Analysis of Code-Mixed Social Media Text

A thesis submitted in partial fulfilment of the requirements for the Degree of

Master of Computer Science and Engineering

By

Souvick Ghosh

Examination Roll No: M4CSE1603
Class Roll No: 001410502003
Registration No: 100249 of 2007-08

Under the Esteemed Guidance of

Dr. Dipankar Das

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JADAVPUR UNIVERSITY, INDIA

May, 2016.

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

KOLKATA-700032

TO WHOM IT MAY CONCERN

This is to certify that the dissertation entitled "**Analysis of Code-Mixed Social Media Text**" has been carried out by **Souvick Ghosh** (Reg. No. 100249 of 2007-08, Class Roll No. 001410502003 and Exam Roll No. M4CSE1603), under my guidance and supervision and may be accepted in partial fulfilment of the requirement for the Degree of **Master of Computer Science and Engineering** in the Faculty of Engineering and Technology, Jadavpur University. The research results presented in this thesis have not been included in any paper submitted for the award of any degree in any other University or Institute.

_____

(Dr. Dipankar Das)
Thesis Supervisor,
Department of Computer Science and Engineering,
Jadavpur University, Kolkata- 700032.

Countersigned:

_____

(Prof. Debesh Kumar Das)
Head,
Department of Computer Science and Engineering,
Jadavpur University, Kolkata- 700032.

_____

(Prof. Sivaji Bandyopadhyay)
Dean,
Faculty of Engineering and Technology,
Jadavpur University, Kolkata- 700032.

FACULTY OF ENGINEERING & TECHNOLOGY

JADAVPUR UNIVERSITY

KOLKATA-700032

CERTIFICATE OF APPROVAL*

This is to certify that the thesis entitled "**Analysis of Code-Mixed Social Media Text**" is a bona-fide record of work carried out by **Souvick Ghosh** in partial fulfilment of the requirements for the award of the degree of Master of Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University during the period of June 2015 to May 2016. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose for which it is submitted.

1._____

FINAL EXAMINATION

FOR EVALUATION

OF THESIS

2._____

(Signature of Examiners)

*Only in case the thesis is approved

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis "**Analysis of Code-Mixed Social Media Text**" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Computer Science and Engineering.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name: Souvick Ghosh

Examination Roll Number: M4CSE1603
Registration Number: 100249 of 2007-08
Thesis Title: **Analysis of Code-Mixed Social Media Text**

Signature with Date:

To My Family

Thanks for being the wind beneath my wings!

Acknowledgements

I would like to express my deepest appreciation to all whose valuable suggestion and cooperation provided me the possibility of completing my study.

Foremost I take this opportunity to express my gratitude to my advisor and respected guide, **Dr. Dipankar Das**, Department of Computer Science & Engineering, Jadavpur University, for the tremendous support, motivation, enthusiasm and guidance rendered to me during the course of my study. I could not have imagined a better advisor and mentor for my study. I would like to thank **Prof. Sivaji Bandyopadhyay**, one of the pioneers of NLP studies in India. His immense knowledge and support was a motivation and source of inspiration for me. I would also like to thank **Dr. Sudip Kumar Naskar** for his help and support during the course of study. Each of them has been instrumental in the successful completion of this project.

The support and cheerful company I received from Mr. Tapas Nayak, Mr. Joy Mahapatra, Mr. Ritesh Sarkhel, Ms. Promita Maitra, Mr. Tapabrata Mondal, Mr. Braja Gopal Patra and Mr. Alapan Kulia can never be denied. I would like to express my heartfelt thanks to all laboratory staff, who have been crucial in the course of my study.

Finally, I would like to thank my brother, Mr. Satanu Ghosh, who was an able collaborator in some of the works. I would also like to thank all my family members and loved ones, who have supported me throughout the entire process, both by keeping me harmonious and helping me putting the pieces together. I will be grateful forever for your love.

Each of you can share this accomplishment, for without your support it would not have been possible.

Souvick Ghosh

30th May, 2016.

Table of Contents

List of Figures

List of Tables

Abstract

Ever since the advent of civilization, human beings have shown a proclivity towards knowledge gathering and sharing. From images drawn on walls of the cave to texts written in papyrus, knowledge has always found a way to permeate from one generation to the other, from one civilization to another. The fact that knowledge means power been etched into the basic constitution of human mentality.

In modern times, the rapid advancement of technology has opened doors for processing large volumes of images, texts and speech. The information contained in each of these communication media can be put to a wide variety of uses. For example, climatic patterns can be identified by processing satellite images, number plates on cars can be recognised using digit recognition software, texts can be scanned for threat detection and speech can be analysed for mood prediction.

Our work mainly centres on the diverse applications of text analytics. Text analytics[1] refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. However, our study is not an all encompassing attempt at unveiling all the hidden gems of text analytics. Rather, we concentrate on a focussed area – social media analysis.

Social media[2] – the likes of Facebook, Twitter - are computer-mediated tools that allow people or companies to create, share, or exchange information, career interests, ideas, and pictures/videos in virtual communities and networks. Analysis of social media involves uncovering various hidden patterns and user sentiments dispersed across the online sources. For example, an organization may be interested to know the feelings of customers towards the organization; a car company may want to gauge the consumer sentiment before launching its new car and so on. Overall, the vast knowledge scattered in the web space has immense potential. The scope of application can range from movie reviews, and electoral result prediction to marketing and advertisements to detection of cyberbullying and national threats.

[1] https://en.wikipedia.org/wiki/Text_mining
[2] https://en.wikipedia.org/wiki/Social_media

While various researches have already been conducted in social media analytics, we focus on a particularly interesting area – multilingual transliterated social media text. As social media platforms have spread across the world, it has brought in users from non-native English background. These users often prefer to use their native language in addition to English. Also, the text is not written in native scripts. They are transliterated in Roman script instead. In our work, we have done a five-fold analysis of social media text. Starting from word level language identification, we have tagged the words with their respective parts-of-speech and have identified the named entities along with their types. We have performed sentiment analysis of this multilingual social media text – which is a first of its kind – and lastly, we propose an index which compares the level of complexity between different social media corpus.

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction

1.1 Data Analytics

Analysis refers to breaking a whole into its separate components for individual examination. Analysis of data is the science of inspecting, examining, cleaning, transforming, and modelling raw data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, in different business, science, and social science domains[3].

Statistician John Tukey (Tukey, 1962) defined data analysis in 1961 as:

"Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data."

1.2 Text Analytics

A traditional text analytics framework consists of three consecutive phases: Text Pre-processing, Text Representation and Knowledge Discovery, shown in Figure 1.2. Text pre-processing aims to make the input documents more consistent to facilitate text representation, which is necessary for most text analytics tasks (Hu and Liu, 2012). Traditional text pre-processing methods include stop word removal and stemming. Removal of stop words ensures that all meaningless and common words are removed from the text. Some examples of common stop words are '*a*', '*the*', '*them*', '*you*', etc. These words carry no important information relevant to analysis and hence, are removed before further processing of data. However, some applications of NLP deliberately avoid the removal of stop words to facilitate phrase search. Stemming, on the other hand, reduces words to their root or base form by removing inflection from the word. For example, the words '*universal*', '*university*' and '*universe*' are all reduced to the word '*univers*' by the stemmer.

[3] https://en.wikipedia.org/wiki/Data_analysis

Stemming recognises the fact that different variation of the same root word is identical in their meaning. Text analytics can be used for a large number of applications ranging from plagiarism detection to forensics (like authorship identification and verification) to medicine (detection of epidemics, mental growth in infants, etc.).

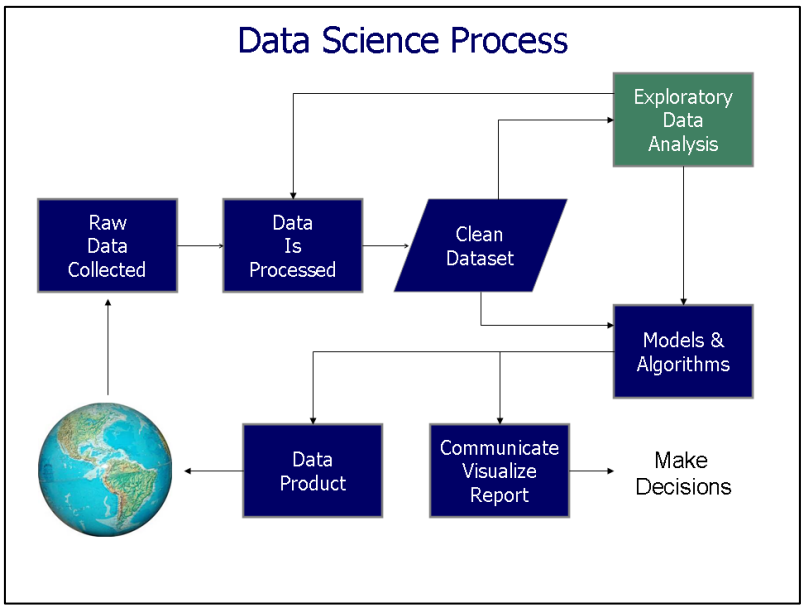

Fig 1.1. Data Science Process. (Source: Wikipedia)

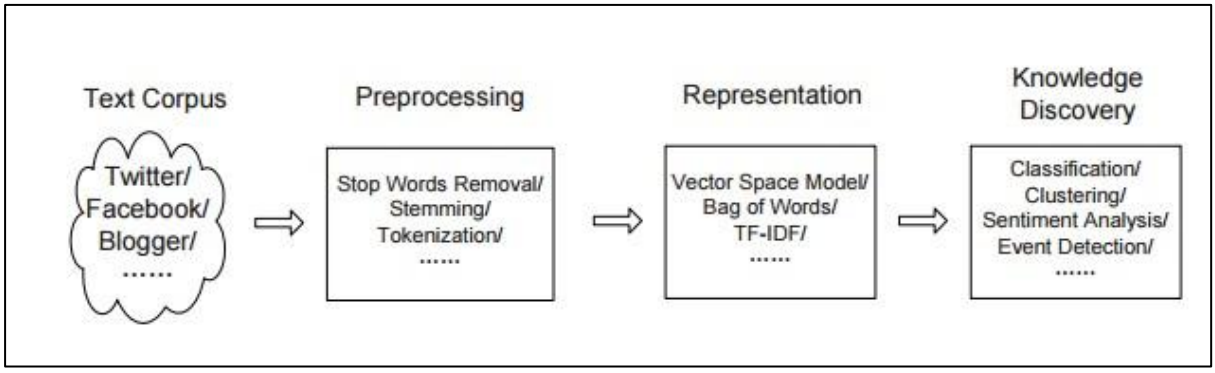

Fig 1.2. A Traditional Framework for Text Analytics (Source: X. Hu and H. Liu).

1.3 Text Analytics of Social Media

The term social media is relatively new. According to Wikipedia[4], which is an Internet encyclopaedia, the term Social Media can be defined as follows:

"*Social media are computer-mediated tools that allow people or companies to create, share, or exchange information, career interests, ideas, and pictures/videos in virtual communities and networks.*"

[4] https://en.wikipedia.org/wiki/Social_media

Social media technologies take on many different forms including blogs, business networks, enterprise social networks, forums, microblogs, photo sharing, products/services review, social bookmarking, social gaming, social networks, video sharing, and virtual worlds (Aichner and Jacob, 2015). The different types of social media are illustrated in Fig. 1.3.

Traditional media like newspaper, radio, television and publications is unidirectional in nature. The flow of information is from the author, writer or business to the reader or consumer. Social media, on the other hand, creates a virtual community online where different users can interact with each other, engage in dialogues and collaborative activities, thus, allowing the information to have a bi-directional and multi-directional flow.

With the advent of Facebook[5] in 2004 and Twitter[6] in 2006, the world witnessed the arrival of a new era in terms of social interaction, opinion sharing and advertising. This has presented the computer analysts and linguistics with an interesting and complex problem of analyzing such data. With computers becoming cheaper and more portable, internet becoming fast and ubiquitous, social media like Facebook, Twitter, etc. have attracted people from all walks of life.

Social media analytics make use of the fast growing nature of social media platforms. Various applications of data and text analytics to social media are detection of events as they are happening in world (captured through live feeds and trending in Facebook and hashtag and trends in Twitter). Google News[7] provides web-based aggregated service. The detection of news events and subsequently ranking them in the order of importance has drawn a lot of interest from research communities. There are a number of Question-Answering (QA) services - Reddit[8] which is a popular entertainment and social news networking service and Quora[9] which is a question-answer website where questions are asked, answered, edited and organized by a large community of users) – where users can search a question directly from the archive or ask new questions. This sort of collaborative QA sites allow for opinion mining on a large scale. Similarly, there are professional networking sites (like LinkedIn[10]) which allows for user profiling and categorization. Social media can be utilized for analyzing user behaviour, consumer and political opinions, creating reviews, building knowledge graphs and so on.

[5] https://www.facebook.com/
[6] https://twitter.com/
[7] https://news.google.com/
[8] https://www.reddit.com/
[9] https://www.quora.com/
[10] https://in.linkedin.com/

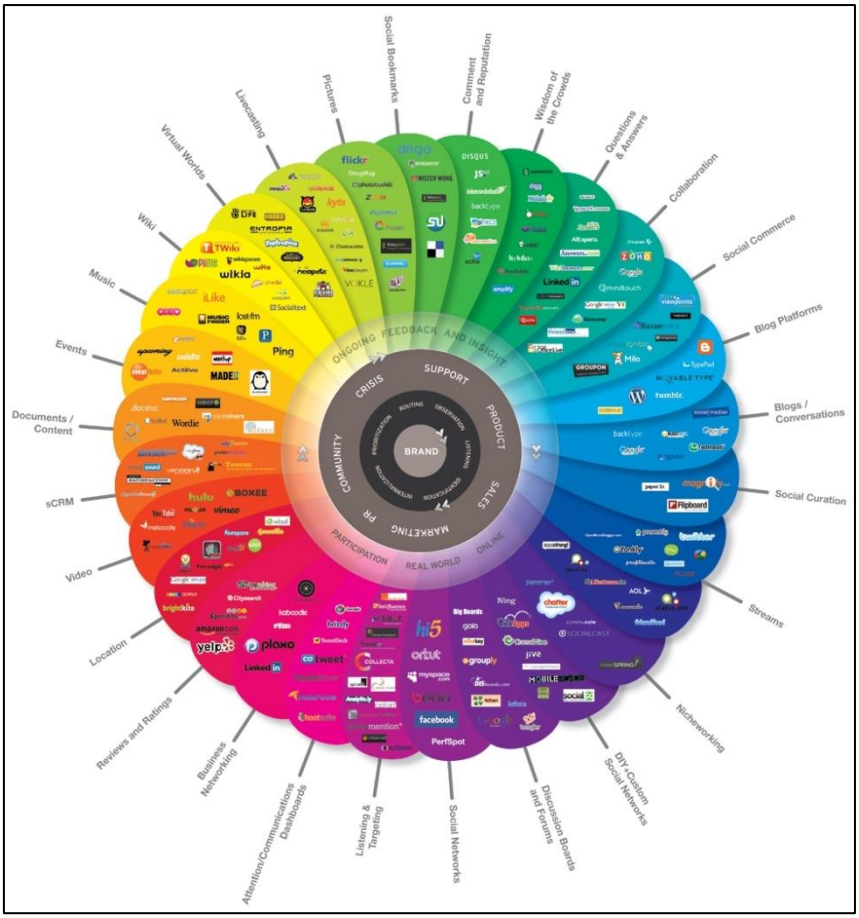

Fig 1.3. Different Types of Social Media (Source: Wikipedia)

1.4 Complexities of Social Media Text

Social media text differs from conventional text (as used in literary works and newspapers) in a variety of ways. The following section describes the complexities of social media text with respect to the other more traditional forms.

- ***Code-Switching And Code-Mixing***

  Often used interchangeably, these two terms refer languages shifts in multilingual social media text. Code mixing is the phenomenon where more than one, usually only two, languages is used in a single tweet or post. Some researchers believe that code mixing is intra-sentential, that is, language changes occur inside a sentence, while code-switching is inter-sentential. Both these phenomenon are common in geographical regions with a high percentage of bilingual individuals. (More details in Section 1.5)

- ***Lexical Borrowings***

  Adoption of individual words from one language to another.

E.g. *café* (from French meaning '*coffee*'), *déjà vu* (French word), *Kindergarten* (from German) are used as part of English vocabulary.

- ***Phonetic Typing***

  The phenomenon of typing words in one language based on how they are pronounced in another language.

  For example, the phrase '*kya chal raha he*' (meaning *what is up?*) is originally in Hindi but has been written in English based on its pronunciation.

- ***Abbreviations***

  Often larger phrases are replaced by short forms, comprising the first letter of each component word.

  E.g.: *OMG* for 'Oh my God!'

- ***Random Contractions***

  Words are truncated into smaller forms.

  E.g.: *'em* in place of *them* , *shan't* in place of *shall not.*

- ***Transliterations***

  Words are written in non-native scripts. Many languages that use non-Roman scripts, like Hindi, Bangla, Chinese, Arabic etc. are often present in a Romanized form [1].

  E.g.: *asombhob* (meaning *impossible*) is a Bengali word written in English.

- ***Spelling Variations***

  Social media text involves a variety of irregular spellings for common words.

  For example, *gr8* is used for '*great'; f9* is used for *'fine'*, and so on.

- ***Style Of Writing***

  The writing does not follow any specific structure as in formal texts.

- ***Brevity Of The Texts***

  With the arrival microposts like Twitter, there is a limit to the maximum number of words that can be used in a post. This makes the text more noisy and irregular.

  E.g.: Traditionally, tweets are 140 characters long and Picasa[11] comments are 512 characters long.

[11] http://picasa.google.com/

- ***Time Sensitivity***

  Most of the social media platforms allow live chats, posts and dialogues in real time. Users share their daily stories as they occur, communicating instantly with other users while on the fly.

- ***Lack Of Gold Standard Data***

  There is a lack of code-mixed data in conventional text-corpora. Researchers need to create data sets which contain code-mixed data exclusively. Most of the time, the corpus needs to be created through crowd-sourcing and hand labelling. However, this makes it difficult to use data-intensive methods for analysis.

1.5 What is Code-Mixing and Code-Switching?

A *code* may be a language or a variety or style of a language. The term *code-mixing* emphasizes hybridization, and the term *code-switching* emphasizes movement from one language to another. According to Wikipedia,

"Code-mixing refers to the mixing of two or more languages or language varieties in speech."

"In linguistics, code-switching occurs when a speaker alternates between two or more languages, or language varieties, in the context of a single conversation. Multilinguals, speakers of more than one language, sometimes use elements of multiple languages when conversing with each other. Thus, code-switching is the use of more than one linguistic variety in a manner consistent with the syntax and phonology of each variety."

Some scholars use the terms "code-mixing" and "code-switching" interchangeably while others assume more specific definitions of code-mixing and code-switching. Code-mixing and Code-switching primarily occur within a multilingual setting where speakers share more than one language.

There are four major types of switching[12]:

- Tag-switching, in which tags and certain set phrases in one language are inserted into an utterance otherwise in another, as when a Panjabi/English bilingual says: *It's a nice day*, *hana?* (*It's a nice day, isn't it?*).

- Intra-sentential switching, in which switches occur within a clause or sentence boundary, as when a Yoruba/English bilingual says: *Won o arrest a single person* (*won o* they did not).

[12] http://www.encyclopedia.com/doc/1O29-CODEMIXINGANDCODESWITCHNG.html

- Inter-sentential switching, in which a change of language occurs at a clause or sentence boundary, where each clause or sentence is in one language or the other, as when a Spanish/English bilingual says: *Sometimes I'll start a sentence in English y termino en español* (and finish it in Spanish). This last may also occur as speakers take turns.

- Intra-word switching, in which a change occurs within a word boundary, such as in *shoppã* (English *shop* with the Panjabi plural ending) or *childrener* (English *children* with the Bengali suffix er, meaning 'of').

1.6 Thesis Contribution

The main contributions of this thesis are the development of tools to analyze different aspects of code-mixed social media text. We have also developed various indexes to measure the complexity of a given social media corpora.

As explained in previous sections, social media text differs greatly from standard texts which we often encounter in newspapers, articles and books. With further addition of code-mixing to social media text, the complexity increases manifold. The available state-of-the-art tools do not work satisfactorily with code-mixed data. In our work, we have processed raw text by first identifying the various languages present in it. We took part in the FIRE[13] shared task which involved eight Indian languages along with English. We developed a system using Conditional Random Field which achieves an overall accuracy of 75.5% for token level language identification.

A common step in the processing of any text is the part-of-speech tagging of the input text. Our participation in the ICON[14] shared task enabled us to tackle code-mixed text from three different languages – Bengali, Hindi and Tamil – apart from English. Once again, our system used Conditional Random Field – a sequence learning method which is useful to capture patterns of sequences containing code switching – along with various pre-processing and post-processing modules to tag each word with accurate part-of-speech information. Our system performed satisfactorily, with 75.22% accuracy in Bengali-English mixed data.

One of the important steps in any text processing is the identification of Named Entities present in the text. In various tasks like Question Answering, Text Summarization and Event Similarity Detection, the identification of named entities is most essential. We took part in the Named

[13] http://research.microsoft.com/en-us/events/fire13_st_on_transliteratedsearch/fire15st.aspx
[14] http://ltrc.iiit.ac.in/icon2015/

Entity rEcognition and Linking Challenge (NEEL 2016)[15] which was a part of the #Microposts2016 Workshop at the World Wide Web 2016 conference. In this task, we identified the various named entities present in tweets and performed a 7-fold classification - Thing, Event, Character, Location, Organization, Person and Product - of named entities. We also linked the entity mentions to an existing knowledge base – DBPedia[16]. Our best performing system used Feed forward neural network for classification.

One of the most interesting applications of social media analysis is sentiment analysis. While some works have been done in code-mixed social media data and in sentiment analysis separately, our work is the first attempt (as of now) which aims at performing sentiment analysis of code-mixed social media text. We have used extensive pre-processing to remove noise from raw text. Multilayer Perceptron model has been used to determine the polarity of the sentiment. We have also developed the corpus for this task by manually labelling Facebook posts with their associated sentiments.

Lastly, we have proposed an evaluation index – Complexity Factor – which evaluates the complexity of a given social media corpus. The complexity evaluation is mainly from the perspective of code-mixing and would help in comparing between two or more social media corpora.

It must be noted that the methodology that we employed in all the five tasks can be used for any resource poor language. We adapted standard learning approaches that work well with scarce data. We have also ensured that the algorithms are portable to different platforms and languages and can be deployed for real time analysis.

1.7 Introduction to Later Chapters

In Chapter 2, we present our work on Language Identification. We took part in the Query Word Labelling subtask organized by Forum for Information Retrieval Evaluation (FIRE 2015). The task involved identifying the language of individual query words. In our work, we follow the footsteps of some recent works done in word level language identification. We develop a system of language identification using word level classification approach using various dictionary and style based features.

[15] http://microposts2016.seas.upenn.edu/challenge.html
[16] wiki.dbpedia.org/

Chapter 3 deals with part-of-speech (POS) tagging of social media text. To analyze any language, POS tagging is one of the fundamental pre-processing steps. For POS tagging of multilingual text, we have used a simple strategy – to divide the text into chunks belonging to same language and use monolingual POS taggers to tag each segment separately. Our system uses Conditional Random Field – a sequential classifier – together with a pre-processing module (which involves chunking or separation of text into monolingual segments) and a post-processing module (to remove ambiguity).

