both corpora contain the message and a reference normalization which has been produced and validated by human annotators.

In [39], the authors trained support vector machine (SVM) for English syllabification. The proposed method improves the accuracy of the letter-to-phoneme conversion. Authors have employed two different approaches to tagging in this paper: one being positional tags [61] which captures where a letter occurs within a syllable; and second being structural tags [62, 63] expresses the role, each letter is playing within the syllable. Word error rate is shown to be reduced by 33% as well. The results reported are based on CELEX and NETtalk dataset. NETtalk and CELEX do not provide the same syllabification for every word. There are numerous instances where the two datasets differ in a perfectly reasonable manner (e.g., forging in NETtalk vs. forging in CELEX).

In [53], the authors propose a syntactic normalization of Twitter Messages. They fed the pre-processed tweets into an SMT model to transform them into standard English. The tool that was used to build this system is Moses [64]. Moses is an SMT package which can produce high-quality translations from one language into another. The authors [54] focus on missing word recall and punctuation amendment.

Authors propose a novel beam-search based decoder for social media text normalization for SMT. The decoder effectively integrates different normalization operations. Authors have created two corpora: a corpus containing 1000 Weibo⁸ messages with their normalizations in Chinese and their English translations; and another corpus which contains 2000 English SMS messages (NUS SMS corpus⁹) in [65].

Authors in [55] prepare a set of (OOV, IV) pairs, ranked by distributional similarity. They use context information to generate possible variant and normalisation pairs and then rank these by string similarity. Evaluation metrics in this work is F-score and word error rate on Twitter dataset.

2.1.3 Phonetic based Approaches

The biggest hurdle comes from the type of texts we are dealing with, which are affected by texting phenomena such as character repetition (for instance, “hiii” for “hi”) or phonetic based character substitution (for instance, “dawg” for “dog” or “lyk” for “like”), to name just a few [66]. In phonetic based approaches, methods like Soundex [24], a phonetic tree matching [67], IPA based [68, 69], LCS based transformation [6]. The overview of the phonetic based approaches is shown in Table 2.3 and the details of each method are enlisted below.

Authors in [67] propose a phonetic tree based microtext normalization on English Wiktionary. The proposed algorithm determines the probable pronunciation of English words based on their spelling. Thus, when the system encounters an out-of-vocabulary (OOV) word, it will determine the most probable in-vocabulary (IV) words with similar pronunciation.

In [24], the authors demonstrated a phonetic-based algorithm to normalize tweets. They show that there is a high (>0.8) similarity index between tweets normalized by their proposed model and tweets normalized by human annotators, in 85.31% of cases. The system enhances the accuracy of the polarity detection module by $>4\%$.

In [68], authors propose a new word-searching strategy based on the idea of sounding out the consonants of the word. The suggested algorithm uses a spelling and

⁸A Chinese version of Twitter at www.weibo.com

⁹Available at www.comp.nus.edu.sg/~nlp/corpora.html

phonetic strategy to extract the base consonant data from both miswritten and real phrases. To serve as a reference, the first phase of their methodology extracts the VS and PS from valid English words. The algorithm also extracts streamlined word reconstructions with vowels re-inserted at the right locations. The second main phase of the methodology is to generate signatures of a new OOV phrase for which data about IPA is unknown. First, the writers accounted for three of the five original word transformations kinds. The third and final step of the methodology is to determine the respective IV words for the OOV phrases. They discover a collection of IV phrases with the same signatures together with their probability of occurrence. The algorithm then applies a number of heuristics in order to score the IV as a match to the OOV word.

In [69] authors propose a cognitive (phonetic) approach to solve the microtext normalization technique. They transform the concepts to their phonemic subspace by using a grapheme to phoneme converter. The proposed framework improves the accuracy of polarity detection by 6% as compared to the earlier model.

In [6], the authors propose a cognitively-inspired normalization technique that integrates different human aspects to normalize the OOV tokens. The method involves improved letter conversion, visual priming, and the resemblance between string and phone. The authors evaluated the system on both words- and message-level using four SMS and Twitter data sets. They also reveal that their method scores more than 90% word-coverage over all the four datasets and the comprehensive word-coverage can be successfully transposed into message-level performance gain.

The authors in [17] propose a new algorithm based on Soundex, Metaphone algorithm with a modified LCS algorithm. They build a syntactic tree matching (STM) and WordNet-based similarity to get semantically similar candidates.

2.2 Grapheme to Phoneme (G2P) Converter

Grapheme-to-Phoneme (G2P) conversion is a method of predicting the utterance of a word or concept given its graphemic or written form. Many Speech and NLP

Table 2.3: Research overview of phonetic based approaches to microtext normalization

Algorithm	Dataset	Evaluation Metric	Algorithm Rule Type
[67]	Twitter	Top-N	Phonetic tree-based framework
[24]	Twitter and SMS	Accuracy	Soundex based framework
[68]	Wiktionary and TheFreeDictionary	Word-searching Strategy	IPA based
[69]	Twitter	Accuracy	IPA based resource generation
[6]	SMS and Twitter data	Word and Message Level Accuracy	Spell Checker+ Character-level HMM
[17]	SMS	Top-n	abbreviation lookup + phonetic algorithm

tasks require a G2P step to convert the orthographic representations to phonemic or phonetic representations [70, 71, 72]. Such tasks include Text-to-speech (TTS), ASR, the training of polyglot phonetic models for (non-speech) NLP tasks, and the implementation of approximate phonetic matching. G2P research focusses on some of the most challenging scenarios in different languages. Though less attention is paid to provide G2P coverage for the full range of languages with a more direct relationship between orthography and phonology. Many such languages require more than merely a mapping table for G2P but can be handled satisfactorily by a more complex rule-based system and certainly do not require a machine learning approach. This is fortunate because many of these are also low resource languages in which sufficient training data are not available but adequate linguistic descriptions do exist. There are a few tools that already occupy this niche. Unitran is a tool that converts orthographic text to WorldBet and X-SAMPA [73]. It is limited, though, in that it does not support Roman scripts and it does not have a mechanism for specifying different behavior for

different languages that use the same script. The following section 2.3 discusses about the different corpora available in the area of microtext normalization.

2.3 Corpus for microtext normalization

According to [74], the rise of microtext is because of the following reasons:

- (i) people are unsure of the correct spellings;
- (ii) it is faster;
- (iii) it has become the norm;
- (iv) people want to represent their dialects and/or demography. The following gives an example of Singlish dialect:

Vanessa: Hi, excuse me, how do I get to Orchard Road?

Nick: Just turn left here and walk straight. Not that difficult lah, just follow the signs.

Often various signals, especially sentiments, are grounded in natural language, and due to the misspelling and other microtext variations present in the data, they become quite hard to pick up. A simple affect word such as “love” might get altered to “luv” or “loveee”, which creates a massive problem for any NLP task of text. A robust finding from the socio-linguistics literature is that non-standard forms that mark social variables, such as regional dialects, are often inhibited informal registers [75]. For example, while the Pittsburgh spoken dialect sometimes features the address term yinz [76], one would not expect to find many examples in financial reports. Similar results for Singaporean dialect, i.e., Singlish for NUS SMS data has been reported in [38]. Abbreviations and phonemic-variants are found as geographically-associated lexical items [77]. Another form of non-standard language is expressive lengthening (e.g., coooolllllll), found [78] to indicate subjectivity and sentiment. In running dialogues - such as in online dating sites: the symbols ‘*’ and ‘\$’ means “Starbucks”. The affect words, e.g., “luv” and “lyk” instead of “love” and “like”

respectively are being used. To detect sentiments for these texts, we need a corpus with sentiment associated with each text. Thus, we develop a corpus for microtext normalization with the focus on sentiment analysis. In this section, we divide the work into Corpus and Sentiment Analysis which is described in subsection 2.3.1. To normalize the microtext or bad language, we need models to be trained on such a corpus. There have been efforts around the research community for creating a corpus for normalizing microtext. A corpus for machine translation [54] has been developed by the name NUS SMS corpus¹⁰. It contains more than 71,000 messages, focusing on Singapore English (Singlish) and Mandarin Chinese.

Authors [79] introduce to an annotated corpus which contains 1164 tweets in the Spanish language for microtext normalization. It contains a set of tweets in Spanish where out-of-vocabulary (OOV) words and their normalized forms is present. Authors in [80] present the first results for the normalization of Turkish and tries to shed light on the different challenges in this area. They divide the normalization problem into 7 categories, namely; letter case transformation, replacement rules and lexicon lookup, proper noun detection, vowel restoration, accent normalization and spelling correction. In [81], authors have developed a dataset for Swiss German microtext normalization. They show their results with the character-level Encoder-Decoder model. The authors shared this corpus as a shared task. Multimodal approaches have also been applied to text normalization [82]. They applied an extrinsic evaluation by evaluating the performance of a part-of-speech tagger, lemmatizer, and named-entity recognizer before and after normalization. Some multi-lingual tools like LexToPlus [83] are also available. It is a dictionary-based parser which detects existing terms in a dictionary for Thai language. It only handles intentional errors caused by the repeated characters at the end of words. The work of [45] is a well-known reference to the task of lexical normalization of tweets, albeit their study focused on English tweets for one-to-one normalization. But, there is some work which focusses on enhancing the current state of art NLP tools such as sentiment analysis by using microtext normalization techniques as a preprocessor to the input text. In [84], authors add Soundex with a

¹⁰github.com/kite1988/nus-sms-corpus

lexicon to normalize the microtext and show the increase in accuracy on sentiment analysis. Similar work [38] to normalize microtext is present. The framework can handle semiotic class normalization, but the accuracy is less for phonemic variation class normalization. The authors [69] propose to tackle phonemic variation class normalization by utilizing the IPA transformation of input microtext. The works have also depicted the change in the sentiment associated with the text before and after normalization. For the machines to understand “bad language”, we need a corpus to train and test the accuracy. The works cover a wide range of microtext normalization corpus but they miss out the application to sentiment analysis. Authors [58] introduce to an annotated Twitter corpus. They randomly sampled 5,200 English. 12 interns and employees at IBM Research Australia were involved in the data annotation. [85] present a Dutch and English dataset that can serve as a gold standard for evaluating text normalization approaches. e focus on both the word and character level and the framework can improve the BLEU score with 20% for both languages.

2.3.1 Sentiment Analysis

Sentiment analysis is the field of natural language processing which studies affective states and subjective information. Recent advances in sentiment analysis is due to introduction of Language Models. Recent advances in neural language modelling (LM) that modelled language to be distributions over sequences of characters [86, 87, 88]. Such models learn internal representations that capture syntactic and semantic properties: even though trained without an explicit notion of word and sentence boundaries, they have been shown to generate grammatically correct text, including words, subclauses, quotes and sentences [89, 90, 38]. More recently, [86] shows that individual neurons in a large LSTM-LM is attributed to special semantic functions, such as predicting sentiment, without explicitly trained on a sentiment label set. [91] describes TASS, an experimental evaluation workshop within SEPLN to foster the research in the field of sentiment analysis in social media, specifically focused on the Spanish language.

2.4 Application of Microtext Normalization

This thesis focusses on sentiment analysis as an application of microtext normalization techniques. Though, the application is not limited to sentiment analysis but rather to any NLP task. Some of the application areas well known are as follows:

2.4.1 Information Retrieval

Information retrieval is the task of examining and retrieving desired knowledge from repositories of data. The first application of information retrieval is in search engines. Search engines aggregate information for a given query. The response to a query part is quite simple though underlying technologies and algorithms which make it possible are complex. However, a user is expected to query proper spelling which is not sustainable in the social media era. Hence, if the query itself is incorrectly spelt then a repository of five billion records and complex systems may not give a proper response for example- if “Alan Turing” becomes “Alen Turnin” if “Captain America” becomes “Capt. Amerika” and if “together” is written as “tgthr”. Review of query logs unveils plenty of misspelt words, wrongly split or merged words, dropping required quotations, unstemmed words, uncommon abbreviations without expansion and a lot of other colloquial variations. These are a serious problem since traditional search engines rely on context and string comparison to retrieve the matching list of information.

2.4.2 Text Classification

The text classification task is the class assignment of documents to a given set of classes. However, usage of microtexts distort the content, and hence, the categorization performance gets affected. Some of the areas where text categorization is useful are call routing, categorization of hand-written client grievances and automated SMS routing.

Study in [92] shows the addition of the artificial noise and ASR noise into a standard text classification dataset had not much degradation in classifier performance.

A generic system for text categorization was proposed based on a statistical analysis in [93]. They assessed their framework on the tasks of categorizing complaints from businesses and abstracts of paper-based German technical reports. Their method achieves 80% accuracy in classifying the texts and is very robust to typo.

2.4.3 Summarization

Selecting important sections or generation of new natural language from the given text comes under the umbrella of summarization of text. Selection of these methods is dependent on statistic, linguist and heuristic techniques. Noisy text poses unprecedented hurdles to the summarization method. The difficulty of summarizing text documents that contain errors as a result of OCR has been studied in [94]. The addition of noise degrades the summarization system. Thus, pre-processing before summarization to normalize the microtext is important.

2.4.4 Information Extraction

The purpose of information extraction is to obtain structured information automatically, categorize them into contextually and semantically well-defined data. The extracted information could be domain-dependent or independent depending upon the application. One of the major tasks in this is named entity recognition and extraction. It involves the extraction of entity names such as “people and organizations, place names, and temporal expressions”. In [95], authors show that noisy text impacts information extraction system’s performance. Authors in [96] studied the performance of named entity extraction under a variety of scenarios involving both ASR and OCR output. Their system was trained on both clean and noisy text to handle noise. The performance degradation is linear as a function of word error rates.

2.4.5 Sentiment Analysis

In recent years, sentiment analysis has become increasingly popular for processing social media data on online communities, blogs, wikis, microblogging platforms, and other online collaborative media [97]. Sentiment analysis is a branch of affective

computing research [98] that aims to classify text into either positive or negative – but sometimes also neutral [99]. Most of the literature is on the English language, but recently an increasing number of works are tackling the multilingual issue [100], especially in booming online languages such as Chinese [101].

Authors in [102] show that *Detection of human stress and relaxation* is crucial for timely diagnosing of stress-related diseases. It performs partial microtext normalization like SCC for missing spelling or repetitive letters which enhances the system performance. Authors in [24] show that microtext normalization plays an important part in polarity detection for tweets.

The above literature shared some common disadvantages. Firstly, the NLP systems are trained on standard language terms and not the texting language. In addition, it misses out the sentiments associated with the real world data by ignoring the affect terms like “luv” for “love”, “lyk” for “like” and so on. The focus of this thesis is to showcase how microtext normalization affects the sentiment analysis task. This thesis tries to address this issue by developing a microtext corpus and some resources like microtext lexicon and PhonSenticNet. This thesis studies the impact of paradigm shift from standard language to texting language on the sentiment analysis and tries to solve the issues standard NLP systems face due to the rise of texting language.

Chapter 3

Motivation

¹Subjective detection aims at distinguishing natural language as either opinionated (positive or negative) or neutral. In [3, 103], authors trained subjective detection model on MPQA dataset. My contribution includes using the pre-trained subjective detection models and try it out on datasets with microtext. It also includes combining Bayesian Deep Convolution Belief Networks with Reinforcement learning to the framework and applying the subjectivity detection framework. This chapter is a motivation to the whole thesis. The application to microtext in this chapter brought the idea to the need of microtext in NLP and shapes the thesis.

In this chapter, we have employed reinforcement learning with dynamic Gaussian Bayesian networks to classify sentences into subjective or objective. Further, we regularize the sentence model using reinforcement learning that updates each weight based on a probabilistic reward function over all the weights. The results show a great significance towards the importance of microtext normalization module.

3.1 Prelude

In this section, we introduce to the notions required to follow the work done in this chapter. First, we introduce dynamic Gaussian Bayesian networks [3] where each node represents a word (w_i) in a sentence (s_k). The neural network learns by minimizing the error function.

¹The work in this chapter is published in Satapathy, Ranjan, Iti Chaturvedi, Erik Cambria, Shirley S. Ho, and Jin Cheon Na. "Subjectivity detection in nuclear energy tweets" *Computación y Sistemas* 21, no. 4 (2017): 657-664.

Notations : N : represents the nodes. T : instances for all the nodes. $X = \{x_i(t)\}_{N \times T}$: $x_i(t)$ represents the i^{th} random variable in t^{th} instance. $\mathbf{a}_i$: set of parent variables regulating variable i .

3.1.1 Gaussian Bayesian Networks

In our experiments we used sequence of sentences which are represented as $s(1), s(2), \dots, s(T)$, where each sentence is tokenized into sequence of words such that $s(t) = (w_1(t), w_2(t), \dots, w_L(t))$, where L is the number of words in sentence $s(t)$. Thus, the probability of a word $p(w_i(t))$ can be given as:

$$p(w_i(t)) = P(w_i(t) | (w_1(t), w_2(t), \dots, w_{i-1}(t)), (s(1), s(2), \dots, s(t-1))) \quad (3.1)$$

A Bayesian Network (BN) is a probabilistic graphical model. The structure of Bayes Net is depicted in Figure 3.1.

$$p(W|S, \boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{w}_i | \mathbf{a}_i, \theta_{i, \mathbf{a}_i}), \quad (3.2)$$

where $p(\mathbf{x}_i | \mathbf{a}_i, \theta_{i, \mathbf{a}_i})$ denotes the conditional probability of node expression $\mathbf{w}_i$ given its parent node expressions $\mathbf{a}_i$, and $\theta_{i, \mathbf{a}_i}$ denotes the maximum likelihood (ML) estimate of the conditional probabilities.

According to Bayes theorem, the optimal Gaussian network S^* is given by maximizing the posterior probability:

$$S^* = \arg \max_S p(S|W) = \arg \max_S p(S)p(W|S), \quad (3.3)$$

where $p(S)$ is the probability of the Gaussian network and $p(W|S)$ is the likelihood of the expression data given the Gaussian network.

Given the set of conditional distributions with parameters $\boldsymbol{\theta} = \{\theta_{i, \mathbf{a}_i}\}_{i=1}^N$, under the assumption that the nodes are multivariate Gaussian. It means, node i can be described with mean μ_i and covariance matrix Σ_i of size $N \times N$. The joint probability of the network can be given as:

$$p(\mathbf{w}_i | \mathbf{a}_i) = \theta_{i, \mathbf{a}_i} \sim \mathcal{N} \left(\mu_i + \sum_{j \in \mathbf{a}_i} (\mathbf{w}_j - \mu_j) \beta, \Sigma_i' \right), \quad (3.4)$$

Figure 3.1: Bayes Net structure [3] used in the experiment.

where $\Sigma'_i = \Sigma_i - \Sigma_{i,\mathbf{a}_i} \Sigma_{\mathbf{a}_i}^{-1} \Sigma_{i,\mathbf{a}_i}^T$ and β denotes the regression coefficient matrix, Σ'_i is the conditional variance of $\mathbf{w}_i$ given its parent set $\mathbf{a}_i$, $\Sigma_{i,\mathbf{a}_i}$ is the covariance between observations of $\mathbf{w}_i$ and the variables in $\mathbf{a}_i$, and $\Sigma_{\mathbf{a}_i}$ is the covariance matrix of $\mathbf{a}_i$. BN does not allow feedback as it is an acyclic graph. But, we know feedback is a primary feature of real-world GN.

3.1.2 Deep Neural Networks

In this framework, the associations among the same layer neurons are not allowed. To train this framework, we calculate the gradient in which the total energy function is represented as E_d concerning the weights in all layers d . This system learns to create a visible layer from each training sample. Next, the Gibbs sampling algorithm is applied to update the hidden layer, which develops the visible layer until convergence.

The continuous state $\hat{h}_j$ of the hidden neuron j , with bias b_j , is a weighted aggregate over all the visible nodes $\mathbf{v}'$ and is given by:

$$\hat{h}_j = b_j + \sum_i v'_i w_{ij}, \quad (3.5)$$

where w_{ij} is the connection weight to hidden neuron j from visible node v'_i . The

binary state h_j of the hidden neuron can be defined by a sigmoid activation function:

$$h_j = \frac{1}{1 + e^{-\hat{h}_j}}, \quad (3.6)$$

Similarly, in the next iteration, each visible node's continuous state v'_i is reconstructed by using the difference between the original and reconstructed visible layer labelled as the vector $\mathbf{v}_{recon}$ using the following:

$$\Delta w_{ij} = \alpha (< v'_i h_j >_{data} - < v'_i h_j >_{recon}), \quad (3.7)$$

where α is the learning rate and $< v'_i h_j >$ is the product of visible unit i and hidden unit j . It will be zero if any of the units are not active at the time of sampling.

To extend the DNN to a DCNN, we partition the hidden layer into multiple groups, Z . Each of the Z groups is connected with a $n_x \times n_y$ filter where n_x and n_y are the width and height of kernel respectively. Assume that, the input has dimension $L_x \times L_y$, then the convolution will result in a hidden layer of Z groups each of dimension $(L_x - n_x + 1) \times (L_y - n_y + 1)$. In a particular group of hidden units, these weights are shared. The energy function of layer l is given as a summation over the energy of individual groups:

$$E^l = - \sum_{z=1}^Z \sum_{i,j}^{(L_x - n_x + 1), (L_y - n_y + 1)} \sum_{r,s}^{n_x, n_y} v'_{i+r-1, j+s-1} h_{ij}^z w_{rs}^l. \quad (3.8)$$

Hence, each layer of a DCNN can be referred to as Convolution RBM (CRBM). We set n_y to 1 as there is no interdependence between 2 nodes embedding given by ϕ_{i_k} and ϕ_{j_k} . In such a model, the lower layers learn abstract concepts and the higher layers learn complex features for clusters of nodes.

3.2 Reinforced Deep Convolutional Neural Networks

In this section, we consider reinforcement learning to regularize the deep convolutional neural networks. The resulting framework is referred to as a Reinforced Deep Convolutional Neural Network (RDCNN). We begin with the description of reinforcement in a single layer neural network and then detail its integration into the complete framework.

3.2.1 Deep Reinforcement Learning

Point-wise probability reinforcements (PPR) are commonly used to robustify NN. Most robust NN inference methods consists in assigning a weight to each instance in order to reduce the influence of outliers [104]. The reinforced maximum likelihood model consists in maximizing :

$$p(y = j|\mathbf{x}) = \frac{e^{\mathbf{x}^T w_j}}{\sum_{k=1}^K e^{\mathbf{x}^T w_k}} \\ -\log p(y|\mathbf{x}, \theta) = -\log p(y|\mathbf{x}, \theta) - \alpha \omega(\mathbf{r}) \quad (3.9)$$

where y is the output class label, $\mathbf{r}$ is the vector or PPRs, α is a reinforcement meta-parameter, and ω is a penalization function. A sparse L1 penalization function is obtained by :

$$\omega(\mathbf{r}) = \sum_{i=1}^n r_i \quad (3.10)$$

which shrinks PPRs towards zero with $r_i \geq 0$ L1-regularized PPRs are given by the closed-form expression:

$$r_i = \max \left(\frac{1}{\alpha} - w_k^{old}, 0 \right) \quad (3.11)$$

where w_k^{old} is the current estimate of the model parameters.

3.2.2 RDCNN Framework

The data pre-processing included removal of stop words (using NLTK²) and punctuation marks from the sentences. Following pre-processing, we determined the part-of-speech for each word in the given sentence with a POS tagger [59]. Subjectivity clues are typically words indicate expression of subjective opinion or judgement - that have been tagged with a prior polarity (i.e. the sentiment they convey in the absence of any context - neutral, positive, negative or both) to identify text passages that are likely to contain subjective expression. The dataset comprises a list of over 8,100 clues. Each clue is a word and its corresponding POS. The frequency of each clue

²https://www.nltk.org/nltk_data/

is computed across the MPQA corpus. MPQA corpus contains both subjective and objective sentences. We filtered out the top 50 frequent clue words from the subjective sentences. Finally, the CNN is collectively trained with both objective and subjective sentences given by Equation 3.4.

This framework is an extension of work [103, 3] where we have added reinforcement learning on top of the architecture in [3]. It consists of $L \times d$ nodes in the visible layer, where L is the number of words in the sentence and d is the number of features for each word. The network consists of first hidden convolution layer of k -gram neurons, second hidden logistic layer of n_h neurons and n_d output neurons. The n_h features expressed at the logistic layer after training form the new input data of T samples. Following this, we add a RNN with time-delays of n_h input nodes and n_r hidden neurons. The n_r features expressed at the hidden neurons after training form the new input data of T samples. Each test sample is used to generate n_h features from deep CNN and n_r features from RNN and finally classified using RNN. Lastly, we use negative log-likelihood probability to regularize the output layer in deep CNN as described by Eq. (3.9) resulting in a Reinforced Deep CNN (RDCNN).

We determine the number of hidden layers by computing the change in visible layer reconstruction error $\Delta\epsilon$ on the training samples. The root mean square error (rmse) is calculated between the input training sample and the reconstructed sample. A new hidden layer is added if there is a significant change in the error $\Delta\epsilon$. The new layer weights are learned, and the reconstruction error is recomputed. The above steps progress until there is no further significant classification error with the addition of layers. In order to determine important features, we apply PCA to individual embeddings of the training instances.

The standard RNN output, $\mathbf{y}(t)$, at timestamp t is given by:

$$\mathbf{y}(t) = f(W_R \cdot \mathbf{h}(t-1) + W_l \cdot \mathbf{n}_h(t)) \quad (3.12)$$

where W_R is the interconnection matrix among hidden neurons $\mathbf{h}$, W_l denotes the weight connecting hidden neurons to the input nodes $\mathbf{n}_h$ and f is the activation function. The output from RNN is a compact representation of a sentence n_r . Figure

Figure 3.2: Proposed framework for subjective detection

3.3 depicts the structure of RNN. All recurrent neural networks have the form of a chain of repeating modules of a neural network. In standard RNNs, this repeating module will have a straightforward structure, such as a single tanh layer. [Source³]

In the final output layer, there is a softmax layer, i.e., each hidden neuron corresponds to either of subjective or objective class. The contrastive divergence method will sample features with high number into the upper layers, resulting in phrases at hidden neurons in the first layer. It follows more significant sentences at the second hidden layer and so on. The algorithm is iterated until there is no substantial change in the weights at the l^{th} layer. Lastly, we train two deep models using English and Spanish MPQA corpus. A sentence is labelled as subjective if either of the two models predicts it as subjective class. The architecture is depicted in Figure 3.2.

3.2.3 Input/Output

This section gives an explanation on Input and Output of each equation with respect to the framework shown in Figure 3.2.

We use word2vec to transform sentences to vectors. It is then passed through CNN layer followed by RNN. The output features of RNN is passed through Fully Connected (FC) layer. The output of FC layer is fed to softmax layer to get the final output. The dimensions of each layer is depicted in the Fig 3.2.

³<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Figure 3.3: Structure of RNN.

Equation 3.4 predicts the probability of a word in a sentence given its neighbours. Here, i is the word index and a_i is the set of two or three parent words. We can use the word vectors for each word to compute this probability. Next, we only consider word sets with high probability.

These are used to initialize the deep neural network in the Figure 3.2. For example, red edges denote one set of word and its parents. The green edges denote a second word and its parent set. Hence, instead of starting with a random network we can set these edges to a high value indicating significant relations.

Next, we pre-train the deep neural network using unsupervised Gibbs sampling. Here Eq 3.5 predicts the state of the hidden neurons h_i in the first layer. Here v_i is the input word vector representation and w_{ij} are the edges to be learned. For Gibbs sampling we use the binary state of each hidden neuron. This is achieved by applying a softmax function in Eq 3.6.

For unsupervised learning we don't have the training labels, hence we reconstruct the input word vectors v_i as a weighted sum over the hidden neurons and the current weights. The difference in the reconstructed word vectors $v_{i_{recon}}$ and the original input v_i is used to update the weights in Eq 3.2.

We are interested in bi-grams such as “cloud computing” or tri-grams such as “not very happy”. Hence, we use convolutional kernels of width 2 or 3, instead of a single neuron h_i . The modified learning for 2 or 3 words together is given by Eq 3.9.

Equation 3.12 shows the RNN used in the proposed framework. The dimension of RNN used is 50.

The final layer of the neuron is a softmax layer with two outputs “subjective” or “objective”. The unsupervised pre-training is followed by supervised fine-tuning with known labels. In this work, we modify the loss function in the output layer by applying an L1 penalty on the loss. To compute the L1 parameter ω in Equation 3.9 we use reinforcement learning.

Eq 3.10 shows that the new loss ω is sum of the individual PPR (point-wise probability reinforcements) for each weight. These are computed using Equation 3.11. Now we simply add the penalty omega to the loss function in the final layer.

3.3 Evaluation

This phase includes data collection from twitter, data pre-processing and testing it on the model.

3.3.1 Data collection

The dataset consists of tweets, which has been collected from twitter streaming API⁴. The crawling started with a single basic keyword, *# nuclearenergy*. It resulted in around 2000 tweets over a period of two days. The collected tweets were parsed and more “hashtags” were collected from each tweet. The hashtags were sorted in terms of frequency. A hashtag was counted only once even if it appeared twice in the tweet. It thereby prevented not giving weightage to a keyword just on the basis of frequency even if it appeared multiple times in one tweet. For example, in the tweet, “The *# nuclearenergy* is for the *# humankind*. *# nuclearenergy* *# nuclear*”, “*# nuclearenergy* occurs twice but our model counts it only once, thereby not giving irrelevant weightage to such hashtags.

219 hashtags were collected with frequencies ranging from 1100 to 10. The cut-off of the frequency was set to 800, which resulted in 1200 tweets. Then the keywords not related to the topics were removed manually. For example: some tweets included *# nuclearenergy* with *# trump* whereas *# trump* is not at all a good candidate of keyword for our dataset. After removing the irrelevant keywords there were 20

⁴<https://dev.twitter.com/streaming/overview>

keywords left. Then, the rest keywords were given input to the twitter API. The output of the twitter were stored and analysed manually to come up with a list of keywords best suited for Nuclear Energy task. Table 3.1 is the list of final keywords for the nuclear energy. Same method was applied for “silk road” topic tweets. Table 3.2 is the list of final keywords for the Silk Road (OBOR).

Table 3.1: Final list of Keywords for Nuclear Energy

save the nukes, nuclear power, nuclear energy
nuclear disaster, why nuclear, nuclear for climate
nuclear power plant, nuclear reactor leak, fukusima and nuclear energy
nuclear energy and green energy

Table 3.2: Final list of Keywords for Silk Road

one belt one road, belt and road initiative
onebeltoneroad, beltandroadinitiative, newsilkroad

It was observed that using “# ” restricted the output tweets with # whereas some people do not enter # tags , thereby losing a major part of opinionated posts. The final set of keywords were then given to twitter API to get streaming tweets. It collected 30,000 tweets for both the topics over a period of two months. Similar method was used to collect tweets for “Silk Road”.

3.3.2 Data preprocessing

The data preprocessing phase followed the following rule in order to remove unnecessary tweets:

- (i) removes usernames (starting with @)
- (ii) urls (<https://www.twitter.com>)
- (iii) Removal of the stop words and punctuation marks

The above mentioned rules resulted in 27,719 tweets for nuclear energy, and 28,000 tweets for silk road which suggests that most of the tweets contained urls, usernames, and stopword. Two additional rules we added to see how many opinionated posts on Twitter contained microtext.

- (i) Removal of emoticons using lexicon⁵ built in Chapter 4.
- (ii) Removal of informal words (such as microtexts), by parsing the text through Aspell dictionary⁶.

After adding last two rules, the resulted in 12,719 tweets for nuclear energy, and 10,000 tweets for silk road. This huge dropout in the number of tweets shows how important microtext analysis is. Dropping microtexts, could lose a major part of opinionated posts. We used the tweets with microtext for our experiments.

	Framework	MPQA	Nuclear Energy	Silk Road
Baseline	CNN	89.4 [3]	55.6	56.3
	GBN+CNN	93.2 [3]	57.4	57.1
	GBN+CNN+RNN	97.2 [103]	58.9	59.4
Proposed Model	GBN+CNN+RNN+Reinforcement	—	59.7	61.5

Table 3.3: Ablation study by different models for classifying sentences as Subjective and Objective with F-measure.

3.3.3 Baselines

The datasets used are MPQA [105] and tweets from Twitter related to nuclear energy and silk road separately. The first baseline is CNN with convolution, dropout and fully connected layer [3]. The trained model is validated on MPQA, Nuclear Energy tweets and Silk Road tweets. The results show that a simple CNN F-measure comes down by more than 30% points.