Chapter 4 is a brief review of our attempts in the Named Entity rEcognition and Linking Challenge (NEEL) at the #Microposts2016. The task is to automatically recognize entities and their types from English microposts, and link them to corresponding DBpedia 2015 entries. We use an ensemble method for identification of named entities. The classification of named entities into various types is achieved using feed forward neural network model with five hidden layers. We also link the named entities to an existing knowledge base to augment with more contextual and semantic information.

Chapter 5 presents a novel approach to sentiment detection and sentiment polarity classification in multilingual social media content. It is the first attempt at capturing sentiment information in code-mixed social media data. Social media is ideal for mining predictive models. Sentiment analysis has a widespread application in marketing, advertising, trend prediction, recommendation systems and threat detection. We use a Multilayer Perceptron model with feedback to classify sentiment polarity. We use a variety of sentiment resources and remove noise through extensive pre-processing of the data.

In Chapter 6, we also propose an extension to the existing code-mixing index to evaluate the complexity of the code-mixed text. The index attempts to capture the level of intermingling of language in the text. It could also be used to compare the performance of various systems which are being developed for separating multiple languages. We discuss the merits and shortcomings of the existing indexes and present a new index which captures the complexity of a text in terms of language mixing, frequency of language shifts and proportion of different languages in a given text.

In Chapter 7, we provide a brief conclusion to each of the works performed as part of this thesis. We also provide an insight into future works and suggested improvements.

CHAPTER

2 LANGUAGE IDENTIFICATION OF CODE-MIXED SOCIAL MEDIA TEXT

Chapter 2

Language Identification of Code-Mixed Social Media Text

In this Chapter, we describe our approach on Query Word Labeling as an attempt in the shared task on Mixed Script Information Retrieval at Forum for Information Retrieval Evaluation (FIRE)[17] 2015. The query is written in Roman script and the words were in English or transliterated from Indian regional languages. A total of eight Indian languages are present in addition to English. We also identify the Named Entities and special symbols as part of our task. A Conditional Random Field (CRF) based machine learning framework is used for labeling the individual words with their corresponding language labels. We use a dictionary based approach for language identification. We also take into account the context of the word while identifying the language. Our system demonstrated an overall accuracy of 75.5% for token level language identification. The strict F-measure scores for the identification of token level language labels for Bengali, English and Hindi are 0.7486, 0.892 and 0.7972 respectively. The overall weighted F-measure of our system was 0.7498.

2.1 Introduction

Language Identification (LI) is a necessary prerequisite for processing any user generated text, where the languages are unknown. The identification of the language can be done at document level or at word level. Previously, language identification involved identifying the (single) overall language of full documents. In modern day, the documents are decreasing in size. The paper copies have been replaced by soft electronic copies and most researches obtain the data from online media. Such data are usually collected automatically using methods like bootstrapping. But the data is noisy in nature which requires use of different lexical resources. Social media has attracted users from all across the globe and not every user is a native English speaker. With geographical diversity, the languages used in social media have become diverse and complicated

[17] http://fire.irsi.res.in/fire/2015/home

as well. Phenomena such as code-switching, code-mixing, lexical borrowings, phonetic typing and transliterations do not allow for language identification on document level. We have to determine the language for every individual word.

Linguistic efforts in the field have mainly concentrated on the sociological and conversational necessity behind code-switching and code-mixing and its nature. For example, on whether it is an act of identity in a group or competence-related (i.e., a consequence of a lack of competence in one of the languages). However, the words are not represented in native script. Instead, they are represented phonetically in non-native script (the phenomenon is known as transliteration). Transliteration is complicated as there exists no accurate mapping to obtain the transliterated form of a word. Code-mixing, on the other hand, is the process of mixing more than one language in a single conversation. In social media texts like Facebook posts or Twitter tweets, code-mixing or switching implies the use of multiple languages in a single post or tweet.

India is a land of many languages with English and Hindi being the most popular languages nationwide. However, Indian society is hardly limited to two languages from a linguistic perspective. Each state and region has a different language of its own. As a result of that, social media data originating from any Indian usually contains two or more languages. In this Chapter, we explore techniques for performing language identification at the word level in mixed language documents. The work was performed as part of the FIRE shared task. The dataset contains 8 Indian languages along with English. The non-English words are transliterated in English. We use a dictionary-based approach. We use various dictionary based features and lexical resources (in English as well as native languages) in our work. We use contextual information of a word to determine its language. Various features have been used to train our model which used conditional random field for classification. We also identify the named entities which are language independent.

The rest of the Chapter is organized as follows. In Section 2.2, we discuss the related work. In Section 2.3, we describe the task objective of language identification along with the problem statement. The details of datasets and resources are presented in Section 2.4. Section 2.5 contains the description of our LI system and Section 2.6 details the features which were used as part of our language identification task. The results and observations are presented in Section 2.7. Section 2.8 is devoted to error analysis while Section 2.9 concludes the Chapter.

2.2 Related Work

Language identification has been recognized as a prominent area of work long time back (Gold, 1967) while Joshi (1982) worked on automatically identifying and analysing code switching in as early as 1980s. Beesley (1988) identified language identification of on-line text as one of the most difficult problems in machine translation. With the growth of online platforms and subsequent surge in language data, the problem of language identification has become an increasingly important issue.

Much of the initial work on language identification was performed on document level. As most of the documents were monolingual, the problem was reduced to closed-set classification problem. Researchers have proposed various approached for the task of language identification. N-gram features were used by Cavnar and Trenkle (1994), Markov models by Dunning (1994), Monte Carlo methods by Poutsma (2002) and so on.

However, automatic language identification for multilingual code-mixed texts has not received much attention before the last two decades. Any processing of code-mixed text needs to identify and label the words or groups of words which are in different languages. Hughes et al. (2006) conducted a survey of textual language identification. They pointed out that language identification has become challenging due to the lack of resources available for minority languages, the poor quality of training data and multilingual documents. Xia et al. (2009), in their work, pointed out that those current methods try to identify languages in large documents and have therefore, shown high accuracy. However, when the size of the document decreases or the number of languages in the document increases, the performance suffers heavily.

The proliferation of online communication media like email and chat and social media like Facebook and Twitter (Herring, 2003; Cardenas-Claros and Isharyanti, 2009; Paolillo, 2011) has ensured abundance of code-mixed data on the web. Social media data has several complexities (see Section 1.4). We have previously discussed on what is code-mixing and how it is different from code-switching (see Section 1.5). Previously, writings which involved code-switching and code-mixing were considered inferior in literary sense. It highlighted the lack of vocabulary and fluency of the writer in the language. The emergence of online forms of communication has led to abundance of such text in social and online media and has removed the stigma associated with it. Many scholars have deliberated over the motivation behind code-mixing and code-switching. (Milroy and Muysken, 1995) (Auer, 2013). These works mainly discuss the sociological, linguistic and conversational factors which motivate code mixing. Hidayat (2012) showed that 45% of

switching used by Facebook users are because of lexical needs, 40% for talking about a particular topic, and 5% for content clarification.

Solorio and Liu (2008) worked on automatically identifying code-switch points in a given document containing Spanish-English text. They worked on detecting code mixing in speech (Solorio and Liu, 2008a) and try to predict the code-switching points inside a set of spoken Spanish-English sentences (Solorio and Liu, 2008b). Other studies on code mixing are by Rosner and Farrugia (2007) on SMS messages and by Gottron and Lipka (2010) on information retrieval queries. The last few decades have seen the development of transliteration systems for Asian languages. Li (2000) and San (2009) worked on Chinese-English code mixing in Hong Kong and Macao respectively. They suggest that code-mixing is a result of linguistic motivations in a bilingual society. San (2009) compared the mixing in blog posts to that in the spoken language in Macao. Some notable transliteration systems were built for Chinese (Li et al., 2004), Japanese (Goto et al., 2003), Korean (Jung et al., 2000) and Arabic (Al-Onaizan and Knight, 2002). Transliteration systems were also developed for Indian languages (Ekbal et al., 2006; Sowmya et al., 2010; Surana and Singh, 2008).

A number of researches have been conducted to automatically identify languages in social media text. Yamaguchi and Tanaka-Ishii (2012) used artificial multilingual data, created by randomly sampling text segments from monolingual documents. King and Abney (2013) used a weakly supervised model, with a sequence labeller – Conditional Random Field – and monolingual text samples for training data. Nguyen and Dogruoz (2013) worked on Turkish and Dutch forum data. They identified various sources of errors – variations in spellings, two closely related languages which contain similar words, and Named Entities - while recognizing a language. They showed that by incorporating contextual information, many of these errors can be reduced considerably. They have used language models, dictionaries, logistic regression classification and Conditional Random Fields in their work. Gella et al. (2013) used language models and dictionaries in their work. In a more recent work (Gella et al., 2014), they evaluated some of the state-of-the- art LI techniques like *linguini*, *langid.py*, *polyglot* and *CLD2*. The existing systems have proven to be inadequate in dealing with code-mixed text. The performance of these systems deteriorates when the languages mixed in the text are not known a priori. Barman et al. (2014) used code-mixed data from Facebook posts and comments. The languages involved were Bengali, English and Hindi. They used simple unsupervised dictionary-based approach, supervised word-level classification with and without contextual clues, and sequence labelling using Conditional Random Fields.

2.3 Task Definition

Let us consider a query q denoted as < $w_1w_2w_3$ ... w_n > and is written in Roman script. The words, w_1,w_2,w_3, ... w_n, could be standard English words or transliterated form of Indian languages (L). The languages (L) can be Bengali (Bn), English (En), Gujarati (Gu), Hindi (Hi), Kannada (Ka), Malayalam (Ml), Marathi (Mr), Tamil (Ta) or Telugu (Te). The objective of the task is to identify the words as English or member of L depending on whether it is a standard English word or a transliterated L-language word. In general, the words of a single query usually come from 1 or 2 languages and rarely from 3 languages. It was also observed from the dataset that in case of mixed language queries, one of the languages is either English or Hindi. Thus, queries are formed by mixing Tamil and English words, or Bengali and Hindi words, but not for example, Gujarati and Kannada words. We were also required to identify the Named Entities as NE (e.g., Sachin Tendulkar, Kolkata, etc).

2.4 Dataset and Lexical Resources

This section describes the dataset that have been used in this work. The training and test data have been constructed using manual and automated techniques and made available to the task participants by the organizers of FIRE 2015 Shared Task. The training dataset consists of 2908 sentences whereas the test set contains 792 sentences.

The following resources provided by the organizers were also employed:

- **English Word Frequency List**[18]: It is in plain tab-separated text file that contains English words collected from a standard dictionary and followed by their frequencies computed from a large corpus. It contains noisy instances (very low frequency entries) as it is constructed from news corpora.

- **Hindi Word Transliteration Pairs 1** (Gupta et al., 2012): It is in plain tab-separated text file containing a total of 30,823 transliterated Hindi words (in Roman script) followed by the same word in Devanagari. It also contains Roman spelling variations for the same Hindi words (the transliteration pairs found using alignment of lyrics of Bollywood songs). However, it does not contain the frequency or occurrence of a particular word transliteration pair.

[18] http://cse.iitkgp.ac.in/resgrp/cnerg/qa/fire13translit/index.html

- **Bangla Word Frequency List**[19]: It is also in plain tab-separated text format. It contains Bengali words (Roman script, ITRANS format) followed by their frequency computed from a large Anandabazar Patrika news corpus. ITRANS to UTF-8 converter is used for obtaining the words in Bengali script.
- **Gujarati Word Transliteration Pairs**[2]: It contains transliterated Gujarati words (Roman script) followed by the same word in Gujarati script. Due to the poor availability of Gujarati resources, a small list of 546 entries was created from the training data of FIRE 2015 shared task[20].
- **Google Input Tools**[21]: We used the lookup table of transliterated word pairs provided in Google Input Tools. Such tables contain the transliterated pairs of native Indian languages in Roman Script. We used these tables for all of the 8 Indian languages to create word list for each language.
- **Corncob Web Dictionary**[22]: The dictionary contains 58110 distinct English words. We have used this dictionary to identify the English words.
- **Stanford NE Tagger**[23]: Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names, etc.

We also developed 11 lists of our own which are as follows:

- *Named Entity List*: We developed this named entity list using the training data. It contains 648 distinct names.
- *Emoticon List*: We developed this list using Wikipedia. This list contains 273 distinct emoticons.
- *Language Wordlist*: We developed nine wordlists for nine different languages using the training data. The wordlists contained few overlapping words.

[19] http://cse.iitkgp.ac.in/resgrp/cnerg/qa/fire13translit/index.html
[20] http://fire.irsi.res.in/fire/2015/working_notes
[21] https://www.google.com/inputtools/
[22] http://www.mieliestronk.com/wordlist.html
[23] http://nlp.stanford.edu/software/CRF-NER.shtml

2.5 System Description

Our primary task in the present context was word-level language identification. However, identification of Named Entities was also necessary in order to achieve our goals. As we elaborated the classification or typification of Named Entities (NEs) in the Chapter 4, this Chapter provides a brief glimpse of named entity identification as part of language identification challenge.

We use Conditional Random Field for sequence labeling each word of the sentence with appropriate language tags. We have trained our system using multiple features which are described in the next section.

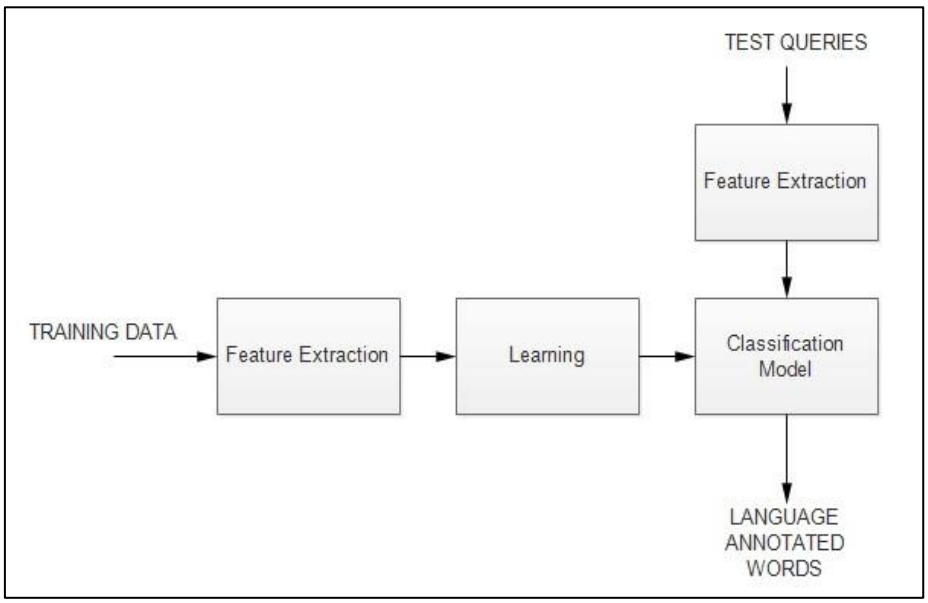

Fig 2.1. Overview of the LI System Architecture.

2.6 Features for Word-Level Language Identification

The following features were used for language identification:

- ***Capitalization***

  Three types of boolean capitalization features have been used for encoding capitalization information. As all the words are in Roman script we use the ASCII value to identify a capital character. The first feature is whether the first character of the word is capital or not. This is an important feature as this is later used for identification of Named Entity.

The second feature is whether the whole word is capital or not. The third feature is if any character in the word is capital or not.

For example, words like *Mumbai*, *BCSE, 3G,* etc.

CAP1: Is the first letter capitalized? If yes, then CAP1 = 1, else 0 (e.g., Mumbai)

CAP2: Is any character capitalized? If yes, then CAP2=1, else 0 (e.g., 3G)

CAP3: Are all characters capitalized? If yes, then CAP3=1, else 0 (e.g., BCSE)

- ***Word-Level Context***

The previous three words and the next three words along with the current token as well as the length of the current token are used as contextual features. As both language identification and points of code-switch are context sensitive (Chittaranjan et al., 2014; Muysken et al., 2001; Poplack et al., 1980), we have used this feature only for classification purpose. This feature is very much crucial to resolve the ambiguity in the word-level language identification problem. Let us consider examples given below:

Sentence 1: Mama *take* this badge off of me.

Sentence 2: Ami *TaKe* boli je ami bansdronir kichu agei thaki.

The word `take' exists in the English vocabulary. However, the backward transliteration of `TaKe' is also a valid Bengali word. Words like 'take', 'are', 'pore', and 'bad' are truly ambiguous words with respect to the word-level language identification problem as they are valid English words as well as their backward transliterations are valid Bengali words. In this regard, context of the word can be used to correctly identify the language for such an ambiguous word. Thus, the dynamic unigram feature used in the CRF++ template file analyses the previous token and the next token for their languages and the language of the current token is annotated according to that context. Therefore, we have considered it as a very useful feature.

CON1: Current token

CON2: Previous 3 and next 3 tokens

CON3: Length of the current token. This feature is important because words in Indian languages tend to be longer than the words in English.

- ***Special Character***

A word might start with some symbol, e.g. #, @, etc. These boolean features indicate the presence of hashtag (#), at the rate (@), hyperlink and emoticons. A list containing 273 distinct emoticons using different kind of special characters was made and used for the identification of emoticons.

For example, @aapyogendra, #aapsweep, http://t.co/pym4cr6xx0,:/

CHR1: If the word starts with #? If yes, then 1 else 0

CHR2: If the word starts with @? If yes, then 1 else 0

CHR3: If the word starts with http? If yes, then 1 else 0

CHR4: If emoticon? If yes, then 1 else 0

- ***Dictionary Feature***

  A total of 9 different languages were there to be identified. Therefore, we used 9 different lexical resources, one for each of the languages. We also used 9 different boolean features to represent if a particular token is present in a particular lexicon. If a particular word is present in more than one lexicon, we use a unigram relational feature in the template file of CRF++ to handle the ambiguity. This unigram relational feature is determined using two or more other features.

  For example, U1: %x[0,20]/%x[0,21]

  LEX1: Is present in English dictionary? If yes, then 1, else 0

  LEX2, LEX3,,…, LEX9 for other languages.

- ***Presence Of Symbol In Word***

  Only one boolean feature is used to identify the words with punctuation marks present in it. The punctuation marks can be an apostrophe ('), a dash (-), etc.

  For example, goalkeepers\, angul-er

  CHR5: Is symbol present? If yes, then 1 else 0

- ***Presence Of Digit***

  This boolean function is used to indicate if a word contains a digit. As the corpus provided contains social media text, this feature was used. In phonetic script people often use digit to shorten or abbreviate their text.

  For example 'gr8' in place of 'great', '4nds' for 'friends'

  CHR6: Is digit present? If yes, then 1 else 0

- ***Number Identification***

  This boolean feature is used to identify if the token is a number or not. For example, numbers like 30, 67, etc.

  CHR7: Is token a number? If yes, then 1 else 0

- ***Named Entity Identification***

  For NE identification we use the Stanford NE Tagger[24] along with a lexicon of named entities. We use two boolean features for this purpose. The first is based on the basic lexicon search (in List 1) and the second is based on the output of the Stanford NE Tagger (in List 2). We use another unigram relational feature in CRF++ for the classification of NE Tags. The basic lexicon is the Named Entity list which we developed for our task.

  NE1: If name entity matches List1, then NE1 = 1, else 0

  NE2: If name entity matches List2, then NE2 = 1, else 0

2.7 Results

In this work, Conditional Random Field (CRF) (Kudo, 2014) has been used to build the classification framework for the word-level language identification. We have used CRF++ toolkit[25] which is a simple, customizable, and open source implementation of CRF.

The accuracies with respect to nine different languages as well as the average and weighted F-measures are shown in Table 2.1 and Table 2.2 respectively. In case of English and Bengali, the precision values are better compared to other languages while the other languages like Hindi, Marathi, Kannada, Tamil, Telegu achieves better recall.

Table 2.1. Token Level Results for Language Identification.

Language	*Precision*	*Recall*	F-Measure
X	0.9423	0.7525	0.8367
Bengali	0.8129	0.6937	0.7486
English	0.9318	0.8555	0.892
Gujarati	0.0757	0.4118	0.1279
Hindi	0.7772	0.8182	0.7972
Kannada	0.2793	0.799	0.4139
Malayalam	0.2597	0.6522	0.3715
Marathi	0.4956	0.8687	0.6311
Tamil	0.5672	0.817	0.6696
Telegu	0.3874	0.8153	0.5252

[24] http://nlp.stanford.edu/software/CRF-NER.shtml
[25] http://crfpp.googlecode.com/svn/trunk/doc/index.html

Table 2.2. Other Performance Metrics.

Tokens Accuracy (in %)	75.4896
Utterances Accuracy (in %)	21.5909
Average F-Measure	0.53839
Weighted F-Measure	0.74983

2.8 Error Analysis

If we look at the confusion matrix (in Table 2.3) for different languages, we can notice that many other languages have been wrongly classified as English. This is primarily due to the overlapping words between English and all other Indian languages. In our task, the accuracies of MIXes and NEs were quite low. One of the primary reasons for the increased error rate in MIX determination was the absence of post processing attempts to identify the mixed words. Also, the sub-classification errors in NE recognition could have been significantly reduced by adding a NE-classification module to our system. Our accuracy also declined for Gujarati, Kannada and Malayalam. Use of larger wordlists and transliterated dictionary should have improved the scores. It is observed from the confusion matrix that Kannada and Telegu are often confused with Bengali.

Table 2.3. Confusion Matrix for Analyzing Language Identification Systems.

	en	***X***	***hi***	***bn***	***Ml***	***mr***	***kn***	***te***	***gu***	**Ta**
en	3772	79	37	47	1	2	1	16	1	6
X	32	1763	2	1	0	0	0	1	0	0
hi	141	84	1242	38	0	6	3	6	9	0
bn	84	71	50	1112	0	7	2	4	9	8
ml	19	38	2	13	60	1	12	0	0	13
mr	23	33	53	65	2	225	3	2	1	1
kn	59	93	8	109	2	2	167	10	0	19
te	54	50	22	102	5	9	5	203	0	6
gu	18	13	77	39	0	3	6	0	14	9
ta	33	74	3	4	20	0	5	0	0	308

2.9 Conclusion

In this Chapter, we presented a brief overview of our system to address the automatic identification of word-level language. While the CRF-based approach was satisfactory, the results could have been improved by including post-processing heuristics for identifying mixed words and named entities. Using more character level features should improve the accuracy of the system. Also some basic knowledge about other languages and better wordlists and dictionary for regional languages should improve the accuracy of the present system. It has to be

mentioned that we used character n-grams (n=1 to 5) as one of the features of CRF++. However, the performance of the system declined on incorporating it. It states that the noisy nature along with language mixes in such social media texts is hard to be captured by character n-grams. However, we will use word n-grams in our future attempts to observe its impact on LI. Our system demonstrated an overall accuracy of 75.5% for token level language identification. The strict F-measure scores for the identification of token level language labels for Bengali, English and Hindi are 0.7486, 0.892 and 0.7972 respectively. The overall weighted F-measure of our system was 0.7498.