The second baseline is a Gaussian Bayes Net with CNN. It shows improvement on MPQA dataset, and the tweet dataset. Though, the F-measure with tweets is again around 57%.

The third baseline is an additional RNN layer to Gaussian Bayes Net and CNN. The F-measure for tweets here is an improvement over previous baselines, which is around 59%.

From all the above experiments, it is clear that the models trained on standard language dataset are unable to handle the microtext.

⁵<https://www.netlingo.com/smileys.php>

⁶<https://ftp.gnu.org/gnu/aspell/dict/0index.html>

These baselines are compared against the proposed model in which we added reinforcement learning. The reinforcement learning has improved the subjectivity detection on Tweets at 59.7% and 61.5%.

3.3.4 Parameters

The framework uses 10 fold cross-validation to determine the accuracy of classifying new sentences using the trained CNN classifier. The framework considers only the 6-grams in the test set that shows the highest activation when convolved with the learned kernels. We evaluated the predicted 6-gram kernel vectors using the root mean square error between predicted and the prior word vectors for each 6-gram learned using co-occurrence data. In our experiments, we tuned the following hyperparameters:

- (i) Width of kernels
- (ii) Dimensionality of word vectors

We find that 32% of the tweets are Objective and 68% of the tweets are Subjective.

3.3.5 Effects on Nuclear Energy tweets

The nuclear energy is produced via nuclear splitting rather than chemical burning; it produces baseload electricity with no production of carbon, the criminal element of global warming. Shifting from coal to natural gas is a step toward decarbonizing since burning natural gas generates about half the carbon dioxide of burning coal. We have illustrated the 6-gram features for nuclear energy as learned by two neurons in Table 3.4. Next, in order to assess the quality of prediction, we manually labelled 100 tweets as Subjective and Objective. To merge two different language models, we simply label each sentence as subjective if it is predicted by either English or Spanish model. Table 3.3 shows the F-measure for each of the datasets: MPQA, Nuclear Energy and Silk Road on different models. It can be seen that the the F-measure is around 60% for both Nuclear Energy and Silk Road related tweets.

Figure 3.4: Subjective Detection on Nuclear Energy Tweets

3.3.6 Effects on Silk Road tweets

The Silk Road alternatively called one belt one road (OBOR) primarily refers to the land routes connecting East Asia and Southeast Asia with South Asia, Persia, the Arabian Peninsula, East Africa and Southern Europe. Some neurons in the first layer

Subjective					
1	2	3	4	5	6
researchers	retract	study	suggesting	countries	.
lag	retract	study	suggesting	countries	lag
clean	safety	regulatory	committee	japan	weekend
authoritarian	saltier	nuclear	draws	thiel	bloomberg
saltier	nuclear	draws	thiel	bloomberg	.
Neutral					
1	2	3	4	5	6
ethics	cie	2	regulator	safe	keeping
buzz	request	proposals	looms	ampcronies	theyll
nuclear	draws	thiel	bloomberg	-	-
draws	thiel	bloomberg	-	-	-
nuclear	draws	thiel	bloomberg	-	-

Table 3.4: 6-gram Features learned by the neurons: Nuclear Energy

Table 3.5: Features as learned by the neurons: silk road

Subjective 1	china, increase, influence, regional, hegemon, la	leaders, enemy, pakistan, please, pakistani, amp	baluchistan, progressing, passing, day, cpec, la	baluchistan, progressing, passing, day, cpec, la	land, routes, exceeded, usd1, trillion, 15
Subjective 2	pakistan, china, pakistan, strengthen, economic, ties	nytimes, cchristine-fair, baluchistan, slowly, progressing, cpechopeful	anupampkher, baluchistan, slowly, progressing, cpechopeful	should, questions, govt, 1, pakistan, prepared	celluar, dezigns, colors, messages, meanings
Subjective 3	pakistan, chinese, firm, tracts, land, cpec	pakistan, chinese, firm, tracts, land, cpec	pakistan, chinese, firm, tracts, land, cpec	pakistan, china, pakistan, strengthen, economic, ties	cpec, holy, cow, china, pakistan, la
Objective	final, results, pakistan, batting, problem, rounders	cpec, pakistan, china, pakistan, strengthen, economic	rimshamengal, baluchistan, progressing, passing, day, cpec	years, dostain, khan, jamal-dini, chairman, la	moeednj, made, pakistan, economist, will, call

learn neutral and some learn subjective features. We have listed the 5-gram features for silk road tweets as learned by the neurons in Table 3.5.

Figure 3.5: Subjective Detection on Silk Road Tweets

3.4 Conclusion

Subjectivity detection, in particular, is a crucial subtask of sentiment analysis since most polarity detection tools are optimized for differentiating between positive and negative text. Subjectivity detection, hence, assures that factual information is filtered out, and only opinionated information is passed on to the polarity classifier. In this framework, we have trained the proposed model on review datasets, but we have tested it on tweets. The data pre-processing phase removes a significant part of the noise in the collected data. The results are compared with baseline model and emphasizes the importance of microtext normalization in social media. The baseline is trained and tested on MPQA dataset. On the other hand, the tweets used for testing are on topics related to “Nuclear Energy” and “Silk Road”. The results show that models trained on standard dataset are incapable in detecting OOV terms which are predominant on social media. The informal words in sentences further deteriorates the performance of the model shown in this chapter.

3.4.1 Subjective Detection on Nuclear Energy Tweets

Three major reasons associated with the results of neutral tweets as seen in Figure 3.4 for Nuclear Energy tweets:

- (i) No nuclear disaster occurred during the data crawling phase.
- (ii) People only share news for nuclear energy unlike other topics.
- (iii) Pre-trained models trained on standard text tend to ignore and/or confuse OOV terms predominant in social media.

3.4.2 Subjective Detection on Silk Road Tweets

Major reason associated with the results of neutral tweets as seen in Figure 3.5 for silk road tweets:

- (i) People tend to share only news for nuclear energy unlike other topics.

The 69.5 % of subjective tweets for silk road (OBOR) shows that Twitter is an important source for opinion mining. The results show the necessity of a microtext analysis module in order to retain tweets which are filtered as noise in the preprocessing phase. The future work includes constructing a dataset with OOV terms to train subjective detection modules for better subjective detection.

The next chapter 4 introduces to a hybrid approach which incorporates rule based with a lexicon to normalize microtext. This is a preliminary task for better understanding the OOV terms and their relationship to IV terms.

Chapter 4

Unsupervised (Hybrid) Microtext Normalization Framework

4.1 Introduction

¹The work done in Chapter 3 makes it clear the need of microtext normalization task as an intermediate module between raw data and any pre-trained model. This chapter proposes a novel framework to deal with microtext normalization in Twitter. Firstly, a large lookup table was automatically assembled with 1,727 OOV words from a variety of Internet sources, generating as much spelling variations as available online for each OOV word. Then, a set of rules was defined to recognize similar OOV words based on the corpus created. Experimental results show that the proposed framework automatically normalizes microtext and increases the accuracy of word level sentiment analysis.

In order to interact with and understand humans, machines need to understand the different unconstrained ways people write and share opinions. In the current chapter we have also incorporated the model into a chatbot architecture and shows an increase in understanding capability of unconstrained language by the chatbot. In our experiments we have used pre-trained retrieval based chatbot.

This chapter provides solutions to semiotic class² of microtext normalization. In this chapter we propose a word level microtext normalization module which incorpo-

¹The work in this chapter has been published in **Ranjan Satapathy**, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. “Phonetic-based microtext normalization for twitter sentiment analysis.” In 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 407-413. IEEE, 2017.

²<https://www.britannica.com/science/semiotics>

rates rule based approach with a lexicon and is evaluated on SenticNet for polarity detection. This rule based with lexicon approach constitute the proposed framework. It shows the effect on SenticNet to detect polarity before and after normalization. Section 4.2 introduces to the proposed Microtext Normalization model; Section 4.3 presents the application of our proposed framework in chatbot; Section 4.4 and 4.5 gives evaluation on polarity and chatbot respectively; finally, Section 4.6 provides concluding remarks.

We showcase the unsupervised microtext normalization application in two ways :

- (i) sentiment analysis in section 4.2;
- (ii) chatbot in section 4.3.

4.2 Framework for unsupervised microtext normalization

Here, we will discuss the model developed in order to normalize microtext. This is a first attempt to normalize microtext in social media. We crawled twitter and website comments' section and built a lexicon on top of that to increase the word-coverage. This study was conducted in three stages to address the normalization challenges of microtext for sentiment analysis tools. As shown in figure 4.1, stage 1 focused on data collection from Twitter; stage 2, assembled a large lookup table, while defining a set of rules to recognize similar OOV words. Finally, stage 3 built on the outcome of the previous stages and applied SenticNet API³ for the sentiment classification.

4.2.1 Data collection

Tweets are known to be noisy and frequently mixed with linguistic variants. The pre-processing tasks that are commonly used include lowercase conversion, handling of stop words as well as removal of URLs and Twitter's usernames found in the content (in the format of username). In this work, Twitter is chosen as the source of data, whereby approximately 4,000 tweets were crawled for normalization purposes.

³<https://sentic.net/demos/#polarity>

These tweets were manually annotated by the three reviewers (Cohen’s kappa = 0.78) in terms of polarity (positive or negative). Whereas, the normalization consisted of transforming OOV words to IV words considering their respective context. The normalization task was carried out by a master’s student. Then, another set of three reviewers were asked to validate the normalisation task. Only tweets where all three agreed were taken for experiments. Table 4.1 depicts the proposed model’s output.

Table 4.1: Output of the proposed model

OOV form	IV form
nt vegan nemor lol	not vegan anymore laughing out loud
how are you so gor- geousss omg omg	how are you so gor- geous oh my god oh my god
pls start sayin dis prayer	please start saying this prayer

4.2.2 Proposed framework

From June to November 2016, the term “English Internet Slang” together with keywords such as “acronyms”, “abbreviations”, “expressions”, “vocabulary”, “words’ and “phrases” were entered in search engines to retrieve microtext terms. The sources used

Figure 4.1: Proposed Microtext Normalization Model

included NetLingo⁴, MakeUseOf⁵, Acronyms and Slang⁶, and Internet Slang⁷. These sources contain an alphabetical list of terms frequently used online along with their respective definitions. Currently, the microtext lookup table includes a total of 1,727 English OOV words and their IV words.

4.2.2.1 Lexicon Based Approach

In order to construct the lookup table, the sources have been crawled by an automatic system to build a noise-free lexicon. Next, IV words have been removed as they are rarely used in the context of microtexts.

Figure 4.2: The evaluation process

It is important to note that, while the lookup table brings an advantageous tool to understand the divergent OOV words and link them to their respective IV words, different OOV words might have the same corresponding IV word (e.g., the word “because” can be written as “BC” or “bcoz”). Keeping all the variations of a OOV is not practical. In the lexicon, we have kept the variations which were very frequent over sms and tweets. Table 4.2 shows a sample of the generated lexicon which has been used in this chapter.

Subsequently, a set of rules was defined to recognize similar OOV words including:

- (i) positional knowledge of the OOV words;

⁴<http://www.netlingo.com/>

⁵<http://www.makeuseof.com/tag/30-trendy-internet-acronyms-slang-need-know-fit/>

⁶<http://acronymsandslang.com/>

⁷<http://www.internetslang.com/>

Table 4.2: Sample of the lexicon based approach

OOV form	IV form	Polarity
ygbk	you gotta be kidding	negative
ykwim	you know what i mean	neutral
ykw	you know what	neutral
ulkgr8	you look great	positive
ymak	you may already know	neutral
ymal	you might also like	positive
ymbkm	you must be kidding me	negative
ynk	you never know	neutral
uok	you ok	neutral

(ii) handling emoticons and numerical digits together, and

(iii) handling repetitions.

4.2.2.2 Rule Based Approach

(i) **Positional knowledge of the OOV:**

(a) It was observed that whenever ‘2’ was present at the end of a text, it meant ‘too’. So, it was converted to ‘too’. For example, **me 2** is converted to **me too**.

(b) If the next word is plural (determined by lemmatizer), then the current symbol is searched in the numerical digit list and converted accordingly. For this purpose, we tokenize the input text and use tags ‘NNS’ or ‘NNPS’ only. If there is a change in the tokens, that suggests a plural token. For example, **2 cars** is converted to **two cars**. lemmatization looks beyond word reduction and considers a language’s full vocabulary to apply a morphological analysis to words. For example: the lemma of “cars” is “car” and the lemma of “mice” is “mouse”.

(ii) **Handling emoticons with numerical digits:** The work included an emoticon list of approximately 500 emoticons⁸ to further refine the rules. For example,

⁸<https://www.netlingo.com/smileys.php>

‘3’ in ‘<3’ is not a numerical digit but rather part of an emoticon (heart to be specific). Therefore, whenever such texts are encountered, the proposed model classifies it as emoticon and handles it accordingly.

(iii) **Handling repetitions in the text:**

- (a) Frequency of the characters were taken into consideration. If a character appears more than twice in a word, then it was classified as a repetition.
- (b) If the character is not present in the emoticon list then it is normalized. For normalization of repeated characters there is no standard window size to shrink the repetitions. The model shrinks it to two characters and then passed through a dictionary. Dictionary determines the nearest word to the shrunk one using Levenshtein Distance [106] and is replaced by it. For example, ‘coool’ is normalized to ‘cool’ and ‘goood’ is normalized to ‘good’, whereas it could mean ‘god’ in some context. The study of tweets suggested that ‘god’ normally appeared as ‘omg’ meaning ‘oh my god’ which is present in the prepared lexicon.

Both the approaches together constitute the proposed unsupervised framework for microtext normalization. The output of this framework is passed to the SenticNet API to determine the polarity of the text. This polarity is checked against human annotated polarised text. The polarity detection is explained in the next section.

4.2.3 SenticNet API for polarity detection

SenticNet API is chosen as a polarity classifier for evaluation purposes in this chapter. SenticNet [107] helps in tagging sentences to their polarity with the semantics and sentics information. In this way, most concept expressions can be caught by SenticNet. While SenticNet can be used as a bag-of-concepts based sentiment lexicon, the right direction is to use the knowledge base for polarity detection in conjunction with sentic patterns [108]. The linguistic patterns used here are called sentic patterns. These patterns indicate polarity by accumulating the affective information within the sentence which is broken down into concepts.

As we saw the application of microtext normalization on pre-trained polarity models. The next section 4.3, introduces to the microtext normalization application for chatbots. As chatbots are becoming very popular in today's e-commerce and other platforms, in the next section it is shown how microtext normalization module is important for chatbots as well.

4.3 Proposed framework for chatbot

Composite nature of the NLP problem is addressed by the suitcase model [2]. In this regard, microtext module is the first step. The syntactics layer aims at preprocessing text so that informal text is reduced to human readable format (any language), inflected forms of verbs and nouns are normalized, and basic sentence structure is made explicit. As social media is dynamic in nature, a rule based model is not exhaustive to handle such events as new short forms evolve rapidly.

The framework shown in Figure 4.3 has a microtext module which acts like a filter for pre-trained chatbot module [109] to understand microtext. This chatbot is a pattern recognition based chatbot. The answers are given only if pattern is triggered. After adding microtext module, the input text from sources like chats and messages (where microtext is predominant) would also be understood by chatbot (Table 4.3). In this framework, an input sentence is passed through microtext module. The technical details of the microtext module is in Figure 4.4. The microtext modules consists of a binary classifier to detect if the input sentence contains OOV word. If there are no OOV words in the sentence, it is passed on to the chatbot module to get a response. If binary classifier detects an OOV word, then the whole sentence is flagged and passed on to microtext module. The module consists of lexicon and rules mentioned in Section 4.2.2. It is important to note that, the lexicon built is a lookup table which brings an advantageous tool to understand the divergent OOV words and link them to their respective IV words. It is also important to note that different OOV words might have the same corresponding IV word.

The binary classifier employs a n-gram based several machine learning techniques as shown in Table 4.4 and Table 4.5. Table 4.2 shows the sample lexicon which helps

Figure 4.3: Overview of the proposed framework for chatbot

Figure 4.4: Technical details of microtext module used in the framework

social robot’s NLP module understand the social media language. The proposed framework is shown in Figure 4.3. The text is passed through microtext module for normalization and then passed on to chatbot’s NLP module.