CHAPTER

3 PART-OF-SPEECH TAGGING OF CODE-MIXED SOCIAL MEDIA TEXT

Chapter 3

Part-of-speech Tagging of Code-Mixed Social Media Text

While Language Identification is the first step to analyzing any code-mixed text, part-of-speech (POS) tagging is the very next step towards a comprehensive analysis. In this Chapter, we describe how we can determine the part-of-speech tags for every word in the document. Recently a task[26] - 'POS Tagging for Code-mixed Indian Social Media Text' – was organized by ICON 2015[27]. We have used the dataset – both training and test – provided by the organizers for our analysis and evaluation.

3.1 Introduction

Part-of-Speech (POS) tagging – a syntactic analysis usually done after language identification - is one of the key tasks in any language processing applications. It is the process of assigning the appropriate part of speech or lexical category to each word in a sentence. Apart from assigning grammatical categories to words in a text, POS tagging also helps in automatic analysis of any text.

To develop an accurate tagger, it is essential to develop various rules based on the language. We also need large annotated corpus which could be used for discovering the rules and training the model. Accurate annotation of a corpus requires the expertise of language experts – which is expensive and time consuming. Also it is not portable from one language to another. Use of unsupervised and semi-supervised machine learning approaches solves all the problems mentioned above.

The increasing popularity of social media platforms – blogs, micro-posts (e.g. Twitter[28]) and chats (Facebook[29]) - has ensured availability of large amount of code-mixed data. But, texts

[26] http://ltrc.iiit.ac.in/icon2015/contests.php
[27] ltrc.iiit.ac.in/icon2015/
[28] twitter.com
[29] www.facebook.com

obtained from various online platforms differ from traditional writings. These texts are predominantly unstructured. Also, many variations can be observed in terms of writing style and vocabulary. Such texts are mostly informal and have multiple languages in a single sentence, or even in a single word. This code-mixed nature of text, coupled with the fact that they are written using Roman script (instead of native script), makes it extremely challenging for linguistics and data analysts to process such data. This has given a new dimension to the traditional problems of language identification and POS tagging.

In this Chapter, we address the problem of part-of-speech Tagging in mixed social media data. India is a land of many languages with Hindi and English recognized as the more popular ones. From the Indian perspective, it is generally observed that one of the languages used in social media conversations are either English or Hindi. In this work, all the three mixed scripts contain English as one of the languages. The Indian languages present are Bengali, Hindi and Tamil.

To tag the words with their corresponding part-of-speech tags, we have used Stanford part-of-speech tagger as our baseline and developed the final system using Conditional Random Field (CRF). We have obtained results for three language pairs, namely Hindi-English (Hi-En), Bengali-English (Bn-En) and Tamil-English (Ta-En). For each pair, the task required us to develop a constrained and an unconstrained system. The constrained system uses only the datasets for analysis while the unconstrained system uses additional language resources. In this Chapter, we concentrate on building our POS tagger system with minimal external resources. While the Stanford POS Tagger uses no additional resource, the CRF model uses only a list of smileys.

The rest of the Chapter is organized as follows. We present an account of the previous works done in the part-of-speech tagging in Section 3.2. In Section 3.3, we discuss datasets followed by the system description in Section 3.4. The results and observations have been presented in Section 3.5 and the conclusion in Section 3.6.

3.2 Related Work

Part-of-Speech tagging has been a centre of many researches for the past few decades. Since it started in the middle sixties and early seventies (Harris, 1962; Klein and Simmons, 1963; Greene and Rubin, 1971), a lot of new concepts have been introduced to improve the efficiency of the tagger and to construct the POS taggers for several languages.

Rule based POS tagger was introduced in the nineties (Karlsson et al., 1995) and gave better accuracy than its predecessors. One of the most successful rule based English tagger (Samuelsson and Voutilainen, 1997) had a recall of 99.5% with a precision of around 97%. The rule based taggers consists of complex but accurate constraints which makes them very efficient for disambiguation. Statistical model based tagger (DeRose, 1988; Cutting et al., 1992; Dermatas and Kokkinakis, 1995; Mcteer et al., 1991; Merialdo, 1994) are widely used because of the simplicity and the independence of the language models. Most commonly used statistical models are bi-gram, tri-gram and Hidden Markov Model (HMM). The only problem with statistical models is that these kinds of taggers require a large annotated corpus. Machine learning algorithms are statistical in nature but the models are more complicated than simple n-gram. Models for acquiring disambiguation rules and transformation rules from the dataset were constructed in late 80's and early 90's (Hindle, 1989; Brill, 1992; Brill, 1995a; Brill 1995b). Neural network has also been used for POS tagging (Nakamura et al., 1990; Schutze, 1993; Eineborg and Gamback, 1993; and Ma and Isahar, 1998). POS taggers were also developed using Support Vector Machine (SVM) (Nakagawa et al., 2001). These taggers were simple and efficient than the previous taggers. The successor of this tagger was developed by Gimenez and Marquez (2003) and the approach they used for POS tagging was considerably faster than its predecessor. The latest development was the use of Conditional Random Field (CRF) for POS tagging (Sha and Pereira, 2003; Lafferty, 2001; Shrivastav et al., 2006). These taggers are better for disambiguation as they find global maximum likelihood estimation.

Recently, a large number of researchers are trying to expand the scope of automatic POS taggers so that they can work on complex non European languages. India is a country with rich linguistics so POS taggers for Indian languages are one of the most explored topics. The first effort was to develop a Hindi POS tagger dated back in the nineties (Bharati et. al., 1995). This tagger was based on a morphological analyzer. The analyzer would provide the root word with its morphological features and generalized POS category. Singh et al. (2006) slightly modified this approach by using a decision tree based classifier and achieved an accuracy of 93.45%. Instead of using a full morphological analyzer Shrivastava and Bhattacharya (2008) used a stemmer to generate suffixes which was in turn used to generate POS tags. Conditional Random Field was also used along with morphological analyzer in a couple of works (Agarwal et. al., 2006; Avinesh et. al., 2006).

One of the earliest works on Bengali POS tagger was conducted by Seddiqui et al. (2003) and Chowdhury et al. (2004). Chowdhury et al. (2004) implemented a rule based tagger which hand

written rules formulated by expert linguists. In more recent work, Hasan et al. (2006) developed a supervised POS tagger. This method was less effective due to lack of tagged training corpus. In later years, we have seen many works on Bengali POS tagger. One of the most successful taggers was developed by using HMM and Maximum Entropy models (Dandapat et. al., 2006; Dandapat et al., 2007). They also used a morphological analyzer to compensate for the lack of annotated training corpus. These two models were used to implement a supervised tagger and a semi-supervised tagger. The accuracy achieved was around 88% for both the models. Ekbal and Bandopadhyay (2007) carried out further research on the tagger. They annotated a news corpus and created two taggers - one SVM based tagger and another CRF based tagger - which reported an accuracy of 86.84% and 90.3% respectively.

In Tamil, Selvam and Natarajan (2009) proposed a rule based morphological analyzer to annotate the corpora and used it to train the POS tagger. They used the Tamil version of Bible for the tagged corpus and achieved an accuracy of 85.56%. Dhanalakhsmi et al. (2009) developed a SVM based tagger using linear programming and a new tagset for Tamil with 32 tags. They used this tagset for building a training corpus and reported an accuracy of 95.63%. Another SVM based POS tagger (Dhanalakhsmi et. al., 2009) was proposed by them in a different work. They extracted linguistic information using machine learning techniques which was then used to train the tagger. This tagger achieved an accuracy of 95.64%.

Even after decades of research on monolingual POS taggers, there are just a few taggers with accuracy over 90%. A new challenge has developed over the past few years in the form of code mixed social media text. This field of research is at a nascent stage. The basic challenges and complicacies of social media text are spelling variations and word sense disambiguation (See Section 1.4 for more details). As traditional POS taggers were not efficient for social media text, new taggers targeting social media text were constructed. However, these taggers are mostly monolingual and not suitable for code-mixed text. The first was developed by Gimpel et al. (2011) for tagging English tweets. They developed a new POS tagset and tagged 1827 tweets for training corpus for a CRF tagger with arbitrary local features in log-linear model adaptation. Owoputi et al. (2013) improved the original Twitter POS tagger as they introduced lexical and unsupervised word clustering features. This increased the accuracy from 90% to 93%.

One of the first POS taggers for code-mixed text was developed by Solorio and Liu (2008). They constructed a POS tagger of English-Spanish text by using existing monolingual POS taggers for both the languages. They combined the POS tag information using heuristic procedures and achieved the maximum accuracy of 93.4%. However, this work was not on social media text and

hence the difficulties were considerably less. Gella et al. (2013) developed a system to identify word level language and then chunk the individual languages and produce POS tags or every individual chunk. They used a CRF based Hindi POS tagger for Hindi and Twitter POS tagger for English and achieved maximum accuracy of 79%. Vyas et al. (2014) developed a English-Hindi POS tagger for code mixed social media text.

3.3 Dataset

The ICON-2015 NLP tools contest[30], POS tagging for Code mixing text is designed for evaluating team's ability to identify the POS tags for code three (Hindi, Bengali and Telugu) mixed Indian languages. Organizers released the code mixed train and test set for each languages.

Table 3.1. Summary of Dataset (Utterances).

EN-HI			***EN-BN***			***EN-TA***		
	Training	***Test***		***Training***	***Test***		***Training***	***Test***
EN	6178	8553	***EN***	9973	5459	***EN***	1969	819
HI	5546	411	***BN***	8330	4671	***TA***	1716	1155
O	4231	2248	***O***	6335	3431	***O***	630	281
Total	15955	11212	***Total***	24638	13561	***Total***	4315	2255

Table 3.2. Summary of Dataset (Sentences).

Language	***Total Of Sentences***	
	Training data	***Test data***
Bengali-English	2837	1459
Hindi-English	729	377
Tamil-English	639	279

Table 3.3. Statistics of POS Tags Present In Training Data.

EN-HI		***EN-BN***		***EN-TA***	
EN	6178	***EN***	9973	***EN***	1969
HI	5546	***BN***	8330	***TA***	1716
O	4231	***O***	6335	***O***	630
Total	15955	***Total***	24638	***Total***	4315

3.4 System Description

We have followed a supervised approach in this work. We have extracted various features that are pertinent to this task. The various steps involved in POS tagging are listed as follows:

[30] http://ltrc.iiit.ac.in/icon2015/contests.php

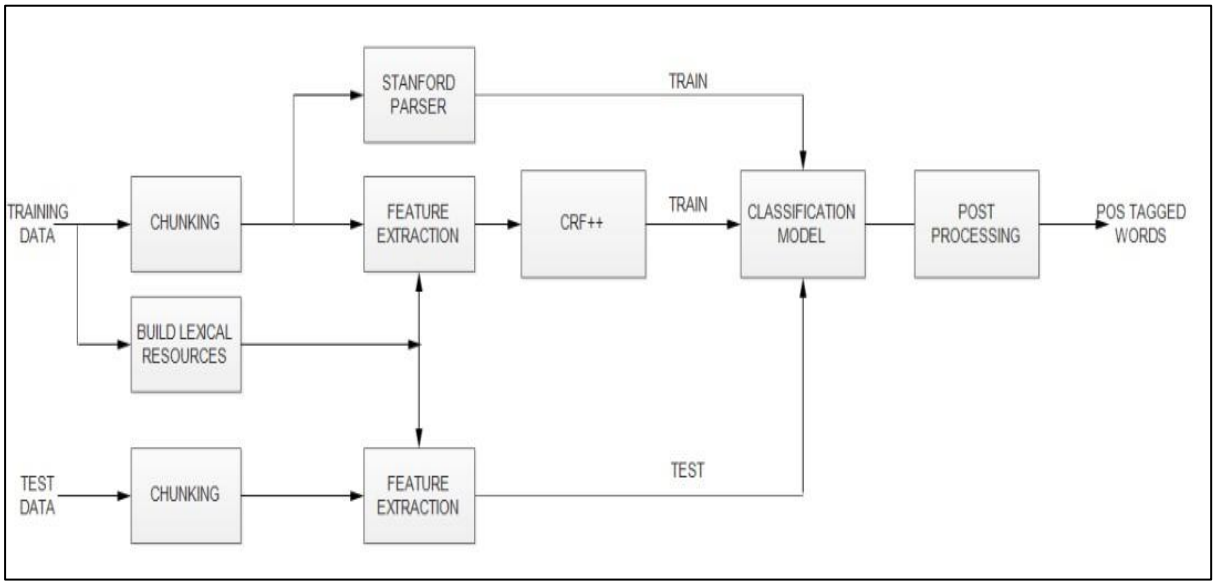

Fig 3.1. Overview of the POS System Architecture.

3.4.1 Chunking

Each of the three given texts contains English as one of the dominant languages. The other dominant language is Bengali, Hindi and Tamil in each of the three texts. For each input file, we have performed chunking on the raw text to segment the words belonging to different languages. The various language tags used in the training and test data are en (English), hi (Hindi), bn (Bengali), ta (Tamil), ne (Named entities), acro (Acronyms), univ (Universal) and undef (Undefined). For each of the language tags, we have created a wordlist belonging to that particular language.

3.4.2 Lexicons for Dominant Languages

English, Bengali, Hindi and Tamil were identified as the dominant languages. For each of these four languages, we have created a list of words which belong to any particular POS tag (from the training files). These lists are essential for extracting feature for training our CRF model.

3.4.3 POS Tagging

We have used two different approaches for POS Tagging of the test data.

- **POS Tagging Using Stanford POS Tagger**

  For our baseline, we trained our system using Stanford POS Tagger (Toutanova et al., 2003). Using the training data, we trained the Stanford POS Tagger initially. The architecture (arch property of the tagger) that we used for training was: words(-1,1), unicodeshapes(-1,1), order(2), suffix(4). Individual models were generated for English, Bengali, Hindi and Tamil. The test data was tagged using these generated models.

- **POS Tagging Using CRF++**

  In this work, Conditional Random Field (CRF) (Kudo et al., 2014) has been used to build the framework for word-level language identification classifier. We have used CRF++ toolkit[31] which is a simple, customizable, and open source implementation of CRF.

The following features were used to train the CRF model:

- **Length Of The Current Word**

  The length of the current word has been used as one of the features. It is often noted that words belonging to a specific language and part-of-speech are often longer than others. We have used this feature to exploit word length in determining the part-of-speech of the word.

- **Current Word**

  For example, if the sentence is *I have been told of the place,* then each word is analyzed at a time. If the word currently being examined for part-of-speech tagging is been, then the word '*been'* is considered as one of the features.

- **Previous Two Words**

  For example, if the sentence is *I have been told of the place* and current word is '*been',* then the previous two words are *I* and *have*.

- **Next Two Words**

  Using the previous example, if the sentence is *I have been told of the place* and the current word is *been*, then the next two words are *told* and *of*.

- **Suffix**

  This feature considers of the suffix of every word. If length of a word is more than 3 then suffix of length 3 and 2 are taken.

  e.g.: *een* and *en* are the suffixes for *been*.

[31] https://taku910.github.io/crfpp/#download

- **Prefix**

  This feature considers of the prefix of every word. If length of a word more than 3 then prefix of length 3 and 2 are taken.

  e.g.: *bee* and *be* are the suffixes for *been.*

- **If Word Contains Any Symbol**

  This feature is boolean in nature and represents if the current word contains any symbol. Presence of symbol in a word gives a possible hint about the part-of-speech of the word.

- **If Word Contains Any Digit**

  Similar to the previous feature, this boolean feature represents if the current word contains any digit. Presence of digit in a word gives a possible hint about the part-of-speech of the word. e.g.: *kheye6ilam*, *ki6u*, *ka6e, 6ghanta*

- **Is Noun**

  This feature represents if the current word is a noun. During the training phase, we build up a list of nouns for every language. This list is used during test phase to evaluate this feature.

  e.g.: *match, love, khushi, kaam, meye*

- **Is Adjective**

  This feature represents if the current word is an adjective. During the training phase, we build up a list of adjectivess for every language. This list is used during test phase to evaluate this feature.

  e.g.: *ekta, beshi, good, nice*

- **Is Verb**

  This feature represents if the current word is a verb. During the training phase, we build up a list of verbs for every language. This list is used during test phase to evaluate this feature.

  e.g.: *hoy, lage, be, will*

- **Is Pronoun**

  This feature represents if the current word is a pronoun. During the training phase, we build up a list of pronouns for every language. This list is used during test phase to evaluate this feature.

  e.g.: *tomar, tumi, you, I*

- **Is Conjunction**

  This feature represents if the current word is a conjunction. During the training phase, we build up a list of conjunctions for every language. This list is used during test phase to evaluate this feature.

  e.g.: *kintu, and, to, but*

- **Is Adverb**

  This feature represents if the current word is an adverb. During the training phase, we build up a list of adverbs for every language. This list is used during test phase to evaluate this feature.

  e.g.: *ekhon, takhon, just, very*

- **Is Determiner**

  This feature represents if the current word is a determiner. During the training phase, we build up a list of determiners for every language. This list is used during test phase to evaluate this feature.

  e.g.: *the, this, a*

- **Is Dollar**

  This feature represents if the word represent any numerical measure.

  e.g.: *1st, 26th, one, two*

- **Is Q**

  This feature represents if the word represent any quantitative measure.

  e.g.: *enuf, more, many, khub*

- **Is U**

  This feature represents if the word is website link

  e.g.: *pdf2fb.net*

- **Is X**

  This feature represents if the word is a non-classified token or if it has no meaning.

  e.g.: *geetamroadpi*

During the training phase, we train the CRF model using all the above features. Four language models are built, corresponding to the four dominant languages – English, Bengali, Hindi and Tamil. In the test phase, we use the generated models to tag the words with their appropriate part-of-speech tags.

3.4.4 Post-processing

All the words belonging to the four dominant languages were tagged by the CRF model. The acronyms, named entities and the universal words were tagged by consulting the lists built during training. All the words which could not be tagged by our model were subjected to a post-processing module. For every language tag (acro, univ, ne), we found out the most frequent part-of-speech tag. Also, we used some logical reasoning to tag the words which were not tagged by our tagger models. For example, any untagged word which contains www, http or .com in it is allotted the U tag. Similarly, we use a smiley list to tag the smileys as E. Punctuations and hashtags were tagged likewise.

3.5 Results and Observations

We evaluated the POS-tagging done by our baseline model (Stanford Parser) and the CRF model. The results are presented in Table 3.4.

Table 3.4: Accuracy of the Model.

Language Pair	*Accuracy in %*	
	Baseline(Stanford Model)	***CRF Model***
Bengali-English	60.05	75.22
Hindi-English	50.87	73.2
Tamil-English	61.02	64.83

The results of Tamil-English are less than that of Bengali-English and Hindi-English. The primary reason for lower accuracy is the variation in tag used in gold standard files of Tamil-English.

3.6 Conclusion

In this task, we have addressed the POS tagging of mixed script social media text. The texts contained two or three languages, with English being one of the three languages. The other languages were Hindi, Bengali and Tamil. We have trained Stanford POS Tagger to build a baseline model. Our final model used Conditional Random Field for part-of-speech tagging. Our results are encouraging and the performance deterioration of Tamil-English mixed text can be attributed to the mismatch of POS-tags.

CHAPTER

4 NAMED ENTITY RECOGNITION AND LINKING FOR SOCIAL MEDIA TEXT

Chapter 4

Named Entity Recognition and Linking for Social Media Text

In this Chapter, we describe our approach for Named Entity rEcognition and Linking Challenge (NEEL)[32] at the #Microposts2016[33]. The task is to automatically recognize entities and their types from English microposts, and link them to corresponding DBpedia[34] 2015 entries. If the resources do not exist, we use NIL identifiers instead. The task is unique as Twitter[35] data is informal in nature with non-conformational spellings, random contractions and various other noises. For this task, we developed our system using a hybrid model. We have used various existing named entity recognition (NER) systems and combined them with our classifier to improve the results. It should be pointed out that Named Entities are always considered to be universal. They are not allotted any language tags and are language-independent. We have developed our system keeping this in mind. Our Named Entity Recognition and Classification system can be extended to any language which is used in social media text.

4.1 Introduction

In present day world, the relevance and importance of various social media platforms are immeasurable. Microposts such as tweets are limited in number of characters. However, the conciseness of the text is barely a pointer to its usefulness. From opinion mining during political campaigns to live feeds during sports events, from product reviews to vacation posts, Twitter is almost ubiquitous. Twitter promotes instant communication. Most celebrities use it to form their own digital presence. It also serves as a common forum where people have the capability to rise from obscurity to prominence through sharing of opinions. If we compare microposts to any standard long document such as blog or news articles, there exist a number of differences. Long articles are usually well written. They follow a definite structure, include headings and follow the

[32] http://microposts2016.seas.upenn.edu/challenge.html#
[33] http://microposts2016.seas.upenn.edu/challenge.html
[34] http://wiki.dbpedia.org/
[35] twitter.com

rules of English grammar. Microposts, on the other hand, are short, noisy and hardly show any adherence to formal grammar. Presence of extraneous characters like hashtags, abbreviations and the lack of structure and context makes it difficult to extract relevant information. Due to this complexity, existing named entity recognition systems (NER) do not perform very well with tweet data.

In NEEL challenge (Rizzo and Erp, 2016) of #Microposts2016 (Cano et al., 2016), we were required to automatically identify the named entities and their types from Twitter data and link them to the corresponding URIs of the DBpedia 2015-04 dataset[36]. Identifying the named entities and linking them to an existing knowledge base enriches the text with more contextual and semantic information. The mentions which could not be linked to any existent DBpedia resource page were recognized as NIL mentions. These mentions were clustered to ensure that the same entity, which does not have a corresponding entry in DBpedia, will be referenced with the same NIL identifier. We have developed three systems for the NEEL challenge, the major difference between the systems being the features used for each run. Our system follows a hybrid approach where Stanford Named Entity Recognition System is used to identify the entity mentions. In the next step, we run ARK Twitter Part-of-Speech Tagger[37] to identify the mentions which are missed formerly. We use our own classifier to detect the type of the mentions. The named entity linking to DBpedia resources is done using Babelfy[38]. It must be noted that we followed a feature-based approach for the NEEL challenge. We also combined the existing tools for Named Entity Recognition and Linking. Each of the existing tools, like the Stanford NER, ARK Part-of-Speech Tagger and Babelfy are state-of-the-art. We explored their strengths and weaknesses in our work.

In Section 4.2, we discuss the related work. In Section 4.3, we present a description about the dataset. Section 4.4 explains the construction of our system. It details the various modules which have been used for named entity recognition and linking. The results and observations are presented in Section 4.5. Section 4.6 is devoted to error analysis while Section 4.7 concludes the Chapter.

4.2 Related Work

Named entity recognition (NER) of longer texts, such as news, is a very well studied problem (Nadeau and Sekine, 2007; Roberts et al., 2008; Marrero et al., 2009). Facebook posts and

[36] http://wiki.dbpedia.org/dbpedia-data-set-2015-04
[37] http://www.cs.cmu.edu/~ark/TweetNLP/
[38] http://babelfy.org/

Twitter tweets present a new and challenging style of text for language technology due to their noisy and informal nature. The performance of "off the shelf" NLP tools, to recognize named entities, is weak on tweet corpora. Supervised named entity recognition was performed in English by Chinchor (1998) and on other languages by Sang and Meulder (2003). Black et al. (1998) have used clustering to group together different nominals referring to the same entity. Semi-supervised and unsupervised approaches have also shown promising results. Collins and Singer (1999) used unsupervised models for NER. In their work, they use spelling rules and contextual information for classification of named entities. Elsner et al. (2009) described a generative model for clustering named entities. The model is unsupervised and it uses features from the named entity itself and its syntactic context. Named entity recognition for Twitter was also experimented using CRFs (Ritter et al., 2011). Bontcheva et al. (2013) developed TwitIE, a Twitter-adapted version of the state-of-the-art Stanford NER.

NER tasks invariably leads to named entity disambiguation and entity linking tasks. For example, given a text like "Harry was exceptional in the match", we have to tell if the "Harry" refers to Harry Kane, an English footballer and not Prince Harry, the English prince or Harry Potter, the fictional character. Establishing these mappings between the mentions and the actual entities is the problem of named-entity disambiguation (NED). Entity Linking also helps in NED. Current approaches of named entity linking establish links not just to entity types, but to the actual entities themselves (Liu et al., 2011). Linking named entities using Wikipedia have met with considerable success in recent years (Milne et al., 2008; He et al., 2011; Meij et al., 2011; Erbs et al., 2011; Cornolti et al., 2013).

Various knowledge resources built from Wikipedia can be used for linking tasks. Navigli and Ponzetto (2012a) built BabelNet, Auer et al. (2007) built DBPedia and Hoffart et al. (2013) created YAGO2. Rao et al. (2013) proposed a distantly supervised entity linking model, to automatically link disambiguated entities (in Freebase) to the corresponding descriptive Wiki pages. Mihalcea and Csomai (2007) disambiguated each word in a sentence independently by exploiting the context in which it occurs. Cucerzan (2007) used lexical context to disambiguate the mentions. Moro et al. (2014) showed that the semantic network structure can be leveraged to disambiguate both word senses and named entities at the same time. Ferragina and Scaiella (2010), Hoffart et al. (2012) and Bohm et al. (2012) have also worked on entity relatedness. While Ferragina and Scaiella (2010) annotated short texts using Wikipedia, Bohm et al. (2012) introduced the notion of k-similarity. Two resources are k-similar, if k of their property/value

combinations is exact matches. Hoffart et al. (2012) measured semantic relatedness between two entities by projecting them onto a high dimensional concept space derived from Wikipedia.

4.3 Dataset

We have used the dataset for Named Entity rEcognition and Linking Challenge (NEEL)[39] at the #Microposts2016. The dataset consists of tweets extracted from a collection of over 18 million tweets. It includes event-annotated tweets provided by the Redites[40] project covering multiple noteworthy events from 2011, 2013 (including the death of Amy Winehouse, the London Riots, the Oslo bombing and the Westgate Shopping Mall shootout), tweets extracted from the Twitter firehose from 2014 and 2015 via a selection of hashtags. Since the task of this challenge was to automatically recognise and link entities, the dataset contains both event and non-event tweets.

4.4 System Description

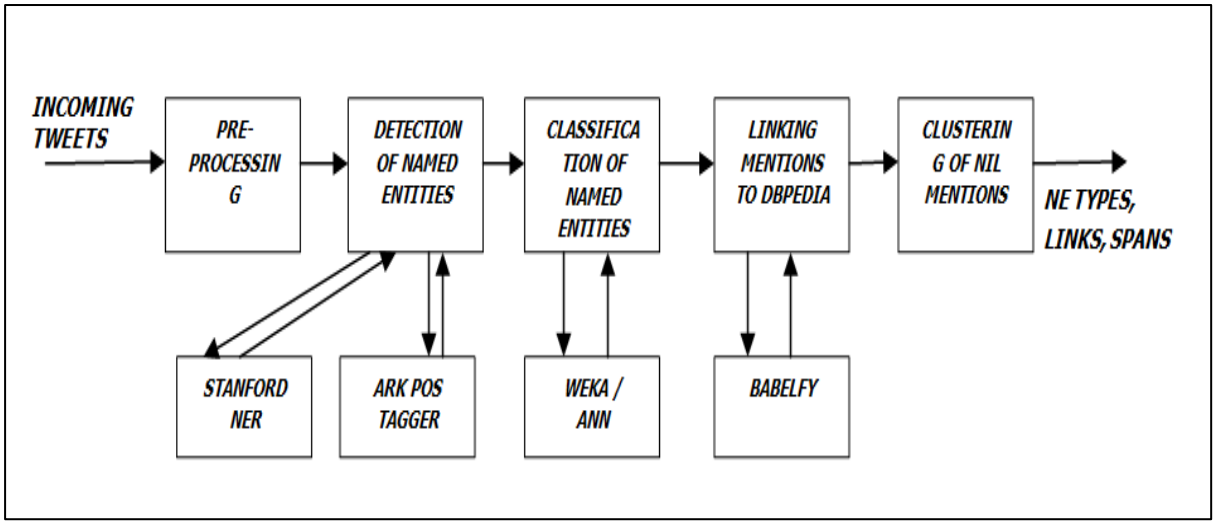

Figure 4.1: Overview of the System Architecture.

Our system follows four steps in pipeline as shown in Figure 1. Mention detection in two stages, followed by mention type classification, mention linking and NIL clustering.

[39] http://microposts2016.seas.upenn.edu/challenge.html
[40] http://demeter.inf.ed.ac.uk/redites/

4.4.1 Pre-processing

From the training data, the mentions referring to the 7 types of entities were extracted to form 7 bags of words. Using the initial words as seeds, the Wikipedia dumps were crawled to expand the set of words. These lists represent potential candidates for Named Entity mentions.

4.4.2 Detection of Entity Mentions

In this step, the named entity mentions in the given tweets are identified using two different approaches.

- ***Using Stanford Named Entity Recognizer***

  The Stanford Named Entity Recognizer[41] (Finkel et al., 2005) was used to extract the named entities. It is a CRF classifier implementing linear chain Conditional Random Field. We use the 3 class model to extract the named entities belonging to classes Location, Person and Organization. While the recall was very low, the precision of Stanford NER was quite good.

- ***Using ARK Twitter Part-Of-Speech Tagger***

  The tweets were tokenized and assigned Part of Speech tags using the ARK Twitter Part-of-Speech Tagger (Gimpel et al., 2011). We used the Twitter POS model with 25-tag tagset. The proper nouns (NNP and NNS tagged as ^) and possessive proper nouns (tagged Z) along with hashtags (#) and at-mentions () were extracted as probable candidates for Named Entity mentions. The mentions which were already identified using Stanford NER are not considered for classification step as they are already classified by the tagger itself. The rest of the mentions are classified using our classifier in the next step.

4.4.3 Classification of Entity Types

In the machine learning software WEKA[42] (Hall et al., 2009), we use the following features to form a feature set and used the Random Forest classifier to generate a pruned C4.5 Decision Tree for 7-way classification of the named entity mentions - Thing, Event, Character, Location, Organization, Person and Product, while providing the identified noun entities from previous steps as input. We checked the accuracy by using various classifiers like Naive Bayes, k-Nearest

[41] http://nlp.stanford.edu/software/CRF-NER.shtml
[42] http://www.cs.waikato.ac.nz/ml/weka/

Neighbour and Support Vector Machine on training data with a 10-fold cross validation. Random Forest gave the best results.

(A) Features for Run 1

The features used for Run 1 were as follows:

- Length of the mention string
  e.g.: If the mention is *"Harry Potter",* then the length of the mention string is 12.
- If the mention is all capitalized
  e.g.: *POTUS* meaning *President of the United States.*
- If the mention contains mixed case
  e.g.: *#ZodiacFacts*
- If the mention contains digits
  e.g.: *@106andpark* which is a twitter handle and denotes a person or entity.
- If internal period is present in mention string
  e.g.: *U.S.* for United States
- If present in list of Persons
  To be considered a Person, the mention should be the name of people. Titles and roles such as Dr. and President are not considered Persons.
  e.g.: *Barack Obama, John Hamm,* etc.
- If present in list of Things
  For a mention to be considered a *Thing,* it has to be a language, ethnic group, nationality, religion, disease, sport or astronomical object.
  e.g.: *#Sagittarius, American,* etc.
- If present in list of Events
  Holidays, sport events, political events and social events are considered as Events.
  e.g.: *London Riots, 2nd World War,* etc.
- If present in list of Characters
  Character can be fictional character, comics character or title character.
  e.g.: *Batman, Wolverine,* etc.
- If present in list of Locations
  For an entity to be considered as Location, it has to be a public place, region, commercial place or building.
  e.g.: *Miami, Yankee Stadium,* etc.

- If present in list of Organizations

  Organization can be company, brand, political party, government body, press name, public organization or collection of people like sports teams, musical bands, religious orders, etc.

  e.g.: *Apple, Police,* etc.
- If present in list of Products

  Products can be movies, tv series, music albums, devices, programming languages or operating systems.

  e.g.: *Mac Os X, Today Show,* etc.

The above-mentioned lists are basically the bag of words produced from the training data in the pre-processing step.

(B) Features for Run 2

We made use of various text based features and bag of words in Run 1. In Run 2, we explored various contextual features in addition to the features of Run 1. So we combined ten new features with the previous twelve features for Run 2. The ten additional features used in Run 2 were as follows:

- Context score for Person entity
- Context score for Location entity
- Context score for Character entity
- Context score for Organization entity
- Context score for Event entity
- Context score for Thing entity
- Context score for Product entity
- Frequency of Part-of-speech of mention
- Frequency of previous Part-of-speech
- Frequency of next Part-of-speech

Context score of a particular mention is calculated for a three word window of the mention. For each class, we have the number of occurrences of each word in a three word window. While calculating the feature value, we assign the sum of the frequency of the words forming that fixed-size window as the mentions context score.

(C) Run 3

We wanted to apply a Feed-Forward neural network (also called the back-propagation networks and multilayer perceptron) to our feature set and see how it performs as these kind of Artificial Neural Networks are useful in constructing a function where the complexity of the feature values makes the decision for building such a function by hand almost impossible. We took the same features of Run 2 and employed a feed-forward neural network based regression model with 5 hidden layers. For the previous two runs, i.e. Run1 and Run2, the tags from Stanford NER were considered as the primary influence over our classifier tags as its accuracy was quite good. For Run 3 however, we omit the Stanford NER influence and let only the neural network model do the tagging to check the efficiency of our classifier.

4.4.4 Linking Mentions to DBpedia

DBpedia[43] is a widely available Linked Data dataset and is composed by a series of RDF (Resource Description Framework) resources. Each resource is uniquely identified by a URI (Uniform Resource Identifier). A single RDF resource can be represented by a series of triples of the type <S,P,O> where S contains the identifier of the resource (to be linked with a mention), P contains the identifier for a property and O may contain a literal value or a reference to another resource.

In this challenge, a mention in a tweet was linked to the identifier of a resource (i.e. the S in a triple). Note that in DBpedia there are cases where one resource does not represent an entity, instead it represents an ambiguity case (disambiguation resource), a category, or it just redirects to another resource. In this challenge, only the final IRI properly describing a real world entity (i.e. containing their descriptive attributes as well as relations to other entities) are considered for linking. Thus, if there is a redirection chain given by the property wikiPageRedirects, the correct IRI is the one at the end of this redirection chain.

We used the Babelfy java API service (Moro et al., 2014) to address the task of entity linking to DBpedia 2015-04 resources. It is a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation based on a loose identification of candidate meanings coupled with a densest subgraph heuristic which selects high-coherence semantic interpretations (Moro et al., 2014).

The Babelfy parameters that we tuned according to our preferences are:

[43] http://wiki.dbpedia.org/

setAnnotationType was set to identify both concepts and named entities, setMatchingType was set to exact matching, setMultiTokenExpression was on to identify multi-word tokens, setScoredCandidates was set in a way so that it obtains only top-scored candidate from the disambiguation list. The rest of the parameters were kept to their default value. The named entities identified by both Babelfy and ARK Tagger were allowed to the linking stage. Initially, we provided the original tweet texts as input to Babelfy. We observed that the number of named entities and concepts recognized and linked solely by Babelfy service was quite low. The named entity recognition suffered because of the noisy nature of tweet text. However, the accuracy of the linked resources was satisfactory. So, we modified our system by altering the tweets slightly. We removed the # and considered only the alphabets from an already recognized named entity (tagged by the ARK tagger). After successfully linking such named entities, we searched for more entities which were syntactically similar to the previously known entities. We linked these new entities to corresponding DBpedia resources and also obtained the disambiguation scores.

4.4.5 Clustering of NIL Mentions

The entities which could not be linked to any existing DBpedia resource are supposed to have NIL identifiers so that each NIL may be reused if there are multiple mentions in the text which represent the same (s/similar/identical) entity. We have considered only a spelling based approach here to calculate the similarity between entities. Two unlinked entities are taken to be similar if one of them contains the other (letter only). In that case, the new entity is assigned the same NIL identifier as that of the previous one.

4.5 Results

We evaluated our approach on the development set consisting of 100 tweets made available by the organizers. In Table 4.1 we have reported on the official metrics for entity detection, tagging, clustering and linking. The precision, recall and f-scores for the above-mentioned three runs show that the Run 3 produces best results for the task with f-score 0.674, 0.380, 0.252 and 0.646 in the categories Strong Mention Match, Strong Typed Mention Match, Strong Link Match and Mention Ceaf respectively.

While all the Runs yield same score in other categories, in Strong Typed Mention Match, we observe better result for our feed-forward neural network model. Our systems for the three different runs only differ in entity type classification module while all other subtasks follow the

same system in all three cases. This results in same result in the last two categories which were mainly the evaluation metrics for linking and nil clustering.

Table 4.1. Summary of Experimental Results.

	Precision	***Recall***	F1
Run 1			
Strong Mention Match	0.729	0.626	0.674
Strong Typed Mention Match	0.301	0.259	0.278
Strong Link Match	0.586	0.161	0.252
Mention ceaf	0.699	0.600	0.646
Run 2			
Strong Mention Match	0.729	0.626	0.674
Strong Typed Mention Match	0.144	0.124	0.133
Strong Link Match	0.586	0.161	0.252
Mention ceaf	0.699	0.600	0.646
Run 3			
Strong Mention Match	0.729	0.626	0.674
Strong Typed Mention Match	0.411	0.353	0.380
Strong Link Match	0.586	0.161	0.252
Mention ceaf	0.699	0.600	0.646

4.6 Error Analysis

If we look at the confusion matrix for different languages, we can notice that many other languages have been wrongly classified as English. This is primarily due to overlapping words between English and all other Indian languages. In our task, the accuracies of MIXes and NEs were quite low. The primary reason for the increased error rate in MIX determination was the absence of post processing measures to identify the mixed words. Also the sub-classification errors in NE recognition could have been significantly reduced by adding a NE-classification module to our system. Our accuracy also declined for Gujarati, Kannada and Malayalam. Use of larger wordlists and transliterated dictionary should have improved the scores.

4.7 Conclusion

In this Chapter, we have described our approach for the #Microposts2016 Named Entity rEcognition and Linking (NEEL)[44] challenge. We have developed a hybrid system using the existing Named Entity Recognizer systems and Twitter-specific Part-of-Speech Taggers in conjunction with the classifier developed by us. The Named Entity Linking was done mainly by

[44] http://microposts2016.seas.upenn.edu/challenge.html

using Babelfy[45], which performs as a multilingual encyclopaedic dictionary and a semantic network. It should be kept in mind that Named Entities are often considered to be universal as they are not dependent on any language. Therefore our Named Entity Detection and Classifications system is independent of language and can be seamlessly extended to any language that is used in social media.

[45] http://babelfy.org/

CHAPTER

5 SENTIMENT IDENTIFICATION AND POLARITY CLASSIFICATION IN SOCIAL MEDIA TEXT

Chapter 5

Sentiment Identification and Polarity Classification in Social Media Text

Sentiment analysis is the Natural Language Processing (NLP) task dealing with the detection and classification of sentiments in texts (Balahur et al., 2013). While some tasks deal with identifying presence of sentiment in text (Subjectivity analysis), other tasks aim at determining the polarity of the text categorizing them as positive, negative and neutral. Whenever there is presence of sentiment in text, it has a source (people, group of people or any entity) and the sentiment is directed towards some entity, object, event or person. Sentiment analysis tasks aim to determine the subject, the target and the polarity or valence of the sentiment.

Extraction of sentiment from social media – like Facebook[46] or microposts like Twitter[47] – can serve a myriad of purposes. These texts often express opinion about a variety of topics. It can be the appraisal of the user about certain product or incident, the state of mind of the speaker or any intended emotional communication that he may want to have with potential readers. In our work, we try to automatically extract sentiment (positive or sentiment) from the posts. This analysis can be put into a variety of uses. Consumers can use sentiment analysis while researching products prior to purchase. Organizations can determine public opinion about their products and services. Similarly, cyber crime departments can identify cyber bullying prevalent in the web space.

5.1 Introduction

Sentiment analysis – sentiment analysis of social media in particular - has become a popular area of research in present times. The massive proliferation of social media has been a catalyst in this regard. A culture shift where in the users comfortably and candidly express their emotions, opinions or sentiments online has also aided in harnessing sentiments from social media. User reviews on ecommerce sites, opinions on web blogs, tweets and Facebook posts, each of them

[46] www.facebook.com
[47] twitter.com

can be mined for assessing polarity of opinion. Businesses use the power of text analytics behind their data mining technology. Sentiment analysis helps businesses in advertising, marketing and making business decisions for better customer satisfaction. It can also be used to investigate the web for forecasting electoral results (by evaluating voter sentiment) and track political preferences. Of recent, social media analysis has been used extensively in identifying cyber-bullying.

Although we have come across various tasks conducted on multilingual texts, the task of sentiment analysis, in particular, has not been explored for multilingual code-mixed texts. This type of text differs significantly from traditional English texts and needs to be processed differently. However, different forms of texts require different methods for sentiment analysis. For example, if we look at sentiments in scientific papers, it is hedged and indirect while the sentiments are more direct in movie or product reviews. Traditional texts like reviews and newspaper are structured and follow a definite pattern. Also, the writing is more formal and composed. Social media texts on the other hand are largely informal. They are concise and informal with several linguistic differences.

In our work, we have used code-mixed social media data which have been collected from Facebook post. The text is informal and conversational in accordance with social media characteristics. It is mostly bilingual though the presence of three languages in a single post is not entirely uncommon in our data. Initially, we pre-process the text to normalize the irregular words. We also remove noise from the text prior to processing it and translate the abbreviations to regular words wherever applicable. However, we do not use part-of-speech tagging as the system shows minimal improvements using part-of-speech features. We make use of various word-level, dictionary-based and stylistics features relevant to social media text to classify the sentiment as subjective or objective. Subjective posts are further categorized as positive or negative in polarity. We use various machine learning algorithms for our final classification. Artificial neural network model performs best in our experiments.

The remainder of this Chapter is structured as follows: Section 5.2 gives an overview of the background and related work. In Section 5.3, we present the dataset. The working model for our system is described in Section 5.4. We describe in detail the pre-processing and feature selection used to build the classification models. In Section 5.5, we present the results obtained using different combinations of features. We evaluate the performance of various machine learning models that we used in our experimentation. Section 5.6 summarizes the main findings of this work and sketches the lines for future work.

5.2 Related Work

Research regarding emotion and mood analysis in text – is becoming more common recently, in part due to the availability of new sources of subjective information on the web. The work of Ortony et al. (1987) was one of the very first in the area of sentiment classification. They focussed on the actual taxonomy and isolation of terms with an emotional connotation.

Identifying the semantic polarity (positive vs. negative connotation) of words has been done using different approaches. Some of the works (knowledge-based) explicitly attempts to find features indicating that subjective language is being used. Hatzivassiloglou and McKeown (1997) made use of corpus statistics, Wiebe (2000) used linguistic tools such as WordNet (Kamps et al., 2004), and Liu et al. (2003) used lexicon-based classifier. Turney's (2002) work on classification of reviews used an unsupervised learning technique based. They found the mutual information between document phrases and the words "excellent" and "poor", where the mutual information was computed using statistics gathered by a search engine. In their work on automatic classification of sentiment in online domains, Pang et al. (2002) evaluated the performance of different classifiers on movie reviews. They demonstrated that that standard machine learning techniques outperform human-produced baselines.

Typically, methods for sentiment analysis produce lists of words with polarity values assigned to each of them. This method have been successfully employed for applications such as product review analysis and opinion mining (Das et al., 2001; Dave et al., 2003; Grefenstette et al., 2004; Pang et al., 2002; Nasukawa and Yi, 2003; Turney and Littman, 2003; Esuli et al., 2006). Holzman and Pottenger (2003) reported high accuracy in classifying emotions in online chat conversations by using the phonemes extracted from a voice-reconstruction of the conversations. Rubin et al. (2004) investigated discriminating terms for emotion detection in short text while Read (2004) described a system for identifying affect in short fiction stories, using the statistical association level between words in the text and a set of keywords. In another work, Read (2005) used distant supervision to build the corpus.

There has been some work by researchers in the area of phrase level and sentence level sentiment classification (Wilson et al., 2005) and on analyzing blog posts (Mishne, 2005). Wilson et al. determined whether an expression is neutral or polar and then disambiguated the polarity of the polar expressions. With this approach, their system was able to automatically identify the contextual polarity for a large subset of sentiment expressions.

Sentiment analysis of social media text has received a lot of interest from the research community in the recent years with the rise to prominence of Facebook and Twitter. Ding et al. (2008) used context-dependent sentiment words in their work and Tan et al. (2008) suggested combining learning-based and lexicon-based techniques using a centroid classifier. Go et al. (2009) used positive and negative emoticons to classify tweet polarity. They showed that machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) have accuracy above 80% when trained with emoticon data. Pak and Paroubek (2010) showed how to automatically collect a corpus for sentiment analysis and opinion mining purposes. They concluded that authors use syntactic structures to describe emotions or state facts and some POS-tags may be strong indicators of emotional text. They obtained best results using Naïve Bayes classifier that uses N-gram and POS-tags as features. Diakopoulos et al. (2010) used crowdsourcing techniques to manually rate polarity in Twitter posts. In their work, Chowdhury et al. (2012) classified human affective states from posts shared on Twitter. Wang and Manning (2012) highlighted the suitability of Support Vector Machine or Naïve Bayes for different domains. Our approach is similar to that of Zhang et al. (2011) who presented the idea of ternary classification system (positive, negative and neutral). They used target words bearing sentiment and supervised learning for classification. We also use some techniques for noise reduction which was inspired by Hu et al. (2013). They proposed building a sophisticated feature space to handle noisy and short messages in their work on Twitter sentiment analysis.

5.3 Dataset

A recent shared task was conducted by Twelfth International Conference on Natural Language Processing (ICON-2015)[48], for part-of-speech tagging of transliterated social media text. For the shared task in that corpus, data was collected from Bengali-English Facebook chat groups. The sentences are in mixed English-Bengali and English-Hindi – and have been obtained from the "JU Confession" Facebook group, which contains posts in English-Bengali with few Hindi words in some cases.

We have modified the ICON Shared Task Corpora for our work on sentiment analysis. The dataset contains three languages – Bengali, Hindi and English. The data set contains 882 sentences in total. The statistics for the dataset have been presented in the following table.

[48] http://ltrc.iiit.ac.in/icon2015/contests.php

Table 5.1. Statistics of the Corpus.

Language Tags	*Number Of Words Present*	*Percentage Of Corpus*
English (En)	9988	52.72
Bengali (Bn)	8330	43.97
Hindi (Hi)	626	3.3

The purpose of the implementation is to be able to automatically classify a tweet as a positive or negative tweet sentiment wise. The classifier needs to be trained and to do that we needed a list of manually classified tweets. We used 2 annotators to classify the tweets into three categories – positive, negative or neutral.