Microtext	Meaning	Polarity
aah	Fright	NEGATIVE
aha	Understanding, triumph (can also be used as “ahh”)	POSITIVE
duh	Expresses annoyance over something stupid or obvious	NEGATIVE
haha	Regular laughter	POSITIVE
wow	Impressed, astonished	POSITIVE

Table 4.3: Examples of unconstrained language with emotions associated with it

4.3.1 Datasets

This section discusses the datasets used in the evaluation of proposed framework. The unconstrained style of speech comes under the umbrella of microtexts. The two classes in both the datasets are equally distributed. In lexical normalization accuracy

Table 4.4: 10-fold Accuracy on NUS SMS dataset

Classifier	10-fold (%)
NuSVC	85.14
<i>Linear SVC</i>	92.95
Multinomial Naive Bayes	89.92
Bernoulli Naive Bayes	89.24
Logistic Regression	91.05
SGD	91.42

Table 4.5: 10-fold Accuracy on Normalized tweets dataset

Classifier	10-fold (%)
NuSVC	84.2
<i>Linear SVC</i>	87.4
Multinomial Naive Bayes	81.8
Bernoulli Naive Bayes	82.7
Logistic Regression	84.9
SGD	83.4

is defined as number of sentences which are exactly matching after normalization. Further we will discuss the datasets and the machine learning models.

4.3.1.1 NUS SMS Corpus

This corpus (Table 4.6) has been created from the NUS English SMS corpus⁹, the authors [54] randomly selected 2,000 messages. The messages were first normalized into standard English and then translated into standard Chinese. For our evaluation purposes, we only used the actual messages and their normalized English version (leaving out their Chinese counterparts).

Social media texts	Expanded forms
I'll meet u b4 lec then...	I will meet you before the lecture then.
Where r u	Where are you
Hey are we going out tmr	Hey are we going out tomorrow
So u stayin in d hostel ?	So you are staying in the hostel ?
R u going to b done anytime soon ?	Are you going to be done anytime soon ?

Table 4.6: Sample real time tweets/SMS

⁹<http://github.com/kite1988/nus-sms-corpus>

4.3.1.2 Normalized Tweet Dataset

Authors in [24], built a lexicon which consists of real time tweets and their IV counterparts. The dataset is available on request. We demonstrate our results on this dataset utilizing the context within the sentences by including unconventional written sentences together with the transformed standard ones.

4.3.2 Machine Learning methods

For our binary classifiers we employed machine learning methods, to differentiate between OOV and IV in a sentence. If a sentence contains one OOV, it is classified as OOV and passed through the microtext normalization module. Next we will discuss each of the method briefly:

4.3.2.1 Logistic Regression (LR)

Logistic regression models the probabilities for classification problems with two possible outcomes, in our case it is OOV or IV. It's an extension of the linear regression model for classification problems. Hence, instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1.

4.3.2.2 Naive Bayes (NB)

In the experiments, we have used both Multinomial NB and Bernoulli NB. The Multinomial NB will classify a document based on the counts it finds of multiple keywords; whereas Bernoulli NB can only focus on a single keyword, but will also count how many times that keyword does not occur in the document.

4.3.2.3 Support Vector Machine (SVM)

In the experiments, we used two variations of SVC i.e., Linear SVC and NuSVC. Nu-Support Vector Classification. Similar to SVC but uses a parameter to control the number of support vectors. Linear SVC is a linear kernel SVC. Also, "NuSVC" and "Linear SVC" are implementations of support vector machines (SVM) which is commonly used to refer to the technique itself.

4.3.2.4 Stochastic Gradient Descent (SGD)

SGD is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function. Thus, SGD randomly picks one data point from the whole data set at each iteration to reduce the computations enormously.

4.4 Evaluation at polarity level

This section demonstrates the model’s results for the tweets normalization. Soundex [110] and Ratcliff/Obershelp pattern-matching algorithm have been incorporated into the model as evaluation techniques. Figure 4.2 is used to analyze the evaluation procedure.

4.4.1 Soundex

Soundex often appears in genealogical contexts and it has mainly been associated with the U.S. Census. As follows, it is used in applications involving searching individuals’ names (e.g., air reservation systems), as well as other tasks presenting typing errors due to phonetic similarity. Soundex is largely considered as a phonetic algorithm, used to group similar sounding letters together and assign each group a numerical number. The main goal of this technique is to use homophones for encoding text with numerical indexing; as a result the numerical representation can be easily matched with other similar sounding characters having same numerical code. This results in retrieving a list of words that are pronounced similarly with very little variation in their homophones [111, 112]. Thus the lexicon based approach and rule based approach represent a significant step further for future work and to solve current normalization challenges. Soundex hash value is calculated by using the first letter of a name and converting its consonants to digits through a simple lookup table. Soundex has been chosen for the purpose of microtext normalization due to its following advantageous features:

- (i) It is simple and intuitive to operate

Table 4.7: Results of Polarity Detection Module

Tweet	Actual Tweet	Human Annotated	Model's output
dance parties with hayley and levi are probs my fav thing ever !	positive	positive	positive
today was fun someone could of made it better lol me gelo and alex was really clowning	positive	positive	positive
ty 4 following tattered website # facebook # itunes & dwnld r free # android app	negative	positive	positive
so ima keep on drinking cause i luv this shit	positive	positive	positive
dads gonna buy stronger chillis so i can do it against one of my siblings lewl	positive	negative	negative
tent erected in the middle of a main road in phase 4 in mamelodi , we knw they r grieving bt not in main road.	positive	negative	negative
you should check out in wan chai. considered one of the best burgers in hk.	negative	positive	negative

- (ii) The straightforwardness of the code enables to modify it according to the purpose
- (iii) The processing time is fairly short
- (iv) It has a high tolerance for discrepancies in words that sound very similar or are identical

Furthermore, the pseudo-code of the algorithm used can be depicted as follows:

- (i) Retain the first letter of the word.
- (ii) Change all occurrences of the following letters to '0' (zero):
- (iii) 'A', 'E', 'I', 'O', 'U', 'H', 'W', 'Y'.
- (iv) Change letters to digits as follows:
 - (a) B, F, P, V $\rightarrow$ 1
 - (b) C, G, J, K, Q, S, X, Z $\rightarrow$ 2
 - (c) D, T $\rightarrow$ 3
 - (d) L $\rightarrow$ 4
 - (e) M, N $\rightarrow$ 5
 - (f) R $\rightarrow$ 6
- (v) Remove all pairs of consecutive digits.
- (vi) Remove all zeros from the resulting string.
- (vii) Pad the resulting string with trailing zeros and return the first four positions, which will be of the form $\langle \text{uppercase letter} \rangle \langle \text{digit} \rangle \langle \text{digit} \rangle \langle \text{digit} \rangle$.

Although Soundex was initially developed for normalizing surnames, the proposed model has proven that this technique is effective with tweets as well.

4.4.2 Ratcliff pattern-matching algorithm

The proposed model uses the Ratcliff/Obershelp pattern-matching algorithm to assess how similar two one-dimensional patterns are, particularly, the human annotated and the proposed model texts taken from Twitter. Since these text strings are one dimensional, this algorithm is able to return a percentage to show how alike these two strings are. The formula for Ratcliff pattern-matching algorithm determines the similarity of two strings S_1 and S_2 , calculating twice the number of matching characters K_m divided by the total number of characters of both strings. Thus, the main advantage of this pattern-matching algorithm is that it enables the recognition of matches in substrings quickly and easily.

4.4.3 Polarity Detection

Table 4.7 shows a sample output of the polarity detection module for each of the text. It can be observed from the table how model's output polarity aligns mostly with the human annotated tweets' output polarity, thereby promising simple yet robust system for text normalization. A dataset of approximately 4,000 tweets (randomly selected) was constructed and annotated by three reviewers (Cohen's kappa = 0.78) in terms of polarity (positive or negative). The framework takes the actual tweet, human annotated tweet and proposed model's output to the Sentic API polarity module. Human annotated tweet was taken as a benchmark and the accuracy of actual and model's output is compared against the human annotated tweet's polarity. The application of microtext normalization module results in an accuracy increase from 77.47% to 81.59% in terms of polarity detection shown in Figure 4.5. This method shows that our model helps to increase the polarity detection method by 4%. So, the proposed system not only normalizes the tweets but also increases the accuracy of polarity detection.

Figure 4.5: Polarity of actual text and proposed model’s text accuracy with the human annotated text

4.5 Evaluation at chatbot level

The Bilingual Evaluation Understudy score (BLEU) score is used to evaluate the sentences’ similarity. The Sentence BLEU¹⁰ is used to score the similarity between normalized sentences output from the proposed framework and human annotated sentences.

4.5.1 Dataset Collection and Annotation

The dataset used in this experiment was collected from Twitter streaming API¹¹. The data was preprocessed using following rules:

- (i) removal of usernames (starting with),
- (ii) urls (eg., <https://www.Twitter.com>),
- (iii) Removal of punctuation marks,

¹⁰https://www.nltk.org/_modules/nltk/translate/bleu_score.html

¹¹<https://developer.twitter.com/en/docs>

4.5.2 Effect on execution time

The results in Table 4.4 and Table 4.5 shows different algorithms applied on both the datasets. The models are trained on the unigram features as microtexts work at the word-level. The result shows **Linear SVC** to be the best binary classifier for both the datasets. The binary classifier reduces the time complexity and makes the overall framework run faster. The framework ran on Python 3.5 on Ubuntu operating system with 64 GB RAM and 30 GB 1080 T_i Nvidia Graphics. It took 11.4 seconds to run without the binary classifier and only 8.8 seconds with the binary classifier. The binary classifier works as a filter, which reduces the overall execution time of the framework.

4.5.3 BLEU score

BLEU score [113] is employed as an evaluation task. It is used to evaluate the quality of text which has been machine-translated from one natural language to another. It's strength is that it correlates highly with human judgements by averaging out individual sentence judgment errors. Figure 4.6 and Figure 4.7 shows the BLEU score for the normalized Tweet and NUS SMS data respectively. The results show **Mean BLEU score** of more than **0.8** is achieved for both the dataset. The model's output is compared against the human annotated text as provided in the datasets.

Figure 4.6: BLEU score for the normalized Tweets data

Figure 4.7: BLEU score for the NUS SMS data

4.6 Conclusion

Social media language is considerably different from other written text. Many of the efforts to illustrate and overcome this discrepancy have focused on normalization. In this work, a novel framework was proposed to deal with microtext normalization in Twitter for sentiment analysis purposes.

The novel technique combined different methods, namely, lexicon and rule based model. While SenticNet API was used as the polarity classifier, Soundex and the Ratcliff/Obershelp pattern-matching algorithm were seen as apt evaluation techniques. Although these methods seemed rather uncomplicated, they enabled to form a robust system. The results demonstrated that 85.31% of texts have a similarity index equal to or greater than 0.8, showing that this framework has the ability to correctly handle many types of normalization challenges.

Moreover, the proposed model has the ability to enhance machine intelligence to detect polarity since the sentiment classifier's accuracy improved from 77.47% to 81.59%.

The framework used for chatbot consists of a binary classifier which classifies a given sentence into either microtext or non-microtext. Binary classifier takes syntac-

tic features to determine a class label. Linear SVC gives an accuracy of 87.4% on Normalized Tweet Dataset and 92.95% on NUS SMS data. The addition of binary classifier also improves the overall execution time of the task. The detected microtexts are then passed through the lexicon. Lexicon transforms the OOV texts to their IV counterparts. BLEU score was taken as an evaluation metric and shows a mean of more than 0.8 in both the datasets. BLEU score is calculated by comparing the output of model to the human annotated IV text. The evaluation suggests that for any pretrained chatbot, there is no need to retrain it. Our microtext module can help to normalize the OOV text. It can be used as a plug and play device. As follows, it will be essential to keep working on adding new rule based approaches, extending the lexicon database, and comparing the proposed framework with existing statistical machine translation approaches. As a result, these improvements might offer the possibility of handling tweets in real time.

Lastly, future work will focus on increasing the accuracy of the proposed model by using a phonemic approximation with the ability to correct unintentional misspellings.

The next chapter 5 introduces IPA based microtext normalization framework. It encapsulates some cognitive features to transform microtext to their standard format.

Chapter 5

IPA based Microtext Normalization framework for Concept-level Sentiment Analysis

5.1 Introduction

¹ Concept-based approaches to sentiment analysis focus on a semantic analysis of text through the use of Web ontologies or semantic networks, which allow the aggregation of conceptual and affective information associated with natural language opinions [108]. The analysis at concept-level is intended to infer the semantic and affective information associated with natural language opinions and hence, to enable a comparative fine-grained feature based sentiment analysis. In this work, we propose PhonSenticNet, a concept based lexicon which advantages from phonetic features to normalize the OOV concepts to IV concepts. The International Phonetic Alphabet (IPA)² is an alphabetic system of phonetic notation based primarily on the Latin script. It was devised as a as a standardized representation of speech sounds in written form. Epitran [114] is used to transform concepts to their IPA encodings. The proposed framework can deal with the texts which are phonemic variants of their IV concepts. The results are discussed at three different levels:

- (i) information loss;

¹The work in this chapter has been published in Ranjan Satapathy, Aalind Singh, and Erik Cambria. “PhonSenticNet: A Cognitive Approach to Microtext Normalization for Concept-Level Sentiment Analysis.” Accepted in CSoNet (2019).

²<https://www.internationalphoneticassociation.org/content/full-ipa-chart>

- (ii) normalization of OOV concepts; and
- (iii) polarity detection.

The proposed system works for the phonetic class of microtext normalization. Though, the acronyms still rely on the lexicon built in [24]. Limitation of microtext normalization research is that it is considered either a find-and-replace pre-processing [13] or a phonetic substitution [24] which doesnot take standard IPA into consideration. We propose a framework which considers microtext as an area closely related to ASR and transforms the text to their phonemic forms.

5.2 Proposed framework

Limitation of microtext normalization research is that it is considered either a find-and-replace pre-processing [13] or a phonetic substitution [24] which does not take standard IPA into consideration. We propose a framework which considers microtext as an area closely related to ASR and transforms the text to their phonemic forms.

The architecture of the proposed model is depicted in Figure 5.1. The framework classifies a sentence as OOV or IV using binary classifier. Following this, the OOV sentence is passed through a concept parser, and then the concepts are transformed to IPA using Epitran. One persistent advantage of Epitran (over lexical resources) is its ability to produce consistent and accurate pronunciations for all words that a system encounters, not just thus listed in the lexicon. The IPA of OOV concepts are matched to the PhonSenticNet, and then the IV concept is fetched. The corresponding polarity of the IV concept is retrieved using sentic computing [2].

5.2.1 How Epitran works?

Epitran is a G2P (grapheme-to-phoneme). It takes word tokens in the orthography of a language and outputs a phonemic representation in either IPA or X-SAMPA. It produces precise pronunciations of an utterance which can provide useful feedback on how the model is performing on OOV words during evaluation. In our framework, we used the Epitran model to produce the pronunciation of words in a sentence. Each

Figure 5.1: Architecture of the framework

word is transposed to their phonetic subspace. We convert the OOV words (if any) in the phonetic subspace.

5.2.2 Classification of microtext

In this subsection, we employ various binary classifiers to detect OOV sentences so as to reduce the execution time of the overall algorithm. OOV sentence are those which contain at least on of the OOV terms. We observed that the execution time of the framework was reduced by 20%. We use the term frequency-inverse document frequency (TF-IDF) [115] approach for the task of feature extraction from a given text.

We first split the document into tokens assigning weights to them based on the frequency with which it shows up in the document along with how repetitive that term occurs in the entire corpora. We used this approach to train four different classifiers. The evaluation metrics such as Precision, Recall, F-measure and Accuracy have been enlisted in the Table 5.3.

5.2.3 Polarity Detection

The polarity detection module utilizes SenticNet lexicon [107]. While SenticNet can be used as any other sentiment lexicon, e.g., concept matching or bag-of-concepts model, the right way to use the knowledge base for the task of polarity detection is in conjunction with sentic patterns, sentiment-specific linguistic patterns that infer

polarity by allowing affective information to flow from concept to concept based on the dependency relation between clauses. The sentiment sources of such affective information are extracted from SenticNet by firstly generalizing multiword expressions and words by means of conceptual primitives and, secondly, by extracting their polarity. The lexicon has polarity values associated with concepts. For a given sentence, we average the polarity over all concepts and output the polarity either as positive, or negative. The following section 5.3 discusses about the datasets available in the area of microtext normalization.