We have calculated Kappa co-efficient to measure the inter-annotator agreement. Kappa co-efficient is a reliable and robust measure to measure the agreement between two users. It takes into account the agreement occurring by chance and hence, is more useful than percent agreement calculation.

$$\kappa = \frac{po - pe}{1 - pe}$$

where p_o is the relative observed agreement among annotators, and p_e is the hypothetical probability of chance agreement. The observed data is used to calculate the probabilities of each observer randomly saying each category. If the annotators are in complete agreement then $\varkappa = 1$. If there is no agreement among the annotators, other than chance agreement (as given by p_e), then $\varkappa \leq 0$.

Table 5.2. Inter-Annotator Agreement.

	Annotator 2			*Total*
Annotator 1	***Positive***	***Neutral***	***Negative***	
Positive	200	146	13	359
Neutral	46	268	26	340
Negative	6	80	97	183
Total	252	494	136	

For the above data, p_o is 0.641 and p_e is 0.3642, therefore giving a Kappa co-efficient of 0.4354.

As we can see that the Kappa measure is low, so we have obtained the sentences where the annotators are unanimous about the sentiment polarity. There are a total of 565 such sentences. We have used these sentences for our sentiment polarity classification.

5.4 System Description

The process of sentiment analysis can be divided into three major parts – pre-processing of raw posts, feature identification and extraction and finally, the classification of sentiment as positive, neural or negative. The steps have been discussed in sequential order.

5.4.1 Expansion of Abbreviations

As social media text is often non-traditional and informal in nature, the posts had to be pre-processed initially to remove noise. We have used an abbreviation list to normalize all the words that were abbreviated. For example, *btw* was replaced by "*by the way*", *clg* by "*college*", *hw* by *"how"* and so on.

5.4.2 Removal of Punctuations

Before processing the post any further, we remove all punctuations from the text. Social media text usually contains a lot of punctuations and their usage is often arbitrary in nature,

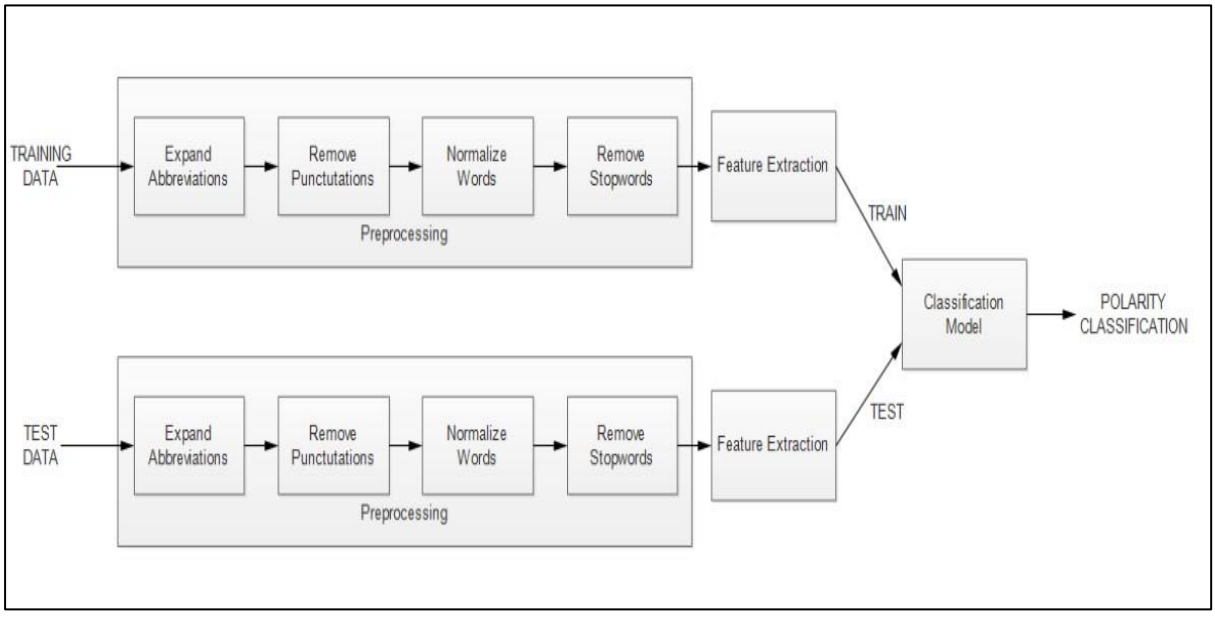

Fig 5.1. Overview of the System Architecture.

not adhering to grammatical norms. To compound the problem further, punctuations like stop, question mark and exclamation marks are often used multiple times in succession. By removing all the punctuations, we try to make our text as noiseless as possible. We keep a record of the number of different punctuations in the text which has been used as a feature for classification.

5.4.3 Removal of Multiple Character Repetitions

It is often found in social media text that certain characters are repeated more than once. These non-conformational spellings are very hard to deal with as they cannot be successfully matched to any dictionary. For example, *lol* (abbreviated form of *laughing out loud*)can be written as *loool*, *looool* or *loooooool*. We use pre-processing in order to reduce all these occurrences to *lool*. Any character which occurs more than two times in a row is replaced by two occurrences of the same character. Some other examples are *ahhhh* (reduced to *ahh*) and *uhhhh* (reduced to *uhh*). However, we maintain a record of the number of repetitions as this could be used by the author in specific situations to reflect sentiment.

5.4.4 Feature Extraction

In our work, we used the following features to train our machine learning model.

- ***Number Of Word Matches With Sentiwordnet (SWN)***

  We have used SentiWordNet[49] as one of the sentiment resources. SWN is a lexical resource for sentiment analysis. It assigns three sentiment scores – positivity, negativity and objectivity to each synset of WordNet. So, a given word can have a positive or negative score or both. We have extracted all the positive and negative words from SWN. The final list contains 17027 positive words and 17992 negative words. For a given sentence, we find if the normalized words are a match with any words in these two lists. We find the number of positive and negative matches and the difference between the two is assigned as the feature.

- ***Number Of Word Matches With Opinion Lexicon (OL)***

  Similar to SentiWordNet, Opinion Lexicon[50] is another lexical resource for sentiment analysis. It contains a list of positive and negative opinion words or sentiment words for English. There is a total of 2006 positive words and 4783 negative words. We find the number of matches to both the lists and the difference is taken as our second feature.

- ***Number Of Word Matches With English Sentiment Words (ESW)***

  We have collected a list of positive and negative words from the internet for sentiment classification. This list contains 3075 positive words and 4003 negative words. This list concentrates more on the words which appear in social media context. Similar to the

[49] http://sentiwordnet.isti.cnr.it/
[50] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

previous two features, we find the number of matches to the positive and negative lists and the difference between the two is considered as our third feature.

- ***Number Of Word Matches With Bengali Sentiment Words (BSW)***

  This list was developed to tackle the presence of sentiment in Bengali words. As we are dealing with multilingual text, it was essential to develop this list for Bengali. Das et al. (2010; 2011; 2012) developed SentiWordNet for Indian Languages. However, this list contained words in Bengali (or Brahmic) scripts. As we are dealing with transliterated text, this wordlist required transliteration to English. Finally, we developed a positive and a negative wordlist for transliterated Bengali words. The number of words in the positive wordlist is 1778 while the negative wordlist contains 3713 words. The difference in number of matches to both the lists is considered as our next feature.

- ***Number Of Colloquial Bengali Sentiment Words (CBW)***

  We have created this list for Bengali words which often appear in social media text. It must be noted that Bengali Sentiment Words developed previously is more formal in nature and therefore, not sufficient for identifying colloquial words which appear in Facebook posts or Twitter texts. For example, words like *jata, hebby, phot* are not captured by Bengali Sentiment Words. We create two lists – positive and negative wordlists - tries to incorporate all such words which may indicate the presence of sentiment in the text. The number of matches to both the lists is determined and the difference is assigned as feature.

- ***Density Of Curse Or Bad Words (CW)***

  We have used a list of curse words developed by Huang et al. (2014) in their work on cyberbullying. In their work, the authors collected 713 curse words (e.g. `asshole', `bitch' etc.) and hieroglyphs (such as `5hit', `@ss' etc.) based on online resources. We have used this list to find out all the words which have been used with a negative sentiment.

- ***Part-Of-Speech Tags (POS)***

  Part-of-speech tags (JJ, RB and JJ-RB) can be considered as features to detect presence of sentiment in commonly occurring unigram and bigrams in the training data.

- ***Number Of All Uppercase Words (UW)***

  Based on the findings of Dadvar et al. (2013), capital letters can represent shouting or strong opinion in online chats and posts. We have identified the number of words in a post which are written in all capital letters. This is used as a feature to detect the presence of emotion or sentiment in online settings.

- ***Density Of Exclamation Points (E)***

  Just like the uppercase letters, exclamation points also stand as emotional comments. To identify strong emotions in social media context, we chose the number of exclamation points as a feature for our model. The number of exclamation points is normalized by the number of words present in the text.

- ***Density Of Question Marks (Q)***

  Similar to the last feature, multiple question marks in the text can denote surprise, excitement or agitation of the user. We chose the number of question marks as our next feature. The number of question marks is normalized by the number of words present in the text.

- ***Number Of Character Repetitions In A Word (R)***

  It is often observed that users tend to repeat a number of characters – vowels or consonants – to stress their opinion in social media conversations. Words like *loool*, *lolzzzz*, *uffff*, *ahaaa, greaaat* are quite common in social media texts. While we reduce all such words during our pre-processing step, we have also maintained a record of all such occurrences. These repetitions are often indicative of sentiment and we use it as one of our feature.

- ***Frequency Of Code Switches (CS)***

  As we are dealing with multilingual texts, we have considered the frequency of code switching as one of our features. It is often observed that the writer shifts language to clarify his opinion. We have tried to exploit this social and communication needs for this language shifting to determine the presence of sentiment. This frequency is normalized by the number of words in a particular post.

- ***Number Of Smiley Matches (S1 And S2)***

  Smileys are quite prevalent in social media text and often form a primary way of expressing emotion. We have created two resources for identifying smiley in text. The first one contains 269 positive smileys and 170 negative smileys. The second list contains 243 smileys. We found the number of matches to both the lists and used it as a feature.

5.4.5 Classification of Sentiment Polarity

We obtain results for the 565 posts for which both the annotators agreed on the polarity. We use 70% of the dataset for training and 30% for testing purposes.We split the dataset using 400 posts for training and 165 posts for testing.

We use the machine learning software WEKA[51] (Hall et al., 2009). We combine the above features to form a feature set and employ a number of machine learning algorithms for classification. The best results were produced by Multilayer Perceptron model. This classifier uses back propagation to classify instances into three categories – *positive, negative* and *neutral*. The nodes in this network are all sigmoid. The learning rate and momentum rate for the back propagation algorithm was kept at 0.3 and 0.2 respectively. The number of epochs was set to 500 and the random number generator was seeded using value 0.

Individually, none of the features was able to detect positive or negative instances in citation. This is due to the biasness of the system. We perform feature analysis by removing one feature at a time to determine if any feature is more important than the other. We also check by adding one feature group at a time.

The classification confidence score from WEKA and the number of matches to our citation specific lexicon is used to develop a post-processing algorithm.

5.5 Results and Observations

In the following sections, we perform feature analysis and present the results of our classification task.

5.5.1 Feature Analysis

For feature analysis, we have grouped the different kind of features and obtained the impact of each group in classification. We have grouped the word (or dictionary) based features into Group 1, semantic features into Group 2 and the style based features into Group 3.

Group 1: SWN + OL + ESW + BSW + CBW + CW + S
Group 2: POS
Group 3: UW + E + Q + R + CS

Table 5.3. Impact of Adding Each Feature Iteratively To the Last.

Feature added	*Number of correct classifications*	*Number of incorrect classifications*	*Accuracy*
Group 1	110	55	0.667
Group 1 + Group 2	**113**	**52**	**0.685**
Group 1 + Group 2 + Group 3	101	64	0.612

[51] http://www.cs.waikato.ac.nz/ml/weka/downloading.html

From Table 5.3 it is evident that word based features (Group 1) and semantic features (Group 2) produce the best results collectively. The accuracy decreases when we include the style based features for classification.

Table 5.4. Impact Of Each Feature Calculated By Eliminating One at A Time.

Feature eliminated	*Number of correct classifications*	*Number of incorrect classifications*	*Accuracy*
None	104	61	0.630
SWN	109	56	0.661
OL	103	62	0.624
ESW	102	63	0.618
BSW	104	61	0.630
CBW	101	64	0.612
S	105	60	0.636
POS	**100**	**65**	**0.606**
UW	110	55	0.667
E	107	58	0.649
Q	105	60	0.636
R	107	58	0.649
CS	106	59	0.642
S1	**100**	**65**	**0.606**
S2	106	59	0.642

Table 5.4 serves to highlight the impact of individual features in classification. At each turn, we eliminate one of the features while keeping all the other features. The accuracy suffers the maximum on elimination of POS (*JJ*, *RB* and *RB_JJ*) features and the polar smiley list. Elimination of all the style based features (UW, E, Q, R and CS) shows improvement in accuracy. This is in accordance to our findings in Table 5.3. Elimination of SWN also improves accuracy. Removing BSW – which comprises of conformational (or traditional) Bengali words – do not affect accuracy proving the fact that social media text requires tailor-made resources.

5.5.2 Results

Table 5.5 shows the confusion matrix for the polarity classification (using word based and semantic features). The precision, recall and f-measure of the supervised and baseline systems are compared in Table 5.6.

Table 5.5. Confusion Matrix for Classification.

	Positive	*Neutral*	*Negative*
Positive	25	23	4
Neutral	10	78	3
Negative	4	8	10

Table 5.6. Precision, Recall and F-measure.

	Precision	***Recall***	***F-measure***
Class Positive	0.641	0.481	0.275
Class Neutral	0.716	0.857	0.39
Class Negative	0.588	0.455	0.257

If we consider the baseline model to contain all the instances of neutral polarity, then we can achieve an accuracy of 55.2%. Our best performing system shows an accuracy of 68.5%. So we can see that our supervised system shows improvement over the baseline model. However, the learning algorithm was slightly biased towards neutral classification which is evident from the confusion matrix. Most of the errors are due to positive and negative citations being identified as neutral.

In future works, we will need to fine tune our classification features so that the system can identify positive and negative citations more efficiently. Also using a larger dataset to train the system would eliminate the bias towards neutral classification of polarity.

5.6 Conclusion

As per our knowledge, there exists no sentiment classifier for code-mixed social media text. We have performed a machine learning based sentiment classification of Facebook posts. The polarity of each post has been classified as *positive*, *negative* and *neutral*. As there has not been any similar work before, we had to create a dataset of our own. Two human annotators classified the polarity of each post. Due to the inherent complexity of social media text, use of arbitrary emoticons and presence of sarcasm, the agreement between the human annotators was quite low with a Kappa co-efficient of 0.4354. Although the entire dataset consists of 882 posts, we have used only 565 posts where the annotators were unanimous about the polarity of underlying sentiment. We used word-based, semantic and style-based features for classification. The best result was obtained using a combination of word-based and semantic features with an accuracy of 68.5%.

CHAPTER

6 COMPLEXITY METRIC FOR CODE-MIXED SOCIAL MEDIA TEXT

Chapter 6

Complexity Metric for Code-Mixed Social Media Text

Any data or system requires an evaluation metric for measuring the performance and complexity. In this Chapter, we have discussed the level of complexity in code-mixed social media text. Therefore, we have modified two existing corpora to suit the requirements of our study. In general, text written in multiple languages is often hard to comprehend and analyze. Also, for the purpose of analysis, it may be necessary to determine the complexity of a particular document or text segment. In particular, one often needs to compare two different corpora with regards to their complexity. Thus, in the present Chapter, we have employed the existing metrics for determining the code-mixing complexity of a corpus. Finally, we discuss their advantages and shortcomings and propose some improvements on the existing metric which would better reflect the variety and complexity of a multilingual document.

6.1 Introduction

Social media text differs from other conventional text in many ways. It is noisy in nature and requires comprehensive processing from a multilingual point of view. In social media communication, a multilingual speaker often uses more than one language. The communication, inherently informal in nature, presents the scientific community with a challenging yet interesting problem.

First of all, we need to understand the necessity of language switching. Is it motivational? Or is it circumstantial? Although mixing of languages was prevalent in verbal communication, it was the proliferation of social media which accelerated the use of multiple languages in a single written communication. This is motivated by both social and conversational needs (Auer, 1984). Sometimes, the speaker is not competent in the language he is writing. The lack of vocabulary makes him use words from his native language as a substitute. On some other ocassions, the need is purely social and is used by the writer to mark him as part of a large group.

Automatic identification of the code switching points is important as it helps to understand the frequency of code switching or code mixing and subsequent complexity of the text. Also, it would allow us to determine the language specific models which are better suited for the analysis of such text. It is also important to understand the difference between code switching and code mixing? Both the terms are used interchangeably in the literature. In our work, the term 'code switching' refers to inter-sentential language shift while the term 'code-mixing' refers to intra-sentential shifts of language.

In this work, we have used short utterances collected from Facebook page and Twitter data for our analysis. As the dataset is based on Indian social media text, it is essential that we give a brief statistics about the degree of multilingualism in India. There are more than 20 officially recognized languages in India. The number of Hindi speakers range from 14.5% to 24.5% in total population (Wikipedia[52]). Other languages are spoken by 10% or less of the population. English and Hindi are mostly used for official communication. The diversity of language and needs for faster and efficient communication motivates the mixing of languages in social media context.

This brings us to the problem of transliteration. Most of the time, languages like Hindi or Bengali is not written using their native scripts - Devanagari for Hindi and Eastern Neo Brahmi script for Bengali. Instead, the users prefer using Roman script as it is more convenient with a regular keyboard.

While analyzing a code-mixed transliterate text, it is often useful to determine the complexity of the corpus. For any task on code-mixed corpora – language identification, part-of-speech tagging, information retrieval and question answering system – it is important for the researchers to compare the difficulty of their work with regards to the level of language mixing in the text. Also, it is expected that with increasing complexity and more code-mixing in a text, the accuracy of text processing would decrease and the error rates would increase.

In the Section 6.2, we discuss the previous work which has been done in related area. In Section 6.3, we provide a brief statistics of the corpus and its preparation. The index is presented in Section 6.4. The working of the index is further elaborated using examples in Section 6.5. Section 6.6 contains the results and Section 6.7 concludes the Chapter and discusses scope of future work.

[52] https://en.wikipedia.org/

6.2 Related Work

In 2001, Kilgarriff (2001) discussed the pointed out that corpus linguistics do not have proper methods for comparing corpora. Most of the corpus descriptions are textual and based on the opinion of the researcher. Such impressions are highly subjective and not a proper measure of corpus similarity or complexity. Whenever we are working on a new corpus, the question that inevitably arises is about the limitations and benefits of using that corpus. The size and homogeneity of data are some of the factors which have been used intensively. However, such approaches are mainly word based and are based on monolingual text.

Measuring corpus similarity has a wide array of applications. It has theoretical and research applications – where one can judge the complexity of the dataset before performing the analysis. Also one may want to replace a dataset with another. It is beneficial if there is some way to determine if the two datasets are similar and comparable in terms of complexity and use. This would help in inter-domain portability of NLP systems.

As per my knowledge, Gambäck and Das (2014) proposed the first index for code-mixed social media text. Termed as Code Mixing Index (CMI), the index tries to assess the level of code-switching in an utterance. The measure aimed at comparing one code-switched corpus with another. Das and Gambäck (2014) worked on Hindi/Bengali-English Facebook chat groups. The corpora introduced by them 28.5% of the messages written in at least two languages. CMI can be described as the fraction of total words that belong to languages other than the most dominant language in the text,

$$CMI = 100 * [1 - \frac{max\{wi\}}{n-u}] \text{ , if n>u}$$

$$CMI = 0 \text{ , if n = u}$$

where $n - u$ is the sum of N languages present in the utterance of their respective number of words and $\max\{wi\}$ is the highest number of words belonging to a particular language, n is total number of tokens and u is the number of language independent tags. Das and Gamback (2014) averaged the CMI values for all the sentences and for only the code-mixed sentences to obtain 'CMI all' and 'CMI mixed' respectively.

However, CMI considers only the fraction of words in the corpus which are code-switched. We have used CMI as initial parameter and have suggested some improvements which would take

into account the number of languages present in the corpus and the number of code-switching points present.

6.3 Corpus Preparation

Our task required social media corpus which has diversity in terms of languages and code-mixed content. Forum for Information Retrieval Evaluation[53] (FIRE) organized a shared task on Mixed Script Information Retrieval. The data set used for training and test suited our purpose perfectly. Another shared task which we participated in was organized by the Twelfth International Conference on Natural Language Processing[54] (ICON-2015). This data set was bilingual in nature and used code-mixed social media text. We have modified these corpuses for our task.

6.3.1 The FIRE 2015 Shared Task Corpus

For our research, we have modified the transliterated corpus which was part of FIRE 2015 Shared Task on Mixed Script Information Retrieval. The data set was composed of 3701 sentences and 63526 word tokens. Each word may belong to one of the nine languages present in the entire dataset. The nine languages were Bengali (Bn), English (En), Gujarati (Gu), Hindi (Hi), Kannada (Ka), Malayalam (Ml), Marathi (Mr), Tamil (Ta) and Telugu (Te). The organizers made an exceptional job of collecting the data which is extremely multilingual. The languages present in the dataset are the most prevalent ones that we can find in Indian social media. It must be noted that the words of a single query usually come from 1 or 2 languages and very rarely from 3 languages. This is in line with the language mixing trends that we have witnessed in social media context. The users, even if familiar with multiple languages, rarely uses more than three languages while writing posts or tweets. As a matter of fact, most of the sentences are bilingual in nature with one of the languages as either En or Hi. Thus, we have sentences that mix Ta and En words, or Bn and Hi words, but not for example, Gu and Ka words. The named entities (marked as NE), language independent words (marked as X) and mixed words containing intra-word language switches (marked as MIX), were all considered undefined and assigned UN tag. The number of utterances, tokens for each language pair in the training set is given in the following table.

[53] http://fire.irsi.res.in/fire/2015/home
[54] http://ltrc.iiit.ac.in/icon2015/

Table 6.1. Statistics of FIRE 2015 Corpus.

Language Tags	*Number Of Sentences*	*Number Of Words Present*	*Percentage Of Corpus*
English (En)	2665	21996	34.63
Bengali (Bn)	355	4919	7.74
Gujarati (Gu)	165	1075	1.69
Hindi (Hi)	614	5897	9.28
Kannada (Ka)	373	2212	3.48
Malayalam (Ml)	151	1390	2.19
Marathi (Mr)	229	2414	3.8
Tamil (Ta)	342	3694	5.82
Telugu (Te)	603	7002	1.1
Language Independent	2582	12927	20.35

6.3.2 The ICON 2015 Shared Task Corpus

Another recent shared task was conducted by Twelfth International Conference on Natural Language Processing (ICON-2015), for part-of-speech tagging of transliterated social media text. For the shared task in that corpus, data was collected from Bengali-English Facebook chat groups. The sentences are in mixed English-Bengali and English-Hindi – and have been obtained from the "JU Confession" Facebook group, which contains posts in English-Bengali with few Hindi words in some cases.