5.3 Datasets

This section introduces to datasets used. The twitter dataset is available on request. The concept-level lexicon SenticNet³ and NUS SMS Data⁴ are publically available.

5.3.1 NUS SMS Corpus

This corpus has been created from the NUS English SMS corpus, where [54] randomly selected 2,000 messages. The messages were first normalized into standard English and then translated into standard Chinese. For our training and testing purposes, we only used the actual messages and their normalized English version. It also contains Singaporean colloquial terms (Singlish). Singlish is an English-based creole that is lexically and syntactically influenced by Hokkien, Cantonese, Mandarin, Malay and Tamil [116]. It is primarily a spoken variety in Singapore, to emerge as a means of online communication for Singaporeans [117].

5.3.2 SenticNet

Sentic API provides the semantics and sentics (i.e., the denotative and connotative information) associated with the concepts of SenticNet, that contains 100,000 nodes (words and multiword expressions) and thousands of connections (relationships between nodes). We used a concept parser [118] in order to tokenize the sentences to

³<http://sentic.net/senticnet-5.0.zip>

⁴<http://github.com/kite1988/nus-sms-corpus>

concepts and analyze them. The concepts in the SenticNet contains their corresponding polarities.

Concepts	Polarity	Soundex Encoding	IPA Encoding
a_little	Negative	A000_L340	æ_lɪtəl
abandon	Negative	A153	æbəndæn
absolutely_fantastic	Positive	A124_F532	əbsəluːtli_fəntəstɪk

Table 5.1: Sample Soundex and IPA Encodings with polarities for SenticNet5

5.3.3 Normalized Tweets

The authors in [24], built a dataset which consists of tweets and their transformed IV counterparts. We demonstrate our results by extracting concepts from unconventionally written sentences and then passing them through our proposed module to convert them to standard format concepts' and their corresponding polarities from SenticNet.

Soundex Encoding	Frequency	Example concepts
I536	225	interminable, intermingle
I516	182	impertinence, impertinent
C536	140	candour, candy, canter
U536	137	underfed, underfoot
T652	133	transgender, transgressor
I525	127	imagery, imaginary
O162	117	overconfidence, overconfident
U525	110	uncandid, uncanny
P632	104	participle, particle
I512	98	impious, impish

Table 5.2: Redundancy in Soundex encoding

$$\frac{-46710}{100000} * \log \frac{46710}{100000} \quad (5.1)$$

Table 5.3: Precision, Recall, F1 and Accuracy for each algorithm on different datasets

	NUS SMS Dataset								Twitter Dataset							
	LR				SGD				SVC				MNB			
	IV	OOV	IV	OOV	IV	OOV	IV	OOV	IV	OOV	IV	OOV	IV	OOV	IV	OOV
Precision	0.91	0.95	0.84	0.98	0.87	0.97	0.89	0.97	0.71	0.69	0.63	0.72	0.74	0.67	0.81	0.68
Recall	0.95	0.90	0.99	0.81	0.98	0.85	0.97	0.87	0.68	0.71	0.80	0.52	0.64	0.77	0.61	0.85
F1	0.93	0.92	0.91	0.89	0.92	0.91	0.93	0.92	0.69	0.70	0.70	0.60	0.68	0.72	0.69	0.76
Accuracy	0.9275				0.89875				0.915				0.9225			
									0.6962				0.6605			
													0.7013			
													0.7288			

5.4 Results

This section dives into experiments performed to develop a concept-level microtext normalization module. The experiments performed help in deciding the best set of parameters to achieve state of the art accuracy. We name the lexicon which we built from SenticNet as PhonSenticNet. It contains concepts and their phonetic encoding which is extracted from Epitran. The system classifies a sentence as OOV or IV using the trained classifier. Then it transforms OOV sentences from string to phoneme subspace by using Epitran describe in Algorithm 1 and 2. We show the frameworks application for concept level sentiment analysis using SenticNet. The detailed procedure is explained in the following subsections:

Text	Sentence Polarity before microtext normalization	Sentence Polarity after microtext normalization
I wil kil u	Neutral	Negative
m so hapy	Neutral	Positive
i dnt lyk reading	Positive	Negative
it is awesum 2 ride byk	Neutral	Positive

Table 5.4: Sample sentences with their polarities before and after microtext normalization

5.4.1 Classification of microtext

Different classifiers were trained on the two datasets namely NUS SMS data and Twitter dataset as shown in Table 6.4. The datasets were divided into training, validation and test as 70:20:10. As shown in the Table 6.4, the best performing classifiers for NUS SMS dataset and Twitter dataset are Logistic Regression (LR) and Multinomial Naïve Bayes (MNB). SMS writing is very orthodox, whereas Tweets are dynamic. The difference in accuracy on both the datasets is due to the writing style used on Twitter.

5.4.2 Comparision of IPA with Soundex

We compare our proposed IPA based method to Soundex [24] since only [24] have incorporated phonetic features to improve sentiment analysis. Soundex gives a lot of duplicate encoding which is depicted in Table 5.2, whereas IPA gives no duplicate

Figure 5.2: Visualization of string and phonetic distance

encoding. Each concept is unique in phonetic subspace, thereby increasing the accuracy for microtext normalization at concept-level. The number of concepts present in the lexicon is 100000. The duplicates⁵ due to Soundex encoding are 46080. This shows that using Soundex for microtext normalization has some information loss at the concept-level. As a result of Soundex encoding, we have 46080 ambiguous concepts which affect the microtext normalization in real time. Hence, we propose to use IPA for all the phonetic based microtext normalization methods. The IPA based encoding has no redundancy, and thereby no information loss occurs during microtext normalization.

Algorithm 1 Algorithm for microtext normalization using phonetic features

```

Sentence (S) =  $s_1, s_2, \dots, s_n$ 
 $c_i = \text{concept-parser}(S)$ 
For each concept  $c_i$  in  $S_n$ 
  closest-match-concept = PhonSenticNet( $c_i$ )
  if closest_match( $C_i$ , SenticNet) then
    return concept polarity
  else
    return sentence polarity
  end if
average over polarity of concepts for sentence polarity EndFor
return sentence polarity

```

⁵Repetition of a soundex encoding for greater than one

Algorithm 2 Closest Match Algorithm

```

Concept (C) =  $c_1, c_2, \dots, c_m$ 
For each concept  $c_i$  in C
  Sorensen ( $c_i$ , Senticnet)
EndFor
return phonetically closest matching concept

```

5.4.3 Microtext normalization of concepts using IPA

To enable a comparative fine-grained feature-based sentiment analysis [119, 120], the analysis at concept level is needed. Semantic analysis of text is the key to Concept-based approaches to sentiment analysis. It allows the aggregation of conceptual and affective information associated with natural language opinions [121, 122]. In order to normalize concepts found on social media, we built a resource for concept-level phonetic encodings by using concepts from SenticNet. We used a concept parser to extract concepts from the input text. The concepts are then transformed to a subspace where they are represented by their phonetic encodings. Table 5.1 shows sample concepts with their respective Soundex encodings and IPA from SenticNet. Table 5.2 depicts the redundancy in Soundex encodings, which makes it less viable algorithm to use. Epitran transforms each concept or word to their unique transformations in IPA space.

Phonetic encoding transforms concepts from the string subspace into their phonetic subspace. This transformation eliminates the redundant concept encoding produced by Soundex. The input concept is then passed through this phonetic subspace (IPA used in PhonSenticnet) to find the most phonetically similar concept and then returns it. The algorithm 1 and 2 describe the procedures in detail.

5.4.4 Polarity Detection with SenticNet

We compare accuracies of our framework with that of [24] for polarity detection⁶. The accuracy increases significantly by 6% as shown in Figure 5.3. The application of microtext normalization to sentiment analysis shows the importance of microtext normalization module for the texting data.

⁶<https://sentic.net/demos/#polarity>

Figure 5.3: Accuracy for polarity detection

5.5 Discussion and Future Work

The proposed framework contains concepts from SenticNet and their phonetics which is interpreted using Epitran. The lexicon is called as PhonSenticNet which is used as a lexicon for concept-level microtext normalization. The input sentence is broken down into concepts and then transformed into their phonetic encoding. The phonetic encoding is matched with the PhonSenticNet. Then the most similar matching concept and its corresponding polarity is returned as depicted in the algorithm 1 and 2. Microtext is language-dependent: the same set of characters could have completely different meaning in different languages, e.g., ‘555’ is negative in Chinese language because the number ‘5’ is pronounced as ‘wu’ and ‘wuwuwu’ resembles a crying sound but positive in Thai since the number 5 is pronounced as ‘ha’ and three consecutive 5s correspond to the expression ‘hahaha’. In conclusion:

- (i) We have taken Sorensen similarity to measure the distance. The Sorensen similarity shows how similar the two input texts are to one another, where 0 means similar and 1 means dissimilar as shown in Figure 5.2.
- (ii) Figure 5.2 shows distance between some of the non-standard and their standard concepts. The similarity is shown at both string and phonetic level.

- (iii) Previous work [24] shows sentence-level sentiment analysis, whereas in this work we focus on concept-level microtext normalization.
- (iv) Table 5.2 depicts the redundancy in Soundex which makes it difficult to be used in real time.
- (v) It can be observed from Table 5.3 that the twitter dataset does not perform as good as the NUS SMS data. The reason behind it is, the twitter dataset contains acronyms like lol, rofl, etc instead of phonetic substitution. This also suggests how the way of writing differs in both messages and tweets.

In the next chapter 6, we discuss Deep Learning based approaches to microtext normalization.

Chapter 6

Deep learning based microtext normalization framework

6.1 Introduction

¹ This chapter tries to solve semiotic and phonetic class of microtext normalization. In order to solve it, variations of seq2seq models are developed. The models were trained on 5 different datasets. The proposed framework for normalization consists of two parts: an encoder for reading the input sequence and encoding it into a fixed-length vector, and a decoder for decoding the fixed-length vector and outputting the predicted sequence. Indeed, the use of these parts gives the framework its' name of encoder-decoder LSTM which is designed specifically for sequence to sequence (seq2seq) problems. The task of translating OOV words can be understood from the perspective of machine learning as learning the conditional distribution $p(\mathbf{w} \mid \mathbf{s})$ of a target sentence $\mathbf{w}$ given a source sentence $\mathbf{s}$. Once the conditional distribution is learned by a model, one can use the model to directly sample a target sentence given a source sentence either by actual sampling or by using a search algorithm to find the maximum of the distribution.

This chapter shows application of microtext normalization on sentiment analysis with Deep Learning approach. It also introduces to a microtext normalization dataset. It shows how the microtext normalization improve the sentiment analysis by a significant margin. We compare four Deep Learning models which improved the accuracy of

¹The work in this chapter has been published in Ranjan Satapathy, Yang Li, Sandro Cavallari, and Erik Cambria. "Seq2Seq Deep Learning Models for Microtext Normalization" In International Joint Conference on Neural Networks (IJCNN), pp.1–8, 2019.

Figure 6.1: Encoder-Decoder framework.

the sentiment analysis namely: Long Short-Term Memory (LSTM), Attentive LSTM, Convolutional Neural Network (CNN) with LSTM and gated recurrent units (GRU). Accuracy is sentence level matching, i.e., output of the framework is concatenated to make a sentence and is matched with the annotated sentence. Results show that the attentive LSTM and GRU both increase the sentiment analysis accuracy in the range of 4%-7% though LSTM and CNN with LSTM improves the accuracy in the range of 2%-4%.

We first introduce a corpus for English Microtext normalization (CEMt-Norm), which is manually annotated for microtext normalization and sentiment analysis.

6.2 Corpus for English Microtext Normalization (CEMt-Norm)

This section introduces to a corpus for microtext normalization in the English language. We describe a methodology to build the corpus with guidelines followed during the annotation process. We also present some baseline results of the corpus on ma-

chine learning algorithms and make the code and resources publicly available. A total of 1432 OOV sentences and their IV sentences made it to the final corpus with their corresponding polarity. We also show the accuracy of character-level sentiment analysis module as a baseline. With the Natural Language Processing (NLP) techniques relying on the dataset and corpus from the internet, it is an absolute necessity to build corpus specific for the tasks. While many NLP researchers and engineers are struggling with the difficulties imposed by bad language [123], there has been relatively little consideration of why the language in social media is so different from our other corpora. The next section 6.2.1 discusses the preparation of corpus.

6.2.1 Preparation of Corpus

This section discusses the procedure followed to create a corpus for English Microtext Normalization (CEMt-Norm). We start by crawling twitter using Twitter’s streaming API. There were no constraints on what keywords to use as a filter. So, the data we got was across the globe and covered different topics starting from a well-wish tweet to a debate on nuclear energy. We collected around 10K tweets in a day. Next, we employ three annotators to clean each tweet and write it’s corresponding in-vocabulary(IV) text. It followed a majority voting to decide the final tweet and its IV text. We dropped the IV text where any of the three annotations were different, and also remove their corresponding tweet from the corpus to maintain coherence. A total of 1432 tweets and their IV text made it to the final corpus.

6.2.1.1 Preprocessing

We used the basic modules such as a dictionary, affix analysis, number and dates detection to analyze tokens in the corpus, and a token is considered as an OOV if there is no match in any of the modules. As a first step, we tokenized the sentences to get tokens. The tokenizer’s rules were tuned to remove usernames (), hashtags (#), e-mail addresses, URLs, and emoticons. As we aim to detect only the sentiments and not their intensities, emoticons were redundant for this matter. As a second

preprocessing step, we used Pyenchant² to identify OOV words in the corpus. The dataset contains sentences with at least one OOV word.

6.2.1.2 Exploration of corpus

The corpus has 1432 OOV text with their corresponding IV text. The corpus also contains sentiment of the text. The corpus comprises a total vocabulary of 15,798 words. 3541 of 15,798 words are OOV. We calculated the OOV words using Pyenchant tool as a dictionary. The words which return **False** were regarded as OOV. We also calculated the unique set of vocabulary for both total words and OOV. The unique words in total vocabulary are 4679, and unique OOV words are 1861, which is about 40% of total unique words.

	Total Vocab.	Unique
OOV	3541	1861
Words	15798	4679

Average unique words per sentence : $4679/1432 \approx 3.3$

Average unique OOV words per sentence : $1861/1432 \approx 1.3$

6.2.1.3 Corpus Annotation

The corpus contains around 1432 text and is annotated by 3 experts (cohen’s kappa = 0.79) in terms of polarity (positive, negative or neutral). We also asked 3 independent experts to normalize each of the OOV text to IV text (Table 6.1). The normalization task is important, so all the reviewers agreeing was important to build a corpus. So, the final corpus did not contain sentences with at least 1 expert disagreeing on the normalized text. All the annotators agreed on the final normalization, so no metric to measure inter-annotator agreement needed for normalization. If human’s do not agree on a normalization then it will only add bias to the model if trained on this data, hence were discarded. For example: “I dnt lyk this vdo” was tagged as “I do not like this video” whereas one tagger tagged it as “I did not like this video”. Hence,

²<https://github.com/rfk/pyenchant>

“I dnt lyk this vdo” was removed from the corpus. The polarity count is depicted in Figure 6.2. The two most challenging cases we identified during the annotation process are the normalization of abbreviations and phonetic variation.

Table 6.1: Sample corpus showing OOV, IV and it’s polarity

	OOV text	IV text	Polarity
1	pepul wth long fingersil b downgrding soon 2 a nokia	people with long fingers will be downgrading soon to a nokia	Neutral
2	u r a lil ugly	you are a little ugly	Negative
3	im extremely sorrrrrry for not talking this time pls don’t be upset with me	i am extremely sorry for not talking this time please do not be upset with me	Negative

- (i) In the text, “2many of yall gals thinking all the shit” “2” is “too” but in “woow im sorry 2 hear dat.. i hope she ok” “2” is “to”.
- (ii) one-to-many forms of abbreviations: like together is written as 2gethr, tgthr, togethr and many more.
- (iii) We realized there is no standard way to write a word in abbreviations. People rely on creativity to make abbreviations. For example: “*\$” means “Starbucks”. In such cases, experts had a discussion and normalized according to the unified decision.

6.3 Deep Learning Model

The task of translating OOV words can be understood from the perspective of machine learning as learning the conditional distribution $p(w|s)$ of a target w given a source s . In our setup the target w is a word and source s is input character. The model takes an input character, and outputs an IV word with highest probability. An OOV word is unconventional in structure whereas, an IV word is written in its standard form. We structured our normalization task by using a character-based encoder-decoder model with variations which will be discussed in 6.3.1 in detail. Let $\mathbf{s} = (s_1, s_2, \dots, s_n)$ be the character sequences of the input sentence (the source sentence). Similarly, let

Polarity count in the corpus

Figure 6.2: Polarity count in the corpus

$\mathbf{w} = (w_1, w_2, \dots, w_m)$ be the word sequence of the IV sentence (the target sentence, also the output). The notations n and m are the total lengths of the characters and words in $\mathbf{s}$ and $\mathbf{w}$, respectively.

Then, the (normalization) probability of w given OOV sentence $\mathbf{s}$ can be written as Equation 6.1.

$$p(\mathbf{w} \mid \mathbf{s}, \theta) = \prod_{j=1}^m p(w_j \mid w_{<j}, \mathbf{s}), \quad (6.1)$$

where θ represents a set of all model parameters in the encoder-decoder model, which are determined by the parameter-estimation process of a standard softmax cross-entropy loss minimization using training data.

Therefore, given θ and $\mathbf{s}$, our normalization task is defined as finding $\mathbf{w}$ with maximum probability given by Equation 6.2

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{s}, \theta), \quad (6.2)$$

where $\hat{\mathbf{w}}$ represents the model output.

Each character in the input can be encoded using one-hot vector $x_i \in \{0, 1\}^v$. Hence, the inputs are represented as a binary matrix $x_1, \dots, x_n \in (0, 1)$ where v is

Cases	Datasets	LSTM	Attentive LSTM	GRU	CNN with LSTM
With Mask	Lexicon [24]	77.36%	80.63%	78.40%	77.82%
	Tweets [24]	64.71%	64.04%	69.85%	69.90%
	SMS [54]	76.15%	76.84%	77.24%	77.20%
	CMUdict	82.78%	82.17%	85.21%	84.73%
	LexNorm [58]	74.91%	79.91%	68.71%	78.83%
	CEMt	64.71%	63.84%	70.85%	72.86%
Without Mask	Lexicon [24]	78.58%	80.33%	77.22%	78.55%
	Tweets [24]	65.71%	66.93%	65.85%	68.55%
	SMS [54]	76.06%	77.09%	76.60%	76.59%
	CMUdict	81.16%	81.1%	85.75%	86.87%
	LexNorm [58]	75.8%	80.1%	68.81%	79.1%
	CEMt	65.55%	65.24%	68.72%	73.20%

Table 6.2: Accuracy of encoder-decoder models on multiple dataset with mask and without mask

the vocabulary and n is the number of input texts, with padding wherever necessary. Then LSTM was applied to get the feature of the input, in the LSTM cell the hidden units were varied from 128, 256, and 512 and was fixed at 256 as this gave the best result. Based on the learned feature, another LSTM was applied to get the reconstructed characters. In this framework, the learning rate is fixed at 10^{-4} . Batch size is set to 64. The prediction of word at each time step is given by the Equation 6.3:

$$p(w_t|\cdot) = \delta(\phi(x_t), h_{t-1}) \quad (6.3)$$

where $\delta(\cdot)$ denotes the softmax function, w_t refers to the word at time-step t , $\phi(x_t)$ represents the encoding value from the character at time-step t , and h_{t-1} denotes the previous hidden value of LSTM in the decoding process.

To learn the model parameters, there are two cases we have considered. The cross-entropy loss in this framework is depicted in Equation 6.4. Because all of the predicted output are set to the fixed length, whereas the output word length varies. The empty places which has no character is filled with a placeholder, and it will be counted into the final loss in case of “no mask” explained in section 6.5.2.

The cross-entropy losses are trained by the Adam Optimization algorithm [124].

$$L = -\frac{1}{m} \sum_i^m w_i \log p(w_i, \theta) \quad (6.4)$$

$$L_M = -\frac{1}{m} \sum_i^m w_i \log p(w_i, \theta) M \quad (6.5)$$

where m in the denominator is the output length, θ represents a set of all model parameters in the encoder-decoder model. $M \in \{1, 0\}^m$ is from the true length of the output, the place has character is set to 1, otherwise is 0.

6.3.1 Model Variations

The initial work on microtext normalization using LSTM was done by [125]. They have used [58] dataset to show the performance of microtext normalization. Whereas, on the other hand, we used a 2-layer encoder that reads input characters, an embedding layer, and a 2-layer decoder that produces word sequences. We used L2 loss for encoder and ReLU loss for decoder. For training we used AdamOptimizer and learning rate was set to $1e - 4$. Batch size is set to 64 and latent dimension = 256. The data were partitioned on train-test set as 80-20.

We have considered four different encoder-decoder architectures namely, LSTM, LSTM with Attention [126], CNN with LSTM and GRU encoder-decoder. We compare LSTM, attentive LSTM, GRU and LSTM with CNN in the task of sequence modeling for text normalization. Sequence modeling aims at learning a probability distribution over sequences, as described in Equation 6.1, given a set of training sequences. Here are detailed descriptions:

- **LSTM:** LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods is practically their default behavior, not something they struggle to learn. Mathematically LSTM is given by

6.6

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ C_t^+ &= \tanh(W_C [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * C_t^+ \\ o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (6.6)$$

- **Attentive LSTM:** We added attention mechanism [127] in the decoder module.

Then, the attention scores were calculated and that outputs the context vector which will be concatenated with hidden state of the decoder, thereby the final word is predicted. Attention weight-vector α is given by Equation 6.7:

$$\begin{aligned} h &= LSTM(s) \\ \alpha &= \delta(W_h h) \\ c &= \alpha h \end{aligned} \tag{6.7}$$

W_t, W_h are the parameters, $\delta(\cdot)$ is the softmax function, s is input characters, h is the encoding of the input vector (hidden value) and c denotes the output.

- **GRU:** GRUs are a gating mechanism in recurrent neural networks, introduced by [128]. Apart from LSTM, GRU is another popular model for sequence data. The mathematical representations are given as:

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\ h_t^+ &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * h_t^+ \end{aligned} \tag{6.8}$$

where h_t, z_t, r_t denote the hidden value, the update gate vector, and the reset gate vector, respectively. W, U, b are the parameter matrices and vector. σ and $\tanh$ are the sigmoid function and hyperbolic tangent respectively. h_t^+ is the current memory content and h_t is the final memory at the current time step

- **LSTM with CNN as a feature extractor:** We also used CNN as a feature extractor and compared it with the rest of the models. CNNs are often applied both on image data [129] and the text data [130]. In our work, the structure of CNN from [130] is used to extract the features from the input data. The mathematical representations are given as:

$$h_i = \tanh(wx_{i:i+j-1} + b) \tag{6.9}$$

	Features	Microtext Lexicon	Norm Tweets	NUS SMS	CMUdict	LexNorm
Input	Char set	97	165	159	85	108
	Char MaxLen	15	146	223	33	152
Output	Wordset	1565	6328	5397	77	9077
	Word MaxLen	13	46	59	32	41

Table 6.3: Statistics of the datasets

This filter is applied to each possible window of characters in the source sentence

$x_{1:j}, x_{2:j+1}, \dots, x_{n-j+1:n}$ to produce a feature map given by:

$$h = [h_1, h_2, \dots, h_{n-j+1}] \quad (6.10)$$

where h is the hidden value.

One-hot embedding has been used to convert input texts (in all the datasets) to vectors. As microtext comprises OOV words, word embeddings [131] pretrained on English words will not be useful as they will not be able to determine the correct embeddings.

Instead, each input (word or sentence) is treated as a composition of characters, which contains no more than 165 characters in total in our corpus. In our model, after encoding all of the characters with one-hot embedding, a single layer of the dense embedding is added.

6.4 Datasets Used

The purpose of our work is to normalize any kind of data irrespective of their structure and property. Hence, we evaluate our model on five datasets from three different domains. In the tables below, we renamed Lexical Normalization dataset as LexNorm Microtext Lexicon as Lexicon, Normalized Tweets as Tweets, NUS Social Media Text Normalization and Translation Corpus as SMS and the phonetic dictionary as the Carnegie Mellon Pronouncing Dictionary (CMUdict). The statistics of these datasets are shown in Table 6.3. Our model is a character-level input to word-level output encoder-decoder model. The *Input char set* is the total number of input characters in the respective datasets. *Word set* is the number of output words in the respective

corpus corresponding to the given input. *Input MaxLen* is the max length of the input, and the *Word MaxLen* which is the max length of the corresponding words. The details of the datasets used are explained in the following subsections.

6.4.1 LexNorm2015

The LexNorm2015 dataset was introduced for the shared task on lexical normalization, hosted at WNUT2015 [58]. In our experiment we took only the test set as there is no annotation available for Test set. It contains 2950 tweets which constitutes about 44385 tokens.

6.4.2 Microtext Lexicon

Authors in [24], developed a lexicon for microtext normalization. This acts as a look-up table for performing normalization. We used their dataset to learn the intrinsic character-level behavior of the different classes of microtext as mentioned in their work. We ran our code on the whole dataset and also only on “PHON” classes. It was observed that “PHON” class helped the models to learn quickly, however its limited number has overfitted all the models. Because of this overfitting issue, we continued with the whole dataset for the normalization task.

6.4.3 Carnegie Mellon Pronouncing Dictionary

Carnegie Mellon Pronouncing Dictionary (CMUdict) is a free pronouncing dictionary of English, suitable for uses in speech technology CMU dictionary³. It consists of phonetic representation for each word. We use this dictionary to learn the phonetics of the words in the lexicon. Our results show that the encoder-decoder model works best with the CMU dictionary. As it is a speech dictionary, preprocessing was required before inserting it to our model. Preprocessing includes removing terms like ‘!EXCLAMATION-POINT’ which is written as ‘EH2 K S K L AH0 M EY1 SH AH0 N P OY2 N T’ from the dictionary.

³<http://speech.cs.cmu.edu/cgi-bin/cmudict>

6.4.4 NUS SMS Corpus

This corpus has been created from the NUS English SMS corpus⁴, the authors [54] randomly selected 2,000 messages. The messages were first normalized into standard English and then translated into standard Chinese. For our training and testing purposes, we only used the actual messages and their normalized English version. It also contains non-English terms specifically the Singaporean colloquial language called Singlish, which LSTM had no problem in learning. Singlish is an English-based creole that is lexically and syntactically influenced by Hokkien, Cantonese, Mandarin, Malay and Tamil [116]. It is primarily a spoken variety in Singapore, to emerge as a means of online communication for Singaporeans [117].

6.4.5 Normalized Tweets

Authors in [24], built a lexicon which consisted of normalized OOV words from tweets and their IV counterparts. We demonstrate our results on this dataset utilizing the context within the sentences by including unconventional written sentences together with the transformed standard ones.

6.4.6 Corpus for English Microtext Normalization (CEMt-Norm)

This dataset has been introduced in section 6.2. The corpus consists of a total of 1432 OOV sentences and their IV sentences made it to the final corpus with their corresponding polarity. We use it to train and test the models in Table 6.2.

6.5 Results and Discussion

In this work, we show different seq2seq models to normalize informal English to plain English. Initially, we observed no change in accuracy with a consistent decrease in model loss. With a change in variation in the batch size, we found that 32 is the best batch size for all these datasets. We use accuracy as an evaluation method in this chapter. In lexical normalization, it is number of sentences which are exactly

⁴<http://github.com/kite1988/nus-sms-corpus>

matching after normalization. In this section, we will introduce the result discussions with and without mask and discussion on experimental results.

Features	CEMt-Norm Dataset		
	LR	Linear SVC	MNB
Unigram	96.7%	98.7%	93.4%
Bigram	96.7%	98.6%	93.9%
Trigram	96.4%	98.5%	93.6%

Table 6.4: Accuracy of classifiers on the corpus

Next, we will also discuss the evaluation of CEMt-Norm, which we introduced in this chapter.

6.5.1 Evaluation of the corpus (CEMt-Norm)

The source of our corpus, i.e., Twitter, is used by people mostly to share opinions which makes it very suitable for sentiment analysis. As evident from Figure 6.2, the positive and negative polarities are almost similar in number whereas Neutral polarity is less in number. Microtext normalization is a character-based approach with phonetics rather than just words, so we chose character-based embedding [132]. We tested our corpus for sentiment analysis with Flair’s sentiment classifier. It is a character-level LSTM neural network which takes sequences of letters and words into account when predicting. The pre-trained model shows accuracy of 46.4 % on OOV text. This shows that there is a huge room of improvement for sentiment analysis.

6.5.1.1 Discussion on corpus

The corpus has 1432 sentences with 38% positive, 39.9% negative and 22.1% neutral. There are 3.3 average unique words per sentence and 1.3 average OOV words per sentence. Unique OOV words constitute 40% of total unique words. It covers all the different variations of microtext from abbreviations to phonemic variations. The results depict that Linear SVC outperforms in all the features. The sentiment analysis by a pre-trained character level embedding model shows the accuracy of 47.6% shown in Figure 6.3. The Figure shows the difference in sentiment analysis accuracy. When we dived deeper into the the wrongly predicted models, we found most of the

Figure 6.3: Sentiment Analysis accuracy for the corpus on character-level pre-trained model

sentences contained phonetic microtext, for example: “luv” instead of “love”. As discussed before in Chapter 5, phonetic replacements require phonetic (IPA) knowledge to transform them. Last but not least, we are aware that the size of the corpus is limited. Extending the corpus and considering the different OOV categories would enable us to perform a more detailed evaluation, especially for machine learning purposes. The corpus can also be used to benchmark any microtext normalization system. It will help the coupling of sub-symbolic (phonetics) with symbolic (machine learning) Artificial Intelligence to transform the OOV concepts into their standard IV form⁵.

6.5.2 Effect of Masking in the Deep Learning model

The padding process has shown that not only the model can be efficiently processed but also that the output is affected as perceived in our results. On the one hand, the addition of the mask is performed to the position where there are no characters in the output. On the other hand, without the mask it pads the output vector to make it

⁵A word of caution that this corpus contains some usage of foul languages as it is collected from public sources.

Figure 6.4: Architecture for polarity detection comparison.

have a fixed length. The results have been different since without mask the position without a word is also counted as the right prediction.

The results of each model on different datasets have been depicted in Table 6.2. In this work, we show different seq2seq models to normalize the unconventional texts to standard English texts.

6.5.3 Sentiment Analysis

Sentiment analysis is evaluated on the NUS SMS data and normalized tweets, and not on the other lexicons as most of the words are abbreviated and do not contain any sentiment associated with the words. For polarity detection, we relied on the Sentic API⁶. In this section, we compare the OOV and models' IV texts for polarity detection. Both the OOV and model's output IV texts are fed to the polarity detection module.

Both of them are compared against the human annotated IV texts' output, as shown in figure 6.4. Table 6.5 depicts the output for polarity detection module for each of the microtext normalization models discussed in section 6.3.1. It can be

⁶<http://sentic.net/api>

seen that attentive LSTM and GRU have improved the accuracy by 6% and 6.3% respectively for the normalized tweets and 4.4% and 5.3% respectively for the NUS SMS data. Other models also increase the accuracy but only in the range of 2%–4% for both the dataset. The improvement in accuracy depicts that microtext is indeed an important preprocessing module when dealing with social media texts.

6.5.4 Discussion

From Table 6.2, it can be observed that the best results from the different models across a variety of datasets are not the same, e.g., in the Microtext Lexicon, it was observed that the input and output comprised no more than “15 characters” and “13 words”, respectively. This means that attentive LSTM works well for short texts.

Then, when the length of the input and output increases, as in the case of CMUdict, the model GRU works best with mask and CNN with LSTM works best without. Hence, GRU and CNN have the capability to handle the input of up to 30 characters cases. Lastly, when the length of the input increases to 165 and 159, such as in normalized tweets and the NUS SMS datasets respectively, the GRU and CNN with LSTM achieve almost the same accuracy. The GRU network and CNN with LSTM both works best for mid to large number of texts as can be seen in CMUdict, normalized tweets and the NUS SMS datasets. The GRU model works better for both longer and shorter sentences, as well as for words with the mask, yet CNN with LSTM works better without the mask. Therefore, GRU and CNN with LSTM both demonstrate the capability to be used in text normalization in all the three domains. The results suggest that our model is well equipped to work on different datasets which are in

Dataset	OOV Output	Models			
		LSTM	Attentive LSTM	GRU Cell	CNN with LSTM
nus_sms_data	78%	80.4%	82.4%	82.8%	81.3%
norm_tweets	77.47%	79.8%	83.5%	83.8%	81.5%

Table 6.5: Comparison of polarity detection accuracy for different models

three different domains i.e., a microtext lexicon, a microtext corpus (LexNorm), a phonetic dictionary (CMUdict) and Tweets.