We have modified the ICON Shared Task Corpora for our work on developing the index. The dataset contains three languages – Bengali, Hindi and English. The data set contains 2341 sentences and 38199 word tokens. The statistics for the dataset have been presented in the following table.

Table 6.2. Statistics of ICON 2015 Corpus.

Language Tags	*Number Of Sentences*	*Number Of Words Present*	*Percentage Of Corpus*
English (En)	1563	15435	40.41
Bengali (Bn)	1059	13002	34.04
Hindi (Hi)	153	1006	2.63
Language Independent	2268	8756	22.92

6.4 Complexity Factor (CF)

We introduce an index, termed hereafter as Complexity Factor (CF) to measure the complexity of a multilingual corpus. This index can be applied to any document which contains multiple languages. The index uses the concept of CMI as proposed by Gamback and Das (2014) and makes some practical additions on it.

Complexity (CF) considers three different aspects while analyzing any text - Language Factor (LF), Switching Factor (SF) and Mix Factor (MF). CF can be calculated for sentences and easily extended to paragraphs and entire documents. In the next section, we have proposed three variants of Complexity Factor. Complexity Factor 1 (henceforth mentioned as CF1) is a simple baseline which considers LF, SF and MF. Complexity Factor 2 (CF2) and Complexity Factor 3 (CF3) are the two indexes which have been carefully fine-tuned to efficiently represent the complexity of any transliterated text.

6.4.1 Language Factor (LF)

This factor represents the number of different languages present in a sentence as a fraction of the total number of words in the sentences. It is evident that if a sentence becomes more multilingual, the complexity increases manifold. For example,

For any given sentence, Language Factor can be defined as,

$$LF = \left(\frac{W}{N}\right)$$

Where W is the number of words and N is the number of distinct languages in the sentence.

Sentence 1: *"Boss,* ***ajkal ki korchis****? We have been getting no news about you!"*

(English Translation: Boss, what are you doing these days? We have been getting no news about you!)

Sentence 2: *"****Kal khela dekhli?*** *What a game! Virat ne toh kamaal kar diya!"*

(English translation: Did you watch the match yesterday? What a game! Virat was simply superb!)

Sentence 1 contains two languages, Bengali and English, while Sentence 2 contains three languages – English, Bengali and Hindi. In both the sentences, Bengali words are boldfaced and Hindi words are underlined.

Language Factor is 6 for Sentence 1 (W=12, N=2) and 4 for Sentence 2 (W=12, N=3). It must be noted that longer the text block we are considering, it has more probability of finding multiple languages in it. This factor is inversely proportional to Complexity Factor (CF) and rewards shorter sentences with more distinct languages in it.

The LF can range from W (in case of a monolingual text) to 1 (when each word belongs to a different language).

6.4.2 Switching Factor (SF)

It is essential to consider the number of times the writer switches from one language to the other. As the number of switches increases, it becomes more complex to analyze the text for various tasks like language identification, part-of-speech tagging, question-answering, summarization, etc.

For any given sentence, Switching Factor is defined as the ratio of number of switching points present in the sentence to the maximum number of switching points possible for that sentence. For a block of W words, the maximum number of code-switches occurs when each alternate words belong to different languages. So the maximum number of switching points for a W-word sentence is W-1. Switching Factor, denoted by SF, can be written as:

$$SF = \left(\frac{S}{W-1}\right), \qquad if\ W > 1$$

$$SF = 0, \qquad if\ W = 1$$

where S is the number of code-switches and W is the number of words in the sentences or block of text.

Consider the following example,

Sentence 1: ***Ki*** *post* ***korcho****? Public forum* ***eta***

(English translation: What are you posting? This is a public forum)

Sentence 2: *It is painful* ***je khelata harlam***

(English translation: It is painful that we lost the match)

Both the sentences contain a mix of Bengali and English words (Bengali words are boldfaced). It should be noted that while both sentences contain 3 words each in Bengali and English, the

relative arrangement of the words make Sentence 1 more complex than Sentence 2. For sentence 1, SF is 0.8 (S=4, W=6) while for sentence 2, it is only 0.2 (S=1, W=6). Thus, we can observe that Switching Factor captures this complexity perfectly. It is directly proportional to Complexity Factor (CF).

For a single word sentence, SF = 0. SF can reach a maximum value of 1 when no two consecutive words belong to the same language.

6.4.3 Mix Factor (MF)

Mix Factor, referred to as MF for the rest of the Chapter, is based on Code Mixing Index (CMI). It is the ratio of number of words which are not written in the dominant language of the sentence to the total number of language-dependent words present in the sentence. It can be written as:

$$MF = \left(\frac{W' - \max\{w\}}{W'}\right), \text{ if W'>0}$$

$$MF = 0, \text{ if W'= 0}$$

where W' is the number of words in distinct languages, i.e., the number of words except the undefined ones, max{w} is the maximum number of words belonging to the most frequent language in the sentence.

Sentence 1: *"Boss,* ***ajkal ki korchis****? We have been getting no news about you!"*

(English Translation: Boss, what are you doing these days? We have been getting no news about you!)

Sentence 2: *"****Kal khela dekhli?*** *What a game! Virat ne toh kamaal kar diya!"*

(English translation: Did you watch the match yesterday? What a game! Virat was simply superb!)

For sentence 1, MF is 0.25 (BN: 3, EN: 9) while for sentence 2, MF is 0.5 (BN: 3, EN: 3, HI: 6)

MF can range from $\left(1 - \frac{1}{W}\right)$ (when every word in the sentence belongs to a different language) to 1 (for monolingual texts).

6.4.4 Complexity Factor – the Final Index

Finally, we have combined all the three factors to formulate the Complexity Factor as,

$$CF = \left(\frac{a * MF + b * SF}{f(LF)}\right)$$

where a and b are the weights for Mix Factor (MF) and Switching Factor (SF) respectively. f is a function of Language Factor(LF) which we use as a dampening factor.

After some experimentation with the weights, we finally settled for a = 50 and b = 50. We calculated CF by having $f(LF) = LF$, by using a linear function, $f(LF) = \left(\frac{0.25}{W-1}\right)(LF - 1) + 1$ and by using a geometric function, $f(LF) = \frac{\arctan(LF)}{\Pi} + 0.75$. We have calculated CF in three different ways and discuss our results in Table X.

$$CF1 = \left(\frac{50 * MF + 50 * SF}{LF}\right)$$

$$CF2 = \left(\frac{50 * MF + 50 * SF}{\left(\frac{0.25}{W-1}\right)(LF - 1) + 1}\right)$$

$$CF3 = \left(\frac{50 * MF + 50 * SF}{\left(\frac{arctan(LF)}{\Pi}\right) + 0.75}\right)$$

We have considered MF and SF to be equally important while determining the code-mixing complexity of the social media text. However, the number of languages in social media texts is often limited to two or three. So, the impact of the LF on complexity has been dampened by the use of a linear function (in CF2) and a geometric function (in CF3). This ensures that the complexity of any given text is not heavily reduced by the language factor.

6.5 Working of the Index

We have presented a few examples to compare the performance of our index in comparison to the existing index (CMI). The examples presented are from a purely mathematical perspective and serves the purpose of illustrating the mathematical precision of the index which we proposed here. In all the examples w_i and l_i represents the word and the language at position i respectively.

Example 1: $w_1/l_1\ w_2/l_2\ w_3/l_3\ w_4/l_4\ w_5/l_5\ w_6/l_6\ w_7/l_7\ w_8/l_8\ w_9/l_9\ w_{10}/l_{10}$

The sentence contains 10 words each belonging to a different language. Ideally, any index should denote the complexity of such a code-mixed sentence as 100 (in a scale of 0-100). There is no better example of a more complex sentence than this from a multilingual perspective. CMI gives a value of 90 for such a sentence. CF2 and CF3 both give the complexity as 95.

Example 2: $w_1/l_1\ w_2/l_1\ w_3/l_1\ w_4/l_1\ w_5/l_1\ w_6/l_1\ w_7/l_1\ w_8/l_1\ w_9/l1_1\ w_{10}/l_1$

The sentence contains 10 words each belonging to the same language. Here, we would expect the index to show complexity as zero. Each of the three indexes, CMI, CF2 and CF3, gives the complexity as 0. In case of CF2 and CF3, two of the three components - the language factor, switching factor and the mix factor – are zero.

Example 3: $w_1/l_1\ w_2/l_2\ w_3/l_1\ w_4/l_2\ w_5/l_1\ w_6/l_2\ w_7/l_1\ w_8/l_2\ w_9/l1_1\ w_{10}/l_2$

The sentence contains 10 words belonging to two languages. The words are arranged such that no two consecutive words belong to the same language. CMI calculates the complexity of the sentence to be 50. The complexities, as given by CF1 and CF2 are 67.5 and 63.2 respectively (LF=5, MF=0.5, SF=1).

Example 4: $w_1/l_1\ w_2/l_1\ w_3/l_1\ w_4/l_1\ w_5/l_1\ w_6/l_2\ w_7/l_2\ w_8/l_2\ w_9/l1_2\ w_{10}/l_2$

The sentence contains 10 words belonging to two languages. The words are arranged such that first five words belong to the one language and the next five words belong to a second language. Once again, CMI calculates the complexity of the index as 50. CF1 and CF2 calculates the complexity to be 27.52 and 25.73 respectively (LF=5, MF=0.5, SF=0.11). It must be noted that Complexity Factor correctly estimates the sentence to be less complex than the previous example (which contains more switching).

The previous examples were theoretical to mainly highlight the mathematical background of the model. We have collected a few examples from our corpus to further illustrate the robustness of our index. Once again, we compare it against CMI.

Example 5: Koi ni bhai , apne dbc wale hosla ni haarte ... \ " think to score goals instead of thinking abt goalkeepers \ "

The above sentence contains 9 English and 8 Hindi words. The value of CMI is 47 while CF2 and CF3 are 23.21 and 21.31 respectively. Complexity Factor Indexes are less the CMI because there is only one language switch in the sentence.

Example 6: mari bike ma puncture padayu

In the above sentence, there are 2 English and 3 Gujarati words with 4 language switches (each alternate word belongs to a different language). Complexity Factor for the sentence is 64.22 (as in CF2) and 61.75 (as in CF3) with the highest possible Code-Switch Factor (which is 1). CMI gives a value of 40 because it considers only the fraction of non-dominant language present.

Example 7: Mi maza maharastra prem dhakvla .. tu swapnil joshi la hate karun jar saharukla support karanar asel tar saalam malakun ..... I like swapnil because he's maharastrian ... also I have never unbend opinion about you .....

In the above sentence, there are 15 English and 15 Marathi words with 5 language switches. Although the length of the sentence is quite long, it has few language switches. CF2 and CF3 recognize that aspect and assign a complexity of 28.57 and 26.02 respectively while CMI assigns it a complexity of 50.

Example 8: Steve : 10 th anniversary celebarate pannama poiduvomo - nu .

In this case, we have selected a smaller sentence with 3 English and 3 Tamil words. There is 1 language switch present. Once again, CMI assigns it a complexity of 50. CF2 and CF3 are 22.97 and 24.79 respectively. These values are less than that of Example 7 because of fewer number of language switches.

Example 9: BIG B sings the eternal journey of life well ..... " tu shola ban jo khud jalke janha rashan karde ... ekla jalo re "

This sentence contains words from 3 languages – English (9 words), Bengali (3 words) and Hindi (9 words). The high proportion of language mixing makes CMI value 57. However, the words of all the languages occur in clusters with only 2 language switches. Therefore, the CF2 and CF3 values are 30.09 and 26.86 respectively (as it considers the relative ordering of language words along with the presence of non-dominant language).

Example 10: Happy Rakshabandhan(Rakhi) ..... Piyali Kar Lipika Bisht Lopamudra Sarkar Mandira Agrawal Payel Ghosh Trishona Vanhi

This is another example which contains only two words which are language specific (1 English and 1 Bengali word). The remaining words are named entities. CMI assigns it a complexity of 50. CF2 and CF3 values are 25.45 and 23.55 respectively.

Example 11: r february te amar breakup hoy .

This sentence contains 2 English and 4 Bengali words with 4 language switches. CMI value is 33 while CF2 and CF3 values are 45.45 and 43.1 respectively. The frequent switching of languages makes this sentence more complex than usual and Complexity Factor correctly captures it.

In the following section, we discuss the range of all the indexes in both the corpuses.

6.6 Results on Different Corpora

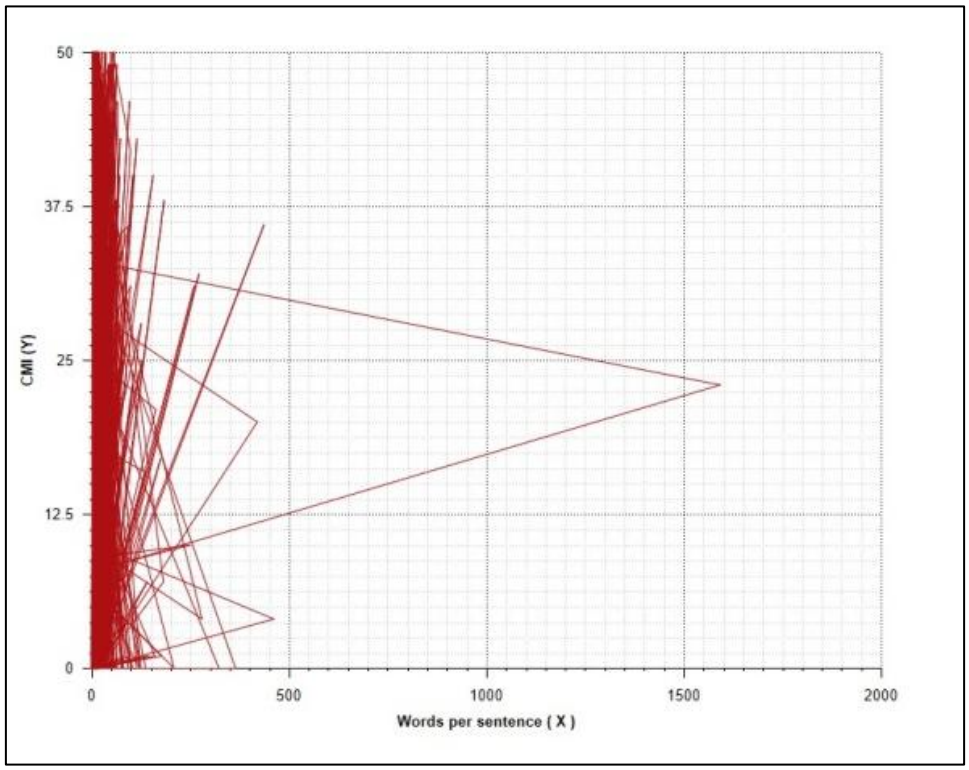

Fig 6.1. FIRE Corpus: Graph of Words per Sentence vs. CMI.

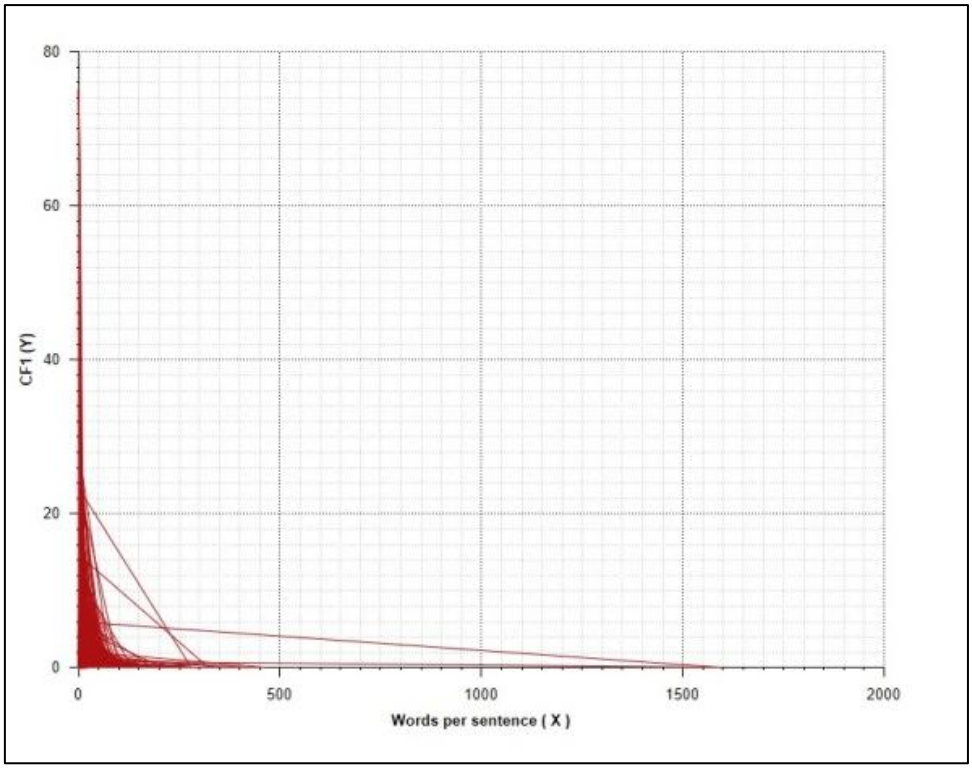

Fig 6.2. FIRE Corpus: Graph of Words per Sentence vs. CF1.

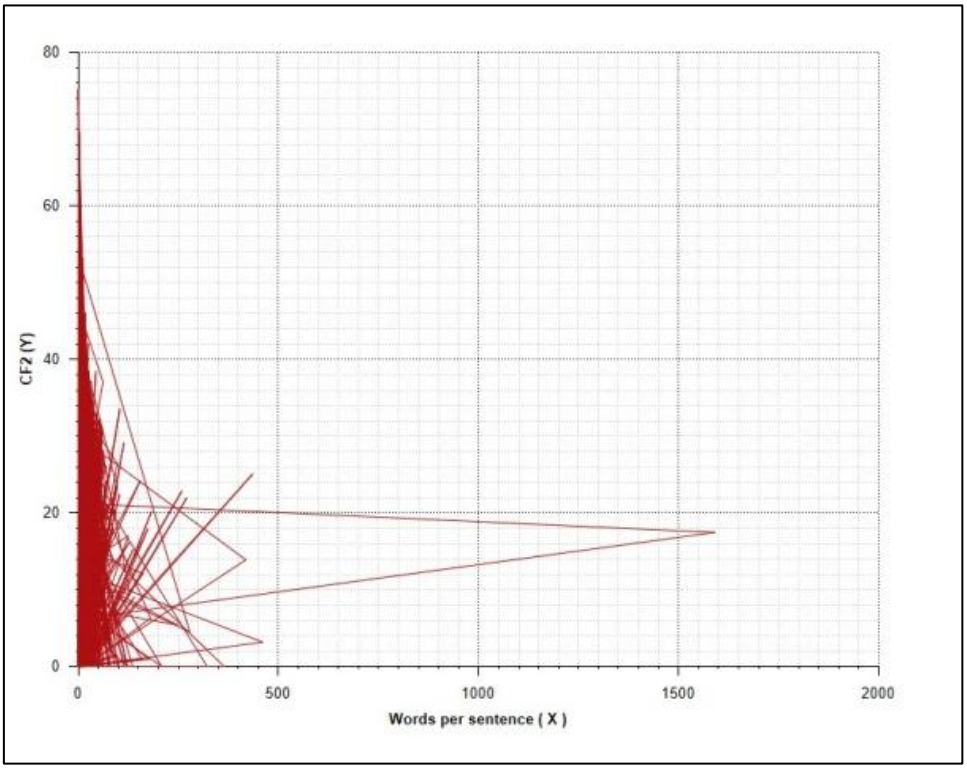

Fig 6.3. FIRE Corpus: Graph of Words per Sentence vs. CF2.

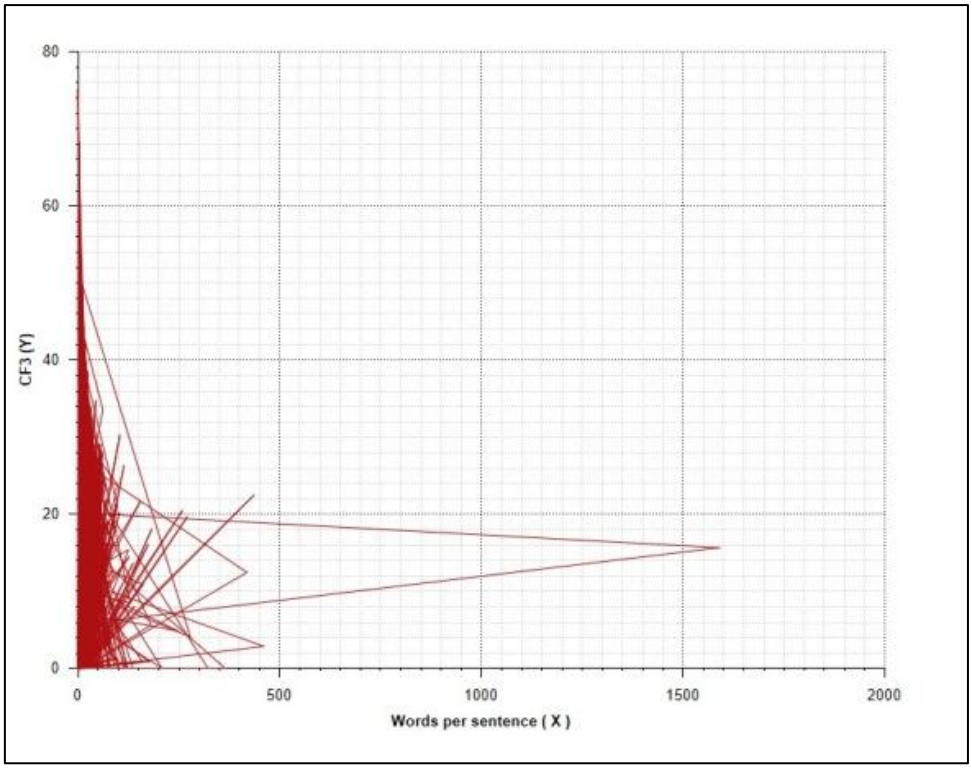

Fig 6.4. FIRE Corpus: Graph of Words per Sentence vs. CF3.

We calculated the complexity of the FIRE and ICON corpuses. Following are the results that we obtained on both the corpuses.

6.6.1 The FIRE 2015 Shared Task Corpus

The following graphs have been plotted where X axis represents the length of the sentence and Y axis represents the respective index.

Table 6.3. Range and Mean of Each Index and Words per Sentence (In FIRE Corpus).

Index	*Minimum Value*	*Maximum Value*	*Average*
CMI	0	50	11.65
CF1	0	75	2.51
CF2	0	75	10.54
CF3	0	75	9.88
Words/sentence	1	1592	17.16

6.6.2 The ICON 2015 Shared Task Corpus

Table 6.4. Range and Mean of Each Index and Words per Sentence (in ICON Corpus).