The next chapter [7](#) summarizes the thesis direction and its contribution with future application of microtext normalization.

Chapter 7

Summary and Future Work

Traditional spelling correction is a subset of microtext normalization. Both the frequency and severity of spelling errors in microtext are significantly greater than in standard text. Typically, a lexicon lookup or checking a text’s grammar cannot decide the microtext term’s validity. Therefore, general-purpose spelling correction methods that rely on lexicon lookup cannot solve normalizing microtext. The problem of normalizing microtext is viewed as recovering the intended word or word sequence given an observed word that may or may not be an OOV. In this chapter, we will review the contribution of this thesis to microtext normalization. Afterwards, we analyzed the limitations of our proposed methods in Section 7.2. The application of microtext normalization is not limited to sentiment analysis. For example, summarization, knowledge graph and other NLP fields [2] can also benefit from a microtext module.

7.1 Summary of Proposed Method

Language imposes constraints on the meaning of a sentence, both on the basis of the syntactic form of the sentence and on the basis of its contextual usage. Knowledge of these linguistic constraints can help us design NLP systems with higher accuracies. The current state of art takes texting language as noise and thus ignore them while models are trained just on standard text. However, with the increasing dependency on social media data, the accuracy of the trained framework is decreasing. The primary reason for the decrease in accuracy is that the texting language is very different from the standard language. Texting language involves the phonetics and abbreviations,

which requires linguistic knowledge to be incorporated into the model. Therefore, this thesis studies the paradigm shift to texting language from the standard language. It also tries to solve the issues standard NLP systems face due to the rise of texting language.

In our experimental setup with Nadine Social Robot [14], we found public providing feedback to Nadine using microtext. The feedback was collected as part of survey through google forms and printout forms. This provides an insight to the usage of microtext, which is not just limited to online platforms like facebook, Twitter and Whatsapp, but also offline and online survey. It suggests that microtext is pertinent in our current society. In Chapter 4, we propose an unsupervised hybrid method to transform OOV sentences to IV sentences. The transformed sentences are then passed through a polarity detection module and compared with human-annotated IV sentences for polarity. To tackle the phonetic replacement challenge posed, in Chapter 5, we develop a framework for IPA based OOV concepts to IV concepts. We propose concept-level microtext normalization algorithm in this regard, to transform the OOV concept or word to its phonetic subspace. In real-time, the phonetic subspace lexicon is used to transform back an OOV concept or word to its IV form using the closest match algorithm. However, as the language is very dynamic, it is cumbersome to maintain a lexicon. The framework needs to be trained dynamically, to learn new OOV words and transform it. The results show that twitter data is more dynamic and uses more abbreviations which is difficult for a machine learning or Deep Learning based method. Towards this end, we constructed a Deep Learning-based framework, as discussed in chapter 6. In the same chapter we introduced a corpus for microtext normalization and show the impact of pre-trained sentiment analysis framework with this corpus. SMS data and tweets are fed to train the proposed framework. The transformed texts were passed through the polarity detection module to show the improvement in the trained polarity detection module. After which, we present some baselines for classification of text as microtext or standard text.

7.2 Current Limitations

In this thesis, we primarily emphasize on microtext normalization as a pre-processing task which transforms the OOV texts to standard texts. The current state of art present in NLP uses the standard texts. The texting language is ever-growing and differs from country to country depending upon the dialect a person speaks.

7.2.1 Hybrid method for microtext normalization

The Soundex encoding does not preserve the phonetic meaning of a word. Hence, in our latest work, Soundex was replaced by a cognitive approach as Soundex gives redundant output whereas IPA gives unique output for each input string. In future, we could train NLP tasks on microtexts so that it reduces the processing time, without affecting the accuracy. The proposed unsupervised method gives much more control over the framework but has limited learning capabilities due to presence of rules and lexicon.

7.2.2 Deep Learning method

The proposed framework improves the accuracy of polarity detection that is limited to the English language. In this, we have addressed the absence of a baseline model in microtext normalization on multiple dataset. These could be further expanded to transformer based model.

7.2.3 IPA based method

Integration of IPA based method to NLP is a more cognitive approach. This method encapsulates phonetic features, thereby missing out on abbreviations and misspellings. It also fails to incorporate the dynamic nature of social media language and hence making it difficult to maintain as a lexicon. We developed a concept based lexicon which takes advantage from phonetic features to normalize the OOV concepts to IV concepts. It is the link to the knowledge-based sentic computing framework [133]. It enhances sentic computing's capability to normalize microtext by incorporating

phonetic knowledge into the knowledge base. In future work we plan to use the phonetic features in addition to the word features to train any NLP model.

7.2.4 Corpus for Microtext Normalization

The corpus encapsulates different variations of microtext in English language. For the future work, we plan to include code-switching [134] and multi-lingual texts [135]. Machine learning or Deep Learning models could be trained from this corpus to transform OOV to IV in any language.

7.3 Future Application

Bringing together researchers from multi-disciplinary fields will pave the path to bring the NLP methods, and tools in line with the current needs of the NLP community in academia, industry, and the general public. Application of microtext analysis could vary from emotion recognition in chats and tweets to online piracy and fake news detection. For instance, fake news is generated and transmitted online by the users due to the presence of echo chamber effect [136]. Many efforts by government and research institution have gone into detecting and remove them before reaching the masses. Social media minimizes the barrier among the public and makes it an ideal platform to disseminate information. The most crucial aspect of fake news is the linguistic features. It is written in a way to lure the public into believing it is true while distorting the facts around it [137]. This surge of activity around online news can lead to grave repercussions, but also substantial potential political and economic benefits. Such benefits encourage malicious entities to create, publish and spread fake news.

The microtext normalization can be applied on computer vision for driverless car. As, shown in Figure 7.1 the word “thru” will not be understood and perceived as “through” by machine trained on standard text. With the increasing numbers of driverless cars, understanding microtext as shown in the figure 7.1 will become very important. Applications are not limited to computer vision, another application is online piracy. It is done by renaming the movies by using their phonetic and/or visual

Figure 7.1: Application of microtext in computer vision

priming. For example renaming “Snowden” for “Sn0wd3n” and uploading it to the internet, thereby making it difficult to track.

Other NLP tasks like topic detection, named entity recognition, subjectivity detection can also benefit from microtext normalization. This thesis focusses on microtext normalization for the English language but the use of microtext is not limited to the English language. Though the literature covers some of the non-English languages, the list is not limited to the mentioned with the popularity of social media. Other languages like Indonesian, Hindi and some of the European languages will also need to come up with their datasets and trained models to normalize text in their language.

Another domain which is code-switching needs to be addressed as well. Code-switching is using two or more languages in one sentence. Thus, if a model is trained on only one language for microtext normalization it will treat cross-lingual texts as noise. Some of the examples are from Hindi + English (generally referred to as Hinglish) tweets:

“Cmon gals, waqt he shine karne ka!” which translates to “Come on girls, it is time to shine!”

It includes microtext *cummon*, *gals* and *waqt*. *Waqt* which translates to *time*

can be written in different ways, hence, classified as *microtext*. Incorporating dialect information is crucial to building an open domain NLP system. In future, we would like to focus on multi-lingual and code switching features of a text to be incorporated in the model.

7.3.1 Beyond Microtext

The electronic computer radically changed the ways our society and economy work at all levels [138]. In brachygraphic terms, it seems that nothing changed yet. This review provides an insight into the work done in microtext normalization and its application. However, we also need to address what the future holds for this research field. Shall we expect the end of brachygraphy when the keyboard will be completely replaced by vocal dictation? Or are we entering into a more sophisticated scenario, that seems to have been already forecasted by the emojis, which (re)discovered the link between orality (words) and visual knowledge aggregation of ideas or emotions (images)?

Indeed, humans use both words and images to share the creations of their imagination (thoughts), and we know from both the humanities and cognitive sciences that human memory and imagination are strictly interrelated. In human communication history, over time and across space, we can see different relationships between text and images, with a predominance of one of them over the other and rarely a real emergence. This has already been a subject of discussion in pedagogy since Quintilianus' *De Institutio Oratoria* (XI, 2, 11-17) and Martianus Capella's *De Nuptiis Philologiae et Mercurii* (V, 538-539), where they refer to the famous story of Simonides of Ceos, mentioned by Plato in *The Republic* (331e-332c, 335e) and in *Protagoras* (338e6-349a6) [139]. According to the Byzantine author Michael Psellos (*Tymotheos. On the Working of Demons*), Simonides of Ceos coined the phrase "the word is the image of the thing" [140], and, according to Plutarch (*On the Glory of the Athenians*), "he calls painting silent poetry and poetry painting that speaks" [141].

In terms of computational speculation on the future of communication figures and tools, the dichotomy between words and images seems to become less rigid, with a

cascade of possible consequences. The advent of the computer has exponentially augmented the capacity of individual human beings of both storing and processing data. Currently, Artificial Intelligence is an ultimate tool developed by humans (artificial, viz. human-made) to expand and augment human imagination. Microtext could facilitate the merging of brachygraphy and emojis in the interest of human empathy as well as a more effective and direct sharing of emotions and sentiments. This could tremendously help to avoid the interpersonal misunderstandings that affect social media communication and also assist computational sciences in the complex processing of the polysemy of human imaging and symbolization (i.e., teaching to machines the different human understandings of reality, which involves cultural diversity). Digital semantic aggregation and visualization could paradoxically free human imagination from the mechanical philosophy constraints and expand the horizon of human creative power.

Appendix A

List of Publications

- **Ranjan Satapathy**, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. “Phonetic-based microtext normalization for twitter sentiment analysis.” In 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 407-413. IEEE, 2017.
- **Ranjan Satapathy**, Yang Li, Sandro Cavallari, and Erik Cambria. “Seq2Seq Deep Learning Models for Microtext Normalization” In International Joint Conference on Neural Networks (IJCNN), pp.1–8, 2019.
- **Ranjan Satapathy**, Iti Chaturvedi, Erik Cambria, Shirley S. Ho, and Jin Cheon Na. “Subjectivity detection in nuclear energy tweets.” *Computación y Sistemas* 21, no. 4 (2017): 657-664.
- David Vilares, Haiyun Peng, **Ranjan Satapathy**, and Erik Cambria. “BabelSenticNet: a commonsense reasoning framework for multilingual sentiment analysis.” In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1292-1298. IEEE, 2018.
- **Satapathy, Ranjan**, Aalind Singh, and Erik Cambria. ”PhonSenticNet: a cognitive approach to microtext normalization for concept-level sentiment analysis.” In International Conference on Computational Data and Social Networks, pp. 177-188. Springer, Cham, 2019.
- Nidhi Mishra, Manoj Ramanathan, **Ranjan Satapathy**, Erik Cambria, and Nadia Magnenat-Thalmann. “Can a Humanoid Robot be part of Organizational

Work Force? A User Study leveraging on Sentiment Analysis.” Accepted in ROMAN (2019).

- Iti Chaturvedi, **Ranjan Satapathy**, Sandro Cavallari, and Erik Cambria. “Fuzzy commonsense reasoning for multimodal sentiment analysis.” Pattern Recognition Letters 125 (2019): 264-270.
- **Satapathy, R.**, Cambria, E., Nanetti, A., Hussain, A., “A Review of Shorthand Systems: From Brachygraphy to Microtext and Beyond”. Cogn Comput 12, 778-792 (2020).

References

- [1] E. Cambria and B. White, “Jumping nlp curves: A review of natural language processing research,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.
- [2] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, “Sentiment Analysis is a Big Suitcase,” *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 74–80, 2017.
- [3] I. Chaturvedi, E. Cambria, S. Poria, and R. Bajpai, “Bayesian deep convolution belief networks for subjectivity detection,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2016, pp. 916–923.
- [4] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*. Cham, Switzerland: Springer, 2017.
- [5] K. D. Rosa and J. Ellen, “Text classification methodologies applied to micro-text in military chat,” in *Proc. Eight International Conference on Machine Learning and Applications*, Miami, 2009, pp. 710–714.
- [6] F. Liu, F. Weng, and X. Jiang, “A Broad-Coverage Normalization System for Social Media Language,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, no. July, 2012, pp. 1035–1044.
- [7] J. Ellen, “All about Microtext-A Working Definition and a Survey of Current Microtext Research within Artificial Intelligence and Natural Language Processing.” *ICAART (1)*, vol. 2011, pp. 329–336, 2011.

- [8] E. Cambria, H. Wang, and B. White, “Guest editorial: Big social data analysis,” *Knowledge-Based Systems*, vol. 69, pp. 1–2, 2014.
- [9] J. Read, “Using emoticons to reduce dependency in machine learning techniques for sentiment classification,” in *Proceedings of the ACL student research workshop*. Association for Computational Linguistics, 2005, pp. 43–48.
- [10] Z. Xue, D. Yin, and B. D. Davison, “Normalizing Microtext,” *Analyzing Microtext*, pp. 74–79, 2011.
- [11] F. Liu, F. Weng, B. Wang, and Y. Liu, “Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision,” *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 71–76, 2011.
- [12] S. Petrović, M. Osborne, and V. Lavrenko, “The Edinburgh Twitter corpus,” in *Proceedings of the NAACL HLT Workshop on Computational Linguistics in a World of Social Media*, 2010, pp. 25–26.
- [13] R. Khoury, “Microtext Normalization using Probably-Phonetically-Similar Word Discovery,” in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on.*, 2015, pp. 392–399.
- [14] N. Mishra, M. Ramanathan, R. Satapathy, E. Cambria, and N. Magnenat-Thalmann, “Can a humanoid robot be part of the organizational workforce? a user study leveraging sentiment analysis,” in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–7.
- [15] J. H. Kietzmann, K. Hermkens, I. P. McCarthy, and B. S. Silvestre, “Social media? get serious! understanding the functional building blocks of social media,” *Business horizons*, vol. 54, no. 3, pp. 241–251, 2011.

- [24] R. Satapathy, C. Guerreiro, I. Chaturvedi, and E. Cambria, “Phonetic-Based Microtext Normalization for Twitter Sentiment Analysis,” in *ICDM*, 2017, pp. 407–413.
- [25] C. Zhang, T. Baldwin, H. Ho, B. Kimelfeld, and Y. Li, “Adaptive parser-centric text normalization,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 1159–1168.
- [26] T. A. Pirinen and K. Lindén, “State-of-the-art in weighted finite-state spell-checking,” in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2014, pp. 519–532.
- [27] R. Beaufort, S. Roekhaut, L.-A. l. Cougnon, and C. d. Fairon, “A hybrid rule/model-based finite-state framework for normalizing SMS messages,” in *ACL*. Association for Computational Linguistics, 2010, pp. 770–779.
- [28] P. Norvig, “How to write a spelling corrector,” *De: <http://norvig.com/spell-correct.html>*, 2007.
- [29] T. A. Pirinen and S. Hardwick, “Effects of weighted finite-state language and error models on speed and efficiency of finite-state spell-checking,” in *Preproceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing FSMNLP*, 2012, pp. 6–14.
- [30] A. Wilcox-O’Hearn, G. Hirst, and A. Budanitsky, “Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model,” in *International conference on intelligent text processing and computational linguistics*. Springer, 2008, pp. 605–616.
- [31] G. Hirst and A. Budanitsky, “Correcting real-word spelling errors by restoring lexical cohesion,” *Natural Language Engineering*, vol. 11, no. 1, pp. 87–111, 2005.