Index	*Minimum Value*	*Maximum Value*	*Average*
CMI	0	57	5.73
CF1	0	33.5	1.02
CF2	0	50	4.83
CF3	0	47.38	4.51
Words/sentence	1	367	16.32

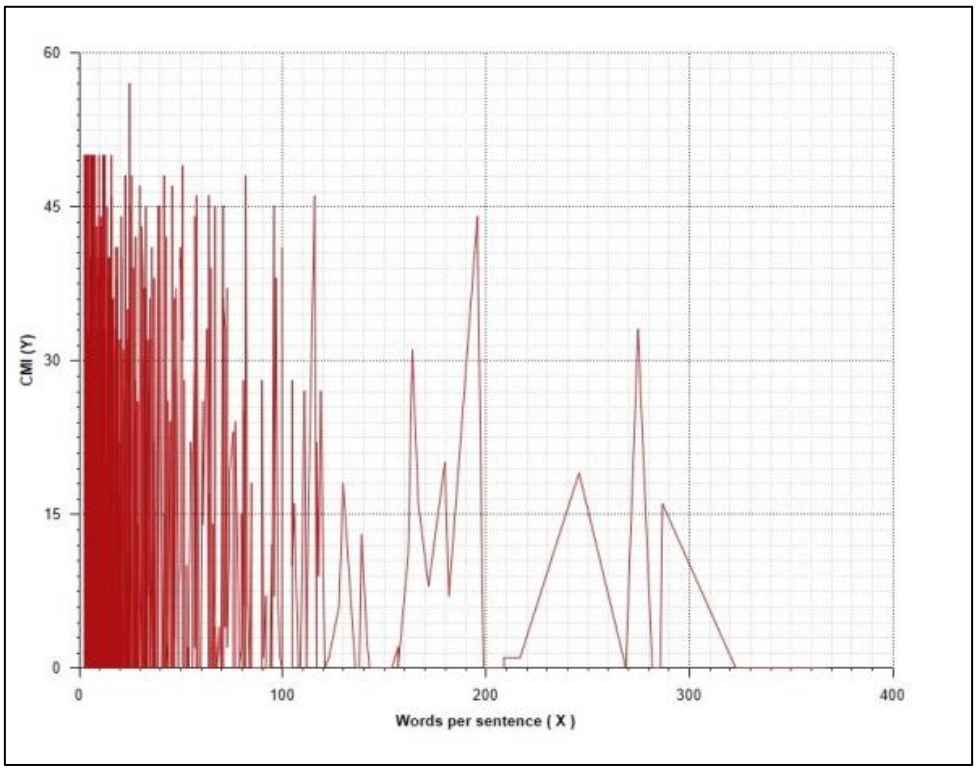

Fig 6.5. ICON Corpus: Graph of Words per Sentence vs. CMI.

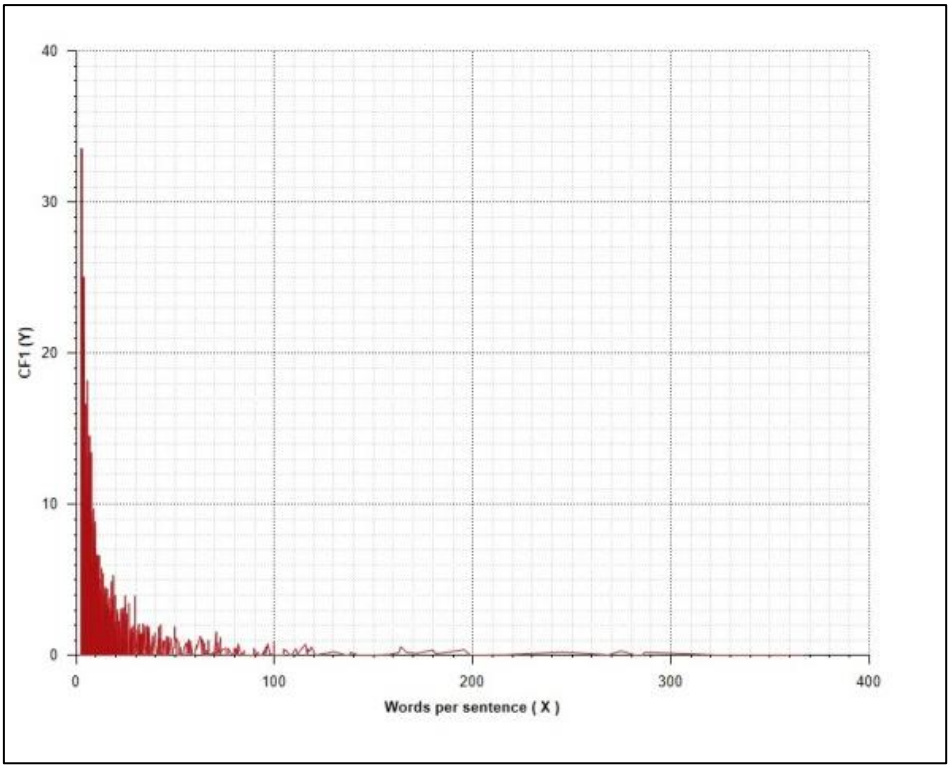

Fig 6.6. ICON Corpus: Graph of Words per Sentence vs. CF1.

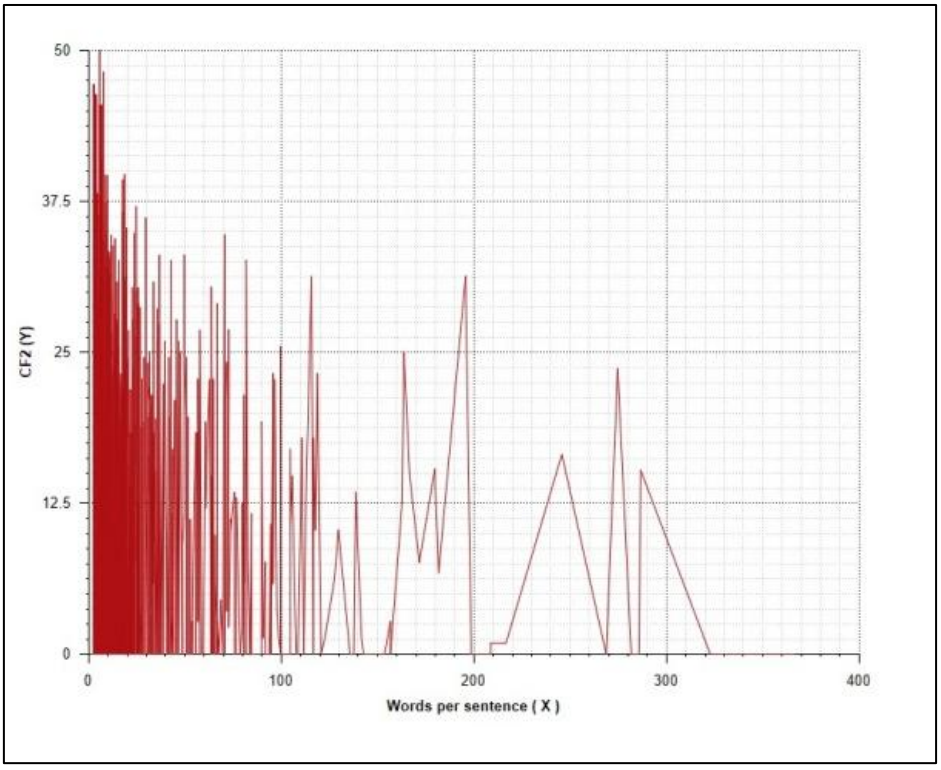

Fig 6.7. ICON Corpus: Graph of Words per Sentence vs. CF2.

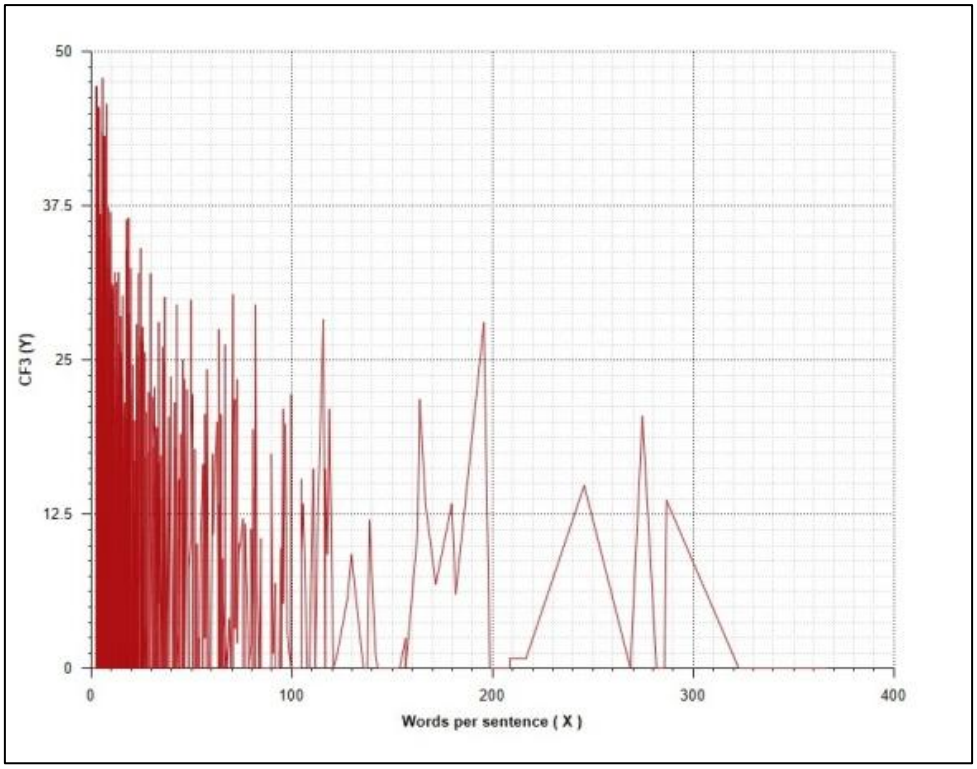

Fig 6.8. ICON Corpus: Graph of Words per Sentence vs. CMI.

The results suggest that the FIRE Corpus was more complex than the ICON Corpus with average value of CF2 and CF3 over the entire corpus being 10.54 and 9.88 respectively. For ICON corpus, CF2 and CF3 are 4.83 and 4.71 respectively which is considerably less than that of FIRE Corpus.

6.7 Conclusion

In this Chapter, we have discussed the need and application of various indexes to represent the complexity of code-mixing in transliterated social media text. We have used various examples – both mathematical and from real-life text – to demonstrate the working of Code Mixing Index. We have highlighted few of the challenges that this index face and have proposed a new index – Complexity Factor (with two variations CF1 and CF2) – which takes into account the relative ordering of words (or the number of language switches) and the number of languages present in addition to the presence of words from non-dominant languages (as done in CMI). Our index provides a more balanced view of the complexity of the text. The index can also be checked for mixed texts from other regions (like Spanish-English, Mandarin-English, etc.).

CHAPTER

7 CONCLUSION AND FUTURE WORK

Chapter 7

Conclusion and Future Work

In this thesis, we have analyzed raw code-mixed text as they occur in various social media platforms like Facebook, Twitter, etc. We have performed a fivefold analysis of the text starting with Language Identification and then sequentially performing Part-of-Speech Tagging, Named Entity Identification, Classification and Linking and Sentiment Analysis. Lastly, we proposed a complexity metric to evaluate the complexity of code-mixed corpora.

7.1 Language Identification

In this Chapter, we presented a brief overview of our system to address the automatic identification of word-level language. We used a sequential classifier CRF for our task of language identification. Our system demonstrated an overall accuracy of 75.5% for token level language identification. The strict F-measure scores for the identification of token level language labels for Bengali, English and Hindi are 0.7486, 0.892 and 0.7972 respectively. The overall weighted F-measure of our system was 0.7498.

While the CRF-based approach was satisfactory, the results could have been improved by including post-processing heuristics for identifying mixed words and named entities. We used character n-grams (n=1 to 5) as one of the features of CRF++[55]. However, the performance of the system declined on incorporating it. In future, we would like to explore some more character level features. Character level features could also be useful in identifying language pairs present in mixed words (containing intra-word code-switching).

Some basic knowledge about other languages and better lexical resources for regional languages should also improve the accuracy of the present system. In our next work, we concentrated specifically on named entities. We would like to make use of Word2Vec and deep learning method for language identification. Recognizing the hidden semantic relationship between words of a language could be beneficial in improving accuracy.

[55] https://taku910.github.io/crfpp/#download

7.2 Part-Of-Speech Tagging

In this task, we have addressed the problem of POS tagging in mixed script social media text. The texts contained two or three languages, with English being one of the three languages. The other languages were Hindi, Bengali and Tamil. We have trained Stanford POS Tagger to build a baseline model. Our final model used Conditional Random Field for part-of-speech tagging. Our results are encouraging and the performance deterioration of Tamil-English mixed text can be attributed to the mismatch of POS-tags.

Currently, there is a lack of quality training data. In the absence of sufficient training data, performance deteriorates using neural network based models or deep learning methods. In future, we would love to explore the effectiveness of Deep learning based features. Word2vec models can also be used to find out words which are semantically similar. We would also like to use of ensemble learning by using various models and combining their results to arrive at the final result. A step in that direction would be to collect more mixed script data from social media and building gold standards using that data. Building an efficient normalization system and disambiguating between similar tags should also improve the accuracy of the system.

In previous Chapter, we have discussed on how to build an automatic language labelling system. Combining our language labelling system with part-of-speech tagger would enable us to process any social media text in real time.

7.3 Named Entity Identification and Linking

In this Chapter, we have described our approach for the #Microposts2016 Named Entity rEcognition and Linking (NEEL) challenge. We have developed a hybrid system using the existing Named Entity Recognizer systems and Twitter-specific Part-of-Speech Taggers in conjunction with the classifier developed by us. The Named Entity Linking was done mainly by using Babelfy, which performs as a multilingual encyclopaedic dictionary and a semantic network.

It should be kept in mind that Named Entities are often considered to be universal as they are not dependent on any language. Therefore our Named Entity Detection and Classifications system is independent of language and can be seamlessly extended to any language that is used in social media.

The performance of our system suffered because of certain restrictions in time. The classification module was slightly biased and the accuracy of classification suffered as result of that. Identifying and selecting better features would have improved our results. A disambiguation module to treat overlapping classes would have been useful. The accuracy of the linking would also improve by taking a semantic similarity approach using synonym sets for the mentions or context word overlapping from the sets while NIL clustering.

7.4 Sentiment Analysis

As per our knowledge, there exists no sentiment classifier for code-mixed social media text. We have performed a machine learning based sentiment classification of Facebook posts. The polarity of each post has been classified as *positive*, *negative* and *neutral*. As there has not been any similar work before, we had to create a dataset of our own. Two human annotators classified the polarity of each post. Due to the inherent complexity of social media text, use of arbitrary emoticons and presence of sarcasm, the agreement between the human annotators was quite low with a Kappa co-efficient of 0.4354. Although the entire dataset consists of 882 posts, we have used only 565 posts where the annotators were unanimous about the polarity of underlying sentiment. We used word-based, semantic and style-based features for classification. The best result was obtained using a combination of word-based and semantic features with an accuracy of 68.5%.

As our dataset is relatively small, we would like to create a larger dataset in future. Sentiment annotation can also be done using distant supervision based on the presence of emoticons. However, such an approach can lead to noisy dataset. Creating a gold standard for all future tasks is a priority for us. In this work, we have not focussed on detection of sarcasm in text. Also, we have not handled negation in data. We would like to concentrate on dealing with these issues in our next work. Apart from that, sentiment classification can be further improved by better handling comparisons and by detecting sentiment targeted towards an entity in particular. Handling of context switches is also important. Developing a real time accurate sentiment classifier model is the ultimate goal which we strive to achieve in future.

7.5 Complexity Metric

In this Chapter, we have discussed the need and application of various indexes to represent the complexity of code-mixing in transliterated social media text. Currently, there exists only one evaluation metric – Code Mixing Index (CMI) – relevant to code-mixed text. In our work, we

have used various examples – both mathematical and from real-life text – to demonstrate the working of CMI. We have highlighted few of the challenges that this index face and have proposed a new index which tackles these issues. Our index (Complexity Factor) takes into account the relative ordering of words (or the number of language switches) and the number of languages present in addition to the presence of words from non-dominant languages (as done in CMI). We have proposed three variations of Complexity Factor – CF1 serves as a baseline or raw index. CF2 and CF3 are more versatile and usable. CF2 uses linear interpolation while CF3 uses geometric functions. Both of these indexes provide a more balanced view of the complexity of the text.

In future, the working of the index can be further explored using a more multilingual text (containing more than two languages). We can also find and compare the complexities of various corpora prior performing tasks like part-of-speech tagging or sentiment analysis. The index can also be checked for mixed texts from other regions (like Spanish-English, Mandarin-English, etc.). In future, another challenging work would be to modify the index to account for complexity caused due to intra-word mixing.

Appendix 1

Tools Used

1 CRF++

We used CRF++[56] for the task of Language Identification and Part-of-speech tagging. CRF++ is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs). It is written in C++ with STL and can be used for segmenting/labelling sequential data. We primarily used it as a classifier in our work. It can be used for several other purposes like Named Entity Recognition, Information Extraction, and Text Chunking etc.

2 Stanford Part-Of-Speech Tagger

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'. Stanford POS tagger[57][58] is a Java implementation of the log-linear part-of-speech taggers. The English tagger uses the Penn Treebank tag set. Like Stanford NER, it also deals with tokens of a sentence. After breaking a sentence into tokens, it has method that assigns each token its parts of speech. The tagger was originally written by Kristina Toutanova. Since that time, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, Michel Galley, and John Bauer have improved its speed, performance, usability, and support for other languages. The system requires Java 1.8+ to be installed. To run a trained tagger, between 60 and 200 MB of memory is required (i.e., in java it requires an option like java -Xms200m). Plenty of memory is needed to train a tagger. It again depends on the complexity of the model but at least 1GB is usually needed, often more (use java option –Xms1024m). The tagger can be retrained on any language, given POS-annotated training text for the language.

[56] https://taku910.github.io/crfpp/
[57] http://nlp.stanford.edu/software/tagger.shtml
[58] http://nlp.stanford.edu/software/stanford-postagger-full-2015-12-09.zip

3 Stanford NER

Stanford NER (also known as CRFClassifier)[59] is a Java implementation of a Named Entity Recognizer. Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. The software provides a general (arbitrary order) implementation of linear chain Conditional Random Field (CRF) sequence models, coupled with well-engineered feature extractors for Named Entity Recognition. The software includes good 3 classes (PERSON, ORGANIZATION, LOCATION) named entity recognizers for English (in versions with and without additional distributional similarity features) and another pair of models trained on the CoNLL 2003 English training data. The distributional similarity features improve performance but the models require considerably more memory. This software also includes recognizer for numerical entities (DATE, TIME, MONEY, and NUMBER). This software[60] processes a sentence and breaks them into tokens and finally it has methods that return the type of named entity that each token is associated with. Moreover, if multiple sentences are given as input, this software first splits them into individual sentences. There are several options to choose different types of Named Entity Recognizer (3 class/7class etc.).

4 ARK Part-Of-Speech Tagger

ARK POS tagger[61] is a fast and robust Java-based tokenizer and part-of-speech tagger for tweets, its training data of manually labeled POS annotated tweets, a web-based annotation tool, and hierarchical word clusters from unlabeled tweets. These were created by Olutobi Owoputi et al. (2013). The Twitter POS model uses a 25-tag tagset which is included with the tagger release and used by default. The recent improvements in the tagger improved accuracy of the results from 90% to 93%. The data and tools are provided as open source to the research community. It enables richer text analysis of Twitter and related social media data sets.

5 Babelfy

Babelfy[62] is a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation based on a loose identification of candidate meanings. It uses a densest

[59] http://nlp.stanford.edu/software/CRF-NER.shtml
[60] http://nlp.stanford.edu/software/CRF-NER.shtml#Download
[61] http://www.cs.cmu.edu/~ark/TweetNLP/
[62] http://babelfy.org/

subgraph heuristic which selects high-coherence semantic interpretations. Babelfy[63] is based on the BabelNet 3.0 multilingual semantic network and jointly performs disambiguation and entity linking in three steps. In the first step, it associates with each vertex of the BabelNet semantic network, i.e., either concept or named entity, a semantic signature, that is, a set of related vertices. This is a preliminary step which needs to be performed only once, independently of the input text. Secondly, it extracts all the linkable fragments from a given input text. For each of the fragments, it lists the possible meanings according to the semantic network. Finally, it creates a graph-based semantic interpretation of the whole text by linking the candidate meanings of the extracted fragments using the previously-computed semantic signatures. It then extracts a dense subgraph of this representation and selects the best candidate meaning for each fragment. The text, written in any of the 271 languages supported by BabelNet 3.0, is output with possibly overlapping semantic annotations.

6 WEKA

Weka[64] is a workbench that contains a collection of visualization tools and algorithms for data analysis and predictive modelling, together with graphical user interfaces for easy access to these functions. Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as one flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining, but there is separate software for converting a collection of linked database tables into a single table that is suitable for processing using Weka. Another important area that is currently not covered by the algorithms included in the Weka distribution is sequence modelling.

[63] http://babelfy.org/download
[64] http://www.cs.waikato.ac.nz/ml/weka/

Appendix 2

Research Publications

---

1. Satanu Ghosh, **Souvick Ghosh** and Dipankar Das. Labeling of Query Words using Conditional Random Field. *In Proceedings of Shared Task on Mixed Script Information Retrieval, FIRE 2015.*

2. **Souvick Ghosh**, Promita Maitra and Dipankar Das. Feature Based Approach to Named Entity Recognition and Linking for Tweets. *In 6th Workshop on Making Sense of Microposts (#Microposts2016), 2016.*

Bibliography

1. Agarwal H., Amni A. (2006). Part of Speech Tagging and Chunking with Conditional Random Fields. *In: NLPAI Machine Learning Competition.*

2. Aichner T. and Jacob F. (March 2015). Measuring the Degree of Corporate Social Media Use. *International Journal of Market Research 57 (2): 257–275.*

3. Al-Onaizan Y. and Knight K. (2002). Named entity translation: Extended abstract. *In HLT, pages 122-124. Singapore, 2002.*

4. Auer J. C. P. (1984). BILINGUAL CONVERSATION. *Amsterdam and Philadelphia: John Benjamins, 1984. Pp. 116.*

5. Auer P. (2013). Code-Switching in Conversation: Language, Interaction and Identity. *Routledge.*

6. Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., and Ives Z. (2007). DBpedia: A Nucleus for a Web of Open Data. *In Proc. of ISWC/ASWC, pages 722–735.*

7. Avinesh P. V. S, Karthik G. (2006). Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning. *In: NLPAI Machine Learning Competition.*

8. Balahur A. (2013). Sentiment Analysis in Social Media Texts. *WASSA 2013, 120.*

9. Barman U., Das A., Wagner J. and Foster J. (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. *In Proceedings of The First Workshop on Computational Approaches to Code Switching, pages 13–23, October 25, 2014, Doha, Qatar.*

10. Beesley K. R. (1988). Language identifier: A computer program for automatic natural-language identification of on-line text. *In Proceedings of the 29th Annual Conference of the American Translators Association, volume 47, page 54.*

11. Bharati A., Chaitanya V., Sangal R. (1995). Natural Language Processing – A Paninian Perspective. *Prentice-Hall India, New Delhi (1995).*

12. Black W. J., Rinaldi F., and Mowatt D. (1998). Facile: Description of the NE system used for muc-7. *In Proceedings of the 7th Message Understanding Conference.*

13. Bohm C., de Melo G., Naumann F., and Weikum G. (2012). LINDA: distributed web-of-data-scale entity matching. *In Proc. of CIKM, pages 2104–2108.*

14. Bontcheva K., Derczynski L., Funk A., Greenwood M. A., Maynard D., and Aswani N. (2013). TwitIE: A Fully-featured Information Extraction Pipeline for Microblog Text. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics.*

15. Brill E. (1992). A simple rule-based part-of-speech tagger. *In Proceedings of the 3rd Conference on Applied NLP. 152-155.*

16. Brill E. (1995a). Transformation-based error-driven learning and Natural Language Processing: A case study in part-of-speech tagging. *Computational Linguistics, 21(4): 543-565.*

17. Brill E. (1995b). Unsupervised learning of disambiguation rules for part of speech tagging. *In Proceedings of 3rd Workshop on Very Large Corpora Workshop, Massachusetts.*

18. Cano A. E., Preotiuc-Pietro D., Radovanovic D., Weller K. and Dadzie A. (2016). #Microposts2016 – 6th Workshop on 'Making Sense of Microposts': Big things come in small packages. *WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada. http://dx.doi.org/10.1145/2872518.2893528.*

19. Cardenas-Claros M. S. and Isharyanti N. (2009). Codeswitching and code-mixing in internet chatting: Between yes, ya, and si a case study. *In The JALT CALL Journal, 5.*

20. Cavnar W. B. and Trenkle J. M. (1994). Ngram-based text categorization. *In Proceedings of the Third Annual Symposium on Document Analysis and Information (SDAIR 94), pages 161–175, Las Vegas, Nevada.*

21. Chinchor N. A. (1998). Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition. *In Proceedings of the Seventh Message Understanding Conference (MUC- 7), page 21 pages, Fairfax, VA, April. version 3.5, http://www.itl.nist.gov/iaui/894.02/related projects/muc/.*

22. Chittaranjan G., Vyas Y., Bali K., Choudhury M. (2014). Word-level Language Identification using CRF: Code-switching Shared Task Report of MSR India System. *In Proceedings of The First Workshop on Computational Approaches to Code Switching, pages 73–79, 2014.*

23. Chowdhury M. S. A., Minhaz Uddin N. M., Imran M., Hassan M.M., and Haque M.E. (2004). Parts of Speech Tagging of Bangla Sentence. *In Proceeding of the 7th International Conference on Computer and Information Technology (ICCIT), Bangladesh, 2004.*

24. Collins M. and Singer Y. (1999). Unsupervised models for named entity classification. *In Proceedings of EMNLP 99.*

25. Cornolti M., Ferragina P., and Ciaramita M. (2013). A framework for benchmarking entity annotation systems. *In Proc. of WWW, pages 249–260.*

26. Cucerzan S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *In Proc. of EMNLP-CoNLL, pages 708–716.*

27. Cutting D., Kupiec J., Pederson J. and Sibun P. (1992). A practical part-of-speech tagger. *In Proceedings of the 3rd Conference on Applied NLP. 133-140.*

28. Dadvar M., Trieschnigg D., Ordelman R., and de Jong F. (2013). Improving cyberbullying detection with user context. *In Advances in Information Retrieval, pages 693-696. Springer, 2013.*

29. Dandapat S. (2007). Part-of-Speech Tagging and Chunking with Maximum Entropy Model. *Workshop on Shallow Parsing for South Asian Languages.*

30. Dandapat S., Sarkar S. (2006). Part-of-Speech Tagging for Bengali with Hidden Markov Model. *In NLPAI ML workshop on Part of speech tagging and Chunking for Indian language.*

31. Das A. and Bandyopadhyay S. (2010). SentiWordNet for Indian Languages. *In the 8th Workshop on Asian Language Resources (ALR), COLING 2010, Pages 56-63, August, Beijing, China.*

32. Das A. and Bandyopadhyay S. (2011). Dr Sentiment Knows Everything! *In ACL/HLT 2011 Demo Session, Pages 50-55, June, Portland, Oregon, USA.*

33. Das A. and Gambäck B. (2012). Sentimantics: The Conceptual Spaces for Lexical Sentiment Polarity Representation with Contextuality. *In the 3rdWorkshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), ACL 2012, Pages 38–46, Jeju, South Korea.*

34. Das S. and Chen M. (2001). Yahoo! for Amazon: Sentiment Parsing from Small Talk on the Web. *In EFA 2001.*

35. Dave K., Lawrence S., and Pennock D. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *In WWW2003: the 13th international conference on World Wide Web.*

36. De Choudhury M., Gamon M., and Counts S. (2012). Happy, Nervous or Surprised? Classification of Human Affective States in Social Media. *ICWSM.*

37. Dermatas E. and George K. (1995). Automatic stochastic tagging of natural language texts. *Computational Linguistics, 21(2): 137-163.*

38. DeRose S. J. (1988). Grammatical category disambiguation by statistical optimization. *Computational Linguistics, 14:31-39.*

39. Dhanalakshmi V., Kumar A., Shivapratap G., Soman K. P., Rajendran S. (2009). Tamil POS Tagging using Linear Programming. *International Journal of Recent Trends in Engineering, 1(2).*

40. Dhanalakshmi V., Kumar M. A., Rajendran S., Soman K. P. (2009). POS Tagger and Chunker for Tamil Language. *Proceedings of Tamil Internet Conference.*

41. Diakopoulos N. and Shamma D. (2010). Characterizing debate performance via aggregated twitter sentiment. *In Proc. CHI 2010. 1195 1198.*

42. Ding X., Liu B. and Yu P. S. (2008). A holistic lexicon-based approach to opinion mining. *In Proceedings of the 2008 International Conference on Web Search and Data Mining, pages 231–240. ACM.*

43. Dunning T. (1994). Statistical identification of language. *Technical report.*

44. Eineborg M. and Gambäck B. (1994). Tagging experiment using neural networks. *In Proceeding of the 9th Nordic Conference of Computational Linguistic, Sweden. 71-81.*

45. Ekbal A., Mandal S. and Bandyopadhyay S. (2007). POS tagging using HMM and rule based chunking. *Workshop on Shallow Parsing for South Asian Languages.*

46. Ekbal A., Naskar S., and Bandyopadhyay S. (2006). A modified joint source channel model for transliteration. *In COLING-ACL, pages 191-198. Australia, 2006.*

47. Elsner M., Charniak E. and Johnson M. (2009). Structured Generative Models for Unsupervised Named-Entity Clustering. *In Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 164–172, Boulder, Colorado, June 2009.*

48. Erbs N., Zesch T., and Gurevych I. (2011). Link discovery: A comprehensive analysis. *In Proc. of ICSC, pages 83–86.*

49. Esuli A. and Sebastiani F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. *Proceedings of LREC (Vol. 6, p. 417 422).*

50. Ferragina P. and Scaiella U. (2010). TAGME:On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). *In Proc. of CIKM, pages 1625–1628.*

51. Finkel J., Grenager T., and Manning C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pages 363–370. Association for Computational Linguistics.*

52. Gambäck B. and Das A. (2014). On measuring the complexity of code-mixing. *In Proceedings of the 11th International Conference on Natural Language Processing, pages 1–7, Goa, India, December. 1st Workshop on Language Technologies for Indian Social Media.*

53. Gella S., Sharma J., and Bali K. (2013). Query word labeling and back transliteration for Indian languages: Shared task system description. *In FIRE Working Notes.*

54. Gimenez J. and Marquez L. (2003). Fast and accurate part-of-speech tagging: The SVM approach revisited. *In Proceedings of RANLP. 158-165.*

55. Gimpel K., Schneider N., O'Connor B., Das D., Mills D., Eisenstein J., Heilman M., Yogatama D., Flanigan J., and Smith N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. *In ACL 11, page 42, 2011.*

56. Go A., Bhayani R., and Huang L. (2009). Twitter sentiment classification using distant supervision. *Technical Report, Stanford.*

57. Gold E. M. (1967). Language identification in the limit. *Information and control, 10(5):447–474.*

58. Goto I., Kato N., Uratani N., and Ehara T. (2003). Transliteration considering context information based on the maximum entropy method. *In MT-Summit IX, pages 125-132. New Orleans, USA, 2003.*

59. Gottron T., Lipka N. (2010). A Comparison of Language Identification Approaches on Short, Query- Style Texts. *Advances in Information Retrieval: 32nd European Conference on IR Research, Proceedings, Springer, Milton Keynes, UK, p. 611-614, March, 2010.*

60. Grefenstette G., Qu Y., Shanahan J., and Evans D. (2004). Coupling niche browsers and affect analysis. *In RIAO'2004.*

61. Gupta K. and Choudhury M. and Bali K. (2012). Mining Hindi-English Transliteration Pairs from Online Hindi Lyrics. *In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC '12), pages 2459-2465, Istanbul, Turkey, 2012.*

62. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations, 11:231, 2009.*

63. Hasan F. M., UzZaman N. and Khan M. (2007). Comparison of Unigram, Bigram, HMM and Brill's POS Tagging Approaches for some South Asian Languages. *In Proc. Conference on Language and Technology (CLT07), Pakistan, August 7 - 11, 2007.*

64. Hasan F., UzZaman N., and Khan M. (2006). Comparison of different POS Tagging Techniques (n-gram, HMM and Brill's tagger) for Bangla. *In Proc. ICS2E 2006.*

65. Hatzivassiloglou V. and McKeown K. R. (1997). Predicting the semantic orientation of adjectives. *In Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics, pages 174–181. Association for Computational Linguistics.*

66. He J., de Rijke M., Sevenster M., van Ommering R., and Qian Y. (2011). Generating links to background knowledge: A case study using narrative radiology reports. *In CIKM '11.*

67. Herring S. (2003). Media and Language Change: Special Issue. *Special issue of the Journal of Historical Pragmatics, 4 (1).*

68. Hidayat T. (2012). An Analysis of Code Switching Used by Facebookers (a Case Study in a Social Network Site). *BA Thesis, English Education Study Program, College of Teaching and Education (STKIP), Bandung, Indonesia, October, 2012.*

69. Hindle D. (1989). Acquiring disambiguation rules from text. *In Proceedings of the 27th annual meeting on Association for Computational Linguistics. Vancouver, British Columbia, Canada. 118-125.*

70. Hoffart J., Seufert S., Ba Nguyen D., Theobald M., and Weikum G. (2012). KORE: keyphrase overlap relatedness for entity disambiguation. *In Proc. of CIKM, pages 545–554.*

71. Hoffart J., Suchanek F. M., Berberich K. and Weikum G. (2013). YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence, 194:28–61.*

72. Hoffart J., Yosef M. A., Bordino I., Furstenau H., Pinkal M., Spaniol M., Taneva B., Thater S., and Weikum G. (2011). Robust disambiguation of named entities in text. *In Proc. of EMNLP, pages 782–792.*

73. Holzman L. and Pottenger W. Classification of emotions in internet chat: An application of machine learning using speech phonemes. *Technical Report LU-CSE-03-002, Lehigh University, 2003.*

74. Hu M. and Liu B. (2004). Mining and summarizing customer reviews. *In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177. ACM.*

75. Hu X. and Liu H. (2012). Text analytics in social media. *Mining Text Data, pages 385–414.*

76. Huang Q., Singh V. K. and Atrey P. K. (2014). Cyber Bullying Detection Using Social and Textual Analysis. *In Proceedings of the 3rd International Workshop on Socially-Aware Multimedia, Pages 3-6. New York, NY, USA, 2014.*

77. Hughes B., Baldwin T., Bird S., Nicholson J., and MacKinlay A. 2006. Reconsidering language identification for written language resources. *In Proc. International Conference on Language Resources and Evaluation, pages 485–488.*

78. Joshi A. K. 1982. Processing of sentences with intra-sentential code-switching. *In J. Horecky, editor, Proceedings of the 9th conference on Computational linguistics – Volume 1 (COLING'82), pages 145–150. Academia Praha, North-Holland Publishing Company.*

79. Jung S. Y., Hong S. L., and Paek E. (2000). An english to korean transliteration model of extended markov window. *In COLING, pages 383-389, 2000.*

80. Kamps J., Marx M., Mokken R., and de Rijke M. (2004). Using WordNet to measure semantic orientations of adjectives. *In Proceedings LREC 2004.*

81. Karlsson F., Voutilainen A., Heikkila J. and Anttila A. (1995). Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. *Mouton de Gruyter, Berlin.*

82. Kilgarriff A. (2001). Comparing corpora. *International Journal of Corpus Linguistics. 6(1):97–133.*

83. King B., Abney S. (2013). Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, Atlanta, Georgia, p. 1110-1119, June, 2013.*

84. Kudo T. (2014). Crf++: Yet another crf toolkit.

85. Lafferty J., McCallum A. and Pereira F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *In Proceedings of the Eighteenth International Conference on Machine Learning. 282-289.*

86. Li D. C. S. (2000). Cantonese-English codeswitching research in Hong Kong: a Y2K review. *World Englishes, 19(3):305–322.*

87. Li H., Min Z., and Su J. (2004). A joint source-channel model for machine transliteration. *In ACL, 2004.*

88. Liu H., Lieberman H., and Selker T. (2003). A model of textual affect sensing using real-world knowledge. *In IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces, pages 125–132, New York, NY, USA, 2003. ACM Press.*

89. Liu X., Zhang S., Wei F., and Zhou M. (2011). Recognizing named entities in tweets. *In ACL: HLT '11, 2011.*

90. Ma Q. and Isahara H. (1998). A multi-neuro tagger using variable lengths of contexts. *In Proceedings of the 17th international conference on Computational linguistics, Montreal, Quebec, Canada. 802-806.*

91. Marrero M., Sanchez-Cuadrado S., Lara J., and Andreadakis G. (2009). Evaluation of Named Entity Extraction Systems. *Advances in Computational Linguistics, Research in Computing Science, 41:47–58.*

92. Mcteer M., Schwartz R. and Weischedel R. (1991). Empirical studies in part-of-speech labeling. *In Proceedings Of the 4th DARPA Workshop on Speech and Natural Language, pp. 331-336.*

93. Meij E., Bron M., Hollink L., Huurnink B., and de Rijke M. (2011). Mapping queries to the Linking Open Data cloud: A case study using DBpedia. *Web Semantics: Science, Services and Agents on the World Wide Web, 9(4):418 -433.*

94. Merialdo B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics, 20(2):155-171.*

95. Mihalcea R. and Csomai A. (2007). Wikify!: linking documents to encyclopedic knowledge. *In Proc. Of CIKM, pages 233–242.*

96. Milne D. and Witten I. H. (2008). Learning to link with Wikipedia. *In CIKM '08, 2008.*

97. Milroy L. and Muysken P., editors. (1995). One speaker, two languages: Cross-disciplinary perspectives on code-switching. *Cambridge University Press.*

98. Mishne G. (2005). Experiments with mood classification in blog posts. *In 1st Workshop on Stylistic Analysis Of Text For Information Access.*

99. Moro A., Cecconi F., and Navigli R. (2014). Multilingual word sense disambiguation and entity linking for everybody. *In 13th International Semantic Web Conference, Posters and Demonstrations (ISWC 2014), page 25, 2014.*

100. Moro A., Raganato A., and Navigli R. (2014). Entity Linking meets Word Sense Disambiguation: A Unified Approach. *Transactions of the Association for Computational Linguistics.*

101. Muysken P. (2001). The study of code-mixing. In Bilingual Speech: A typology of Code-Mixing. *Cambridge University Press. 2001.*

102. Nadeau D. and Sekine S. (2007). A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes, 30(1).*

103. Nakagawa T., Kudoh T. and Matsumoto Y. (2001). Unknown word guessing and part-of-speech tagging using support vector machines. *In Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium. 325-331.*

104. Nakamura M., Maruyama K., Kawabata T., Shikano K. (1990). Neural network approach to word category prediction for English texts. *In Proceedings of the 13th Conference on Computational Linguistics (COLING 90), Helsinki, Finland, 213-218.*

105. Nasukawa T. and Yi J. (2003). Sentiment analysis: capturing favorability using natural language processing. *In Proceedings K-CAP'03: the international conference on Knowledge capture, 2003.*

106. Navigli R. and Ponzetto S. P. (2012a). BabelNet:The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence, 193:217–250.*

107. Nguyen D., Dogruöz A. S. (2013). Word Level Language Identification in Online Multilingual Communication. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, ACL, Seattle, Washington, p. 857-862, October, 2013.*

108. Ortony A., Clore G. L., and Foss M. A. (1987). The referential structure of the affective lexicon. *Cognitive science, 11(3):341–364.*

109. Owoputi O., O'Connor B., Dyer C., Gimpel K., Schneider N., and Smith N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. *In Proceedings of NAACL.*

110. Pak A. and Paroubek P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. *In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta; ELRA, may. European Language Resources Association. 19-21.*

111. Pang B., Lee L., and Vaithyanathan S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *In Proceedings EMNLP 2002.*

112. Paolillo J. C. (2011). Conversational codeswitching on usenet and internet relay chat. *Language@Internet, 8(3).*

113. Poplack S. (1980). Sometimes i'll start a sentence in Spanish y termino en espanol: Toward a typology of code-switching. *Linguistics, 18:581–618.*

114. Poutsma A. (2002). Applying monte carlo techniques to language identification. *Language and Computers, 45(1):179–189.*

115. Rao D., McNamee P., and Dredze M. (2013). Entity Linking: Finding Extracted Entities in a Knowledge Base. *In Multi-source, Multilingual Information Extraction and Summarization, Theory and Applications of Natural Language Processing, pages 93–115. Springer Berlin Heidelberg.*

116. Read J. (2004). Recognising affect in text using pointwise-mutual information. *Master's thesis, University of Sussex.*

117. Read J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. *In Proceedings of ACL-05, 43nd Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.*

118. Ritter A., Clark S., Mausam, and Etzioni O. (2011). Named entity recognition in tweets: An experimental study. *In Proc. of Empirical Methods for Natural Language Processing (EMNLP), Edinburgh, UK.*

119. Rizzo G. and Van Erp M. (2016). Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge. *In 6th Workshop on Making Sense of Microposts (#Microposts2016).*

120. Roberts A., Gaizauskas R., Hepple M., and Guo Y. 2008. Combining terminology resources and statistical methods for entity recognition: an evaluation. *Proceedings of the Conference on Language Resources and Evaluation (LRE'08).*

121. Rosner M., Farrugia P. J. (2007). A Tagging Algorithm for Mixed Language Identification in a Noisy Domain. *In Proceedings of the 8th Annual INTERSPEECH Conference, vol. 3, ISCA, Antwerp, Belgium, p. 1941-1944, August, 2007.*

122. Rubin V., Stanton J., and Liddy E. (2004). Discerning emotions in texts. *In The AAAI Symposium on Exploring Attitude and Affect in Text (AAAI-EAAT), 2004.*

123. S. Wang and Manning C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 90–94. Association for Computational Linguistics.*

124. Samuelsson C., Voutilainen A. (1997). Comparing a linguistic and a stochastic tagger. *In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics (EACL), Madrid, Spain. 246-253.*

125. San H. K. (2009). Chinese-English code-switching in blogs by Macao young people. *Master's thesis, The University of Edinburgh, Edinburgh, UK. http://hdl.handle.net/1842/3626.*

126. Sang E. F. T. K. and De Meulder F. (2003). Introduction to the CoNLL-2003 shared task: Language independent named entity recognition. *In Walter Daelemans and Miles Osborne, editors, Proceedings of CoNLL-2003, pages 142–147. Edmonton, Canada.*

127. Schütze H. (1993). Part-of-speech induction from scratch. *In Proceedings of the 31st annual meeting on Association for Computational Linguistics, Columbus, Ohio. 251-258.*

128. Seddiqui M. H., Rana A. K. M. S., Al Mahmud A. and Sayeed T. (2003). Parts of Speech Tagging Using Morphological Analysis in Bangla. *In Proceeding of the 6th International Conference on Computer and Information Technology (ICCIT), Bangladesh, 2003.*

129. Selvam M., Natarajan A. M. (2009). Improvement of Rule Based Morphological Analysis and POS Tagging in Tamil Language via Projection and Induction Techniques. *International Journal of Computers, 3(4).*

130. Sha F. and Pereira F. (2003). Shallow parsing with conditional random fields. *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada.134-141.*

131. Shrivastav M., Melz R., Singh S., Gupta K. and Bhattacharyya P. (2006). Conditional Random Field Based POS Tagger for Hindi. *In Proceedings of the MSPIL, Bombay, 63-68.*

132. Shrivastava M., Bhattacharyya P. (2008). Hindi POS Tagger Using Naive Stemming: Harnessing Morphological Information Without Extensive Linguistic Knowledge. *In International Conference on NLP (ICON08), Macmillan Press, New Delhi.*

133. Singh S., Gupta K., Shrivastava M., Bhattacharya P. (2006). Morphological Richness Offsets Resource Demand- Experiences in Constructing a POS Tagger for Hindi. *In COLING/ACL, pp. 779-786.*

134. Solorio T., Blair E., Maharjan S., Bethard S., Diab M., Gohneim M., Hawwari A., AlGhamdi F., Hirschberg J., Chang A., Fung P. (2014). Overview for the First Shared Task on Language Identification in Code-Switched Data. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, ACL, Doha, Qatar, p. 62-72, October, 2014. 1st Workshop on Computational Approaches to Code Switching.*

135. Solorio T., Liu Y. (2008). Learning to Predict Code-Switching Points. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, ACL, Honolulu, Hawaii, p. 973-981, October, 2008a.*

136. Solorio T., Liu Y. (2008). Part-of-Speech Tagging for English-Spanish Code-Switched Text. *In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, ACL, Honolulu, Hawaii, p. 1051-1060, October, 2008b.*

137. Sowmya V., Choudhury M., Bali K., Dasgupta T., and Basu A. (2010). Resource creation for training and testing of transliteration systems for Indian languages. *In LREC, 2010.*

138. *Surana H. and Singh A. K. (2008). A more discerning and adaptable multilingual transliteration mechanism for Indian languages. In COLING-ACL, pages 64-71. India, 2008.*

139. Tan S., Wang Y., and Cheng X. (2008). Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. *In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 743–744, New York, NY, USA. ACM. ISBN 978-1-60558-164-4. URL http: //doi.acm.org/10.1145/1390334.1390481.*

140. Toutanova K., Klein D., Manning C. D. and Singer Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1(pp. 173-180). Association for Computational Linguistics.*

141. Tukey J. W. (1962). The Future of Data Analysis. *Ann. Math. Statist. 33 (1962), no. 1, 1—67.*

142. Turney P. and Littman M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS), 21(4), 2003.*

143. Turney P. D. and Littman M. L. (2002). Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *Technical Report EGB-1094, National Research Council Canada.*

144. Vyas Y. and Gella S. and Sharma J. and Bali K. and Choudhury M. (2014). POS Tagging of English-Hindi Code-Mixed Social Media Content. *In Proceedings of the First Workshop on Computational Approaches to Code Switching, EMNLP.*

145. Wiebe J. (2000). Learning subjective adjectives from corpora. *In Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence. AAAI Press / The MIT Press, 2000.*

146. Wilson T., Wiebe J., and Hoffmann P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *In Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Vancouver, CA, 2005.*

147. Xia F., Lewis W., and Poon H. (2009). Language ID in the context of harvesting language data off the web. *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pages 870–878, Athens, Greece, March. Association for Computational Linguistics.*

148. Yamaguchi H., Tanaka-Ishii K. (2012). Text segmentation by language using minimum description length. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, vol. 1, ACL, Jeju, Korea, p. 969-978, July, 2012.*

149. Zhang L., Ghosh R., Dekhil M., Hsu M., and Liu B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011, 89.*