- [32] R. Kohavi *et al.*, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *IJCAI*, 1995, pp. 1137–1145.
- [33] S. Jiampojarn, G. Kondrak, and T. Sherif, “Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion,” in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 372–379.
- [34] G. Chrupała, “Normalizing tweets with edit scripts and recurrent neural embeddings,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 2. Association for Computational Linguistics Baltimore, Maryland, 2014, pp. 680–686.
- [35] S. Leeman-Munk, J. Lester, and J. Cox, “Ncsu_sas_sam: Deep encoding and reconstruction for normalization of noisy text,” in *Proceedings of the Workshop on Noisy User-generated Text*, 2015, pp. 154–161.
- [36] D. L. Pennell and Y. Liu, “Normalization of informal text,” *Computer Speech & Language*, vol. 28, no. 1, pp. 256–277, 2014.
- [37] M. Lusetti, T. Ruzsics, A. Göhring, T. Samardžić, and E. Stark, “Encoder-decoder methods for text normalization,” in *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 18–28. [Online]. Available: <https://www.aclweb.org/anthology/W18-3902>
- [38] R. Satapathy, Y. Li, S. Cavallari, and E. Cambria, “Seq2Seq Deep Learning Models for Microtext Normalization,” in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.
- [39] S. Bartlett, G. Kondrak, and C. Cherry, “Automatic syllabification with structured svms for letter-to-phoneme conversion,” *Proceedings of ACL-08: HLT*, pp. 568–576, 2008.

- [40] C. Li and Y. Liu, “Normalization of text messages using character-and phone-based machine translation approaches,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012, pp. 2330–2333.
- [41] Y. Yang and J. Eisenstein, “A log-linear model for unsupervised text normalization,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 61–72.
- [42] I. Lourentzou, K. Manghnani, and C. Zhai, “Adapting sequence to sequence models for text normalization in social media,” *arXiv preprint arXiv:1904.06100*, 2019.
- [43] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar, and A. Basu, “Investigation and modeling of the structure of texting language,” *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 10, no. 3-4, pp. 157–174, 2007.
- [44] S. Yuan, J. Wu, L. Wang, and Q. Wang, “A hybrid method for multi-class sentiment analysis of micro-blogs,” in *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on*. IEEE, 2016, pp. 1–6.
- [45] B. Han and T. Baldwin, “Lexical normalisation of short text messages: Makn sens a# twitter,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 368–378.
- [46] D. Fossati and B. Di Eugenio, “I saw TREE trees in the park: How to Correct Real-Word Spelling Mistakes.” in *LREC*. Citeseer, 2008, pp. 896–901.
- [47] M.-T. Luong and C. D. Manning, “Achieving open vocabulary neural machine translation with hybrid word-character models,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1054–1063.

- [48] D. L. Pennell and Y. Liu, “A Character-Level Machine Translation Approach for Normalization of SMS Abbreviations,” in *IJCNLP*, 2011, pp. 974–982.
- [49] K. Xu, Y. Xia, and C.-H. Lee, “Tweet normalization with syllables,” 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)*, vol. 1, 2015, pp. 920–928.
- [50] C. Kobus, F. Yvon, and G. é. Damnati, “Normalizing SMS: are two metaphors better than one?” in *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1. Association for Computational Linguistics, 2008, pp. 441–448.
- [51] S. Hocq, “Étude des sms en français: constitution et exploitation d’un corpus aligné SMS-langue standard,” *Rapport interne, Université Aix-Marseille*, 2006.
- [52] C. Fairon and S. Paumier, “A translated corpus of 30,000 French SMS,” in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, 2006, pp. 351–354.
- [53] M. Kaufmann and J. Kalita, “Syntactic normalization of twitter messages,” in *International conference on natural language processing, Kharagpur, India*, 2010, p. 7.
- [54] P. Wang and H. T. Ng, “A Beam-Search Decoder for Normalization of Social Media Text with Application to Machine Translation.” in *HLT-NAACL*, 2013, pp. 471–481.
- [55] B. Han, P. Cook, and T. Baldwin, “Automatically constructing a normalisation dictionary for microblogs,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 421–432.

- [56] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.
- [57] W. Daelemans, J. Zavrel, K. Van Der Sloot, and A. Van den Bosch, “Timbl: Tilburg memory-based learner,” *Tilburg University*, 2004.
- [58] T. Baldwin, M.-C. de Marneffe, B. Han, Y.-B. Kim, A. Ritter, and W. Xu, “Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition,” in *Proceedings of the Workshop on Noisy User-generated Text*, 2015, pp. 126–135.
- [59] C. D. Manning, “Part-of-speech tagging from 97% to 100%: is it time for some linguistics?” in *International conference on intelligent text processing and computational linguistics*. Springer, 2011, pp. 171–189.
- [60] T. Luong, R. Socher, and C. Manning, “Better word representations with recursive neural networks for morphology,” in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 2013, pp. 104–113.
- [61] G. Bouma, “Finite state methods for hyphenation,” *Natural Language Engineering*, vol. 9, no. 1, pp. 5–20, 2003.
- [62] W. Skut, B. Krenn, T. Brants, and H. Uszkoreit, “An annotation scheme for free word order languages,” in *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics, 1997, pp. 88–95.
- [63] W. Daelemans and A. van den Bosch, “Generalization performance of back-propagation learning on a syllabification task,” in *Proceedings of the 3rd Twente Workshop on Language Technology*. Universiteit Twente, Enschede, 1992, pp. 27–38.
- [64] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, “Moses: Open source toolkit

- for statistical machine translation,” in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.
- [65] Y. How and M.-Y. Kan, “Optimizing predictive text entry for short message service on mobile phones,” in *Proceedings of HCI*, vol. 5, 2005, p. .
- [66] C. Thurlow and A. Brown, “Generation txt? the sociolinguistics of young people’s text-messaging,” *Discourse analysis online*, vol. 1, no. 1, p. 30, 2003.
- [67] R. Khoury, “Phonetic normalization of microtext,” in *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*. IEEE, 2015, pp. 1600–1601.
- [68] V. Jahjah, R. Khoury, and L. Lamontagne, “Word Normalization Using Phonetic Signatures,” in *Advances in Artificial Intelligence*, R. Khoury and C. Drummond, Eds. Springer International Publishing, 2016, pp. 180–185.
- [69] R. Satapathy, A. Singh, and E. Cambria, “PhonSenticNet: A Cognitive Approach to Microtext Normalization for Concept-Level Sentiment Analysis,” in *Computational Data and Social Networks*. Cham: Springer International Publishing, 2019, pp. 177–188.
- [70] M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” *Speech communication*, vol. 50, no. 5, pp. 434–451, 2008.
- [71] A. Laurent, P. Deléglise, and S. Meignier, “Grapheme to phoneme conversion using an smt system,” in *Tenth Annual Conference of the International Speech Communication Association*, 2009, pp. 708–711.
- [72] K. Rao, F. Peng, H. Sak, and F. Beaufays, “Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4225–4229.

- [81] M. Lusetti, T. Ruzsics, A. Göhring, T. Samardžić, and E. Stark, “Encoder-decoder methods for text normalization,” in *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Association for Computational Linguistics, Aug. 2018, pp. 18–28.
- [82] S. Schulz, G. D. Pauw, O. D. Clercq, B. Desmet, V. Hoste, W. Daelemans, and L. Macken, “Multimodular text normalization of dutch user-generated content,” *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 4, Jul. 2016.
- [83] C. Haruechaiyasak and A. Kongthon, “LexToPlus: A Thai lexeme tokenization and normalization tool,” in *Proceedings of the 4th Workshop on South and Southeast Asian Natural Language Processing*. Nagoya Congress Center, Nagoya, Japan: Asian Federation of Natural Language Processing, Oct. 2013, pp. 9–16.
- [84] R. Satapathy, C. Guerreiro, I. Chaturvedi, and E. Cambria, “Phonetic-based microtext normalization for twitter sentiment analysis,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 407–413.
- [85] O. De Clercq, S. Schulz, B. Desmet, and V. Hoste, “Towards shared datasets for normalization research,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 1218–1223.
- [86] A. Radford, R. Jozefowicz, and I. Sutskever, “Learning to generate reviews and discovering sentiment,” *arXiv preprint arXiv:1704.01444*, 2017.
- [87] A. Graves, “Generating sequences with recurrent neural networks,” *arXiv preprint arXiv:1308.0850*, 2013.
- [88] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2741–2749.

- [89] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [90] A. Karpathy, J. Johnson, and L. Fei-Fei, “Visualizing and understanding recurrent networks,” *arXiv preprint arXiv:1506.02078*, 2015.
- [91] J. Villena Román, S. Lana Serrano, E. Martínez Cámara, and J. C. González Cristóbal, “Tass-workshop on sentiment analysis at sepln,” in *Proceedings of the TASS workshop at SEPLN*, 2013.
- [92] S. Agarwal, S. Godbole, D. Punjani, and S. Roy, “How much noise is too much: A study in automatic text classification,” in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 3–12.
- [93] T. Bayer, U. Kressel, H. Mogg-Schneider, and I. Renz, “Categorizing paper documents: a generic system for domain and language independent text categorization,” *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 299–306, 1998.
- [94] H. Jing, D. Lopresti, and C. Shih, “Summarizing noisy documents,” in *Proceedings of the Symposium on Document Image Understanding Technology*, 2003, pp. 111–119.
- [95] K. Taghva, J. Borsack, and A. Condit, “Effects of OCR errors on ranking and feedback using the vector space model,” *Information processing & management*, vol. 32, no. 3, pp. 317–327, 1996.
- [96] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel, “Named entity extraction from noisy input: speech and OCR,” in *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 2000, pp. 316–324.

- [97] E. Cambria, “Affective Computing and Sentiment Analysis,” *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.
- [98] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, “A review of affective computing: From unimodal analysis to multimodal fusion,” *Information Fusion*, vol. 37, pp. 98–125, 2017.
- [99] I. Chaturvedi, E. Cambria, R. Welsch, and F. Herrera, “Distinguishing between facts and opinions for sentiment analysis: Survey and challenges,” *Information Fusion*, vol. 44, pp. 65–77, 2018.
- [100] S. L. Lo, E. Cambria, R. Chiong, and D. Cornforth, “Multilingual sentiment analysis: From formal to informal and scarce resource languages,” *Artificial Intelligence Review*, vol. 48, no. 4, pp. 499–527, 2017.
- [101] H. Peng, Y. Ma, Y. Li, and E. Cambria, “Learning Multi-grained Aspect Target Sequence for Chinese Sentiment Analysis,” *Knowledge-Based Systems*, vol. 148, pp. 167–176, 2018.
- [102] R. Gopalakrishna Pillai, M. Thelwall, and C. Orasan, “Detection of Stress and Relaxation Magnitudes for Tweets,” in *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 2018, pp. 1677–1684.
- [103] I. Chaturvedi, E. Cambria, F. Zhu, L. Qiu, and W. K. Ng, “Multilingual subjectivity detection using deep multiple kernel learning,” *Proceedings of Knowledge Discovery and Data Mining, Sydney*, 2015.
- [104] B. Frénay and M. Verleysen, “Reinforced extreme learning machines for fast robust regression in the presence of outliers,” *IEEE Trans. Cybernetics*, vol. 46, no. 12, pp. 3351–3363, 2016.
- [105] J. Wiebe, T. Wilson, and C. Cardie, “Annotating expressions of opinions and emotions in language,” *Language resources and evaluation*, vol. 39, no. 2, pp. 165–210, 2005.

- [106] L. Yujian and L. Bo, “A normalized Levenshtein distance metric,” *IEEE Trans Pattern Anal Mach Intell*, vol. 29, no. 6, pp. 1091–1095, 2007.
- [107] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, “SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 1795–1802.
- [108] S. Poria, E. Cambria, G. W. Winterstein, and G.-B. Huang, “Sentic patterns: Dependency-based rules for concept-level sentiment analysis.” *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.
- [109] R. S. Wallace, “The anatomy of alice,” in *Parsing the Turing Test*. Springer, 2009, pp. 181–210.
- [110] A. Beider and S. P. Morse, “Beider-Morse phonetic matching: An alternative to Soundex with fewer false hits,” *Avotaynu: the International Review of Jewish*, 2008.
- [111] Z. Bhatti, I. A. Ismaili, A. A. Shaikh, and W. Javaid, “Spelling Error Trends and Patterns in Sindhi,” *CoRR*, 2014.
- [112] T. Naseem and S. Hussain, “Spelling Error Trends in Urdu,” in *Proceedings of Conference on Language Technology*, 2007.
- [113] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.
- [114] D. R. Mortensen, S. Dalmia, and P. Littell, “Epitran: Precision G2P for Many Languages,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Paris, France: European Language Resources Association (ELRA), May 2018, pp. 7–12.

- [115] J. Ramos *et al.*, “Using tf-idf to determine word relevance in document queries,” in *Proceedings of the first instructional conference on machine learning*, vol. 242, 2003, pp. 133–142.
- [116] A. Brown, *Singapore English in a nutshell: An alphabetical description of its features*. Federal Publications, 1999.
- [117] M. Warschauer, “The internet and linguistic pluralism,” *Silicon literacies: Communication, innovation and education in the electronic age*, pp. 62–74, 2002.
- [118] D. Rajagopal, E. Cambria, D. Olsher, and K. Kwok, “A graph-based approach to commonsense concept extraction and semantic similarity detection,” in *International World Wide Web Conference*, Rio De Janeiro, 2013, pp. 565–570.
- [119] A. Qazi, K. Syed, R. Raj, E. Cambria, M. Tahir, and D. Alghazzawi, “A concept-level approach to the analysis of online review helpfulness,” *Computers in Human Behavior*, vol. 58, pp. 75–81, 2016.
- [120] N. Howard and E. Cambria, “Intention awareness: Improving upon situation awareness in human-centric environments,” *Human-centric Computing and Information Sciences*, vol. 3, no. 9, 2013.
- [121] E. Cambria, A. Hussain, T. Durrani, C. Havasi, C. Eckl, and J. Munro, “Sentic computing for patient centered application,” in *IEEE International Conference on Information Communication and Signal Processing*, 2010, pp. 1279–1282.
- [122] E. Cambria, T. Benson, C. Eckl, and A. Hussain, “Sentic PROMs: Application of sentic computing to the development of a novel unified framework for measuring health-care quality,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 10 533–10 543, 2012.
- [123] J. Eisenstein, “What to do about bad language on the internet,” in *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 359–369.

- [124] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *3rd International Conference for Learning Representations (ICLR)*, 2015.
- [125] W. Min and B. Mott, “Ncsu.sas.wookhee: A deep contextual long-short term memory model for text normalization,” in *Proceedings of the Workshop on Noisy User-generated Text*, 2015, pp. 111–119.
- [126] M. Luong, E. Brevdo, and R. Zhao, “Neural Machine Translation (seq2seq) Tutorial,” <https://github.com/tensorflow/nmt>, 2017.
- [127] M.-T. Luong, H. Pham, and C. D. Manning, “Effective Approaches to Attention-based Neural Machine Translation,” in *Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 1412–1421. [Online]. Available: <http://aclweb.org/anthology/D15-1166>
- [128] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *arXiv e-prints*, vol. abs/1412.3555, 2014, presented at the Deep Learning workshop at NIPS2014. [Online]. Available: <https://arxiv.org/abs/1412.3555>
- [129] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [130] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.
- [131] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

- [132] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in *COLING 2018, 27th International Conference on Computational Linguistics*, 2018, pp. 1638–1649.
- [133] E. Cambria and A. Hussain, “Sentic computing,” *Cognitive Computation*, vol. 7, no. 2, pp. 183–185, 2015.
- [134] C. Myers-Scotton, *Code-Switching*. John Wiley and Sons Ltd, 2017, ch. 13, pp. 217–237. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781405166256.ch13>
- [135] D. Vilares, H. Peng, R. Satapathy, and E. Cambria, “Babelsenticnet: A commonsense reasoning framework for multilingual sentiment analysis,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1292–1298.
- [136] K. H. Jamieson and J. N. Cappella, *Echo chamber: Rush Limbaugh and the conservative media establishment*. Oxford University Press, 2008.
- [137] X. Zhou, R. Zafarani, K. Shu, and H. Liu, “Fake news: Fundamental theories, detection strategies and challenges,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 2019, pp. 836–837.
- [138] D. S. Robertson *et al.*, *Phase Change: The computer revolution in science and mathematics*. Oxford University Press, USA, 2003.
- [139] J. Molyneux, “Greek Lyric, Vol. III Stesichorus, Ibycus, Simonides, and Others ed. by David A. Campbell,” *Echos du monde classique: Classical views*, vol. 37, no. 3, pp. 490–493, 1993.
- [140] M. (Psellus.), “De operatione daemonum,” *A.M. Hakkert*, p. 2, 1964.
- [141] Plutarch, “Moralia,” *Harvard University Press*, vol. 4, p. 500, 1936.