Towards Understanding Code-Mixed Telugu-English Data

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS in Computational Linguistics
by
Research

by

Divya Sai Jitta
201225167
jittadivya.sai@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2018

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Towards Understanding Code-Mixed Telugu-English Data" by Divya Sai Jitta, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Radhika Mamidi

Dedicated to my beloved Amma and Nanna.

Acknowledgments

Firstly, I would want to extend my heartfelt thanks to Dr.Radhika Mamidi. This thesis would not have been possible without her consistent guidance and moral support. She has been the motivating force throughout my research period. She has never failed to lift my spirits when I felt under motivated during the days of failure and hopelessness. She strongly motivated me towards taking up research in resource poor Indian languages. Though I have struggled in the initial days with collection and annotation of data, I take pride in myself today for completing my research successfully. She has been a great guide and also taught me how to linguistically analyze the computational results of experiments.

Furthermore, I am also grateful to Professor Dipti Mishra Sharma and Dr. Manish Shrivastava for their dedication and passion that they displayed while helping a student to understand a problem more deeply and scientifically. Without any fail, the lab environment at LTRC is a great opportunity to discuss research related problems and share knowledge continuously. This not only has helped me in doing good research, but also helped me in developing good presentation skills.

As a dual degree student, I have experienced both the competitiveness in programming and systematic research that IIIT Hyderabad is mainly popular for. I have learned a lot from both these tracks. I also learned that techniques of one domain when adopted to another with careful understanding and modeling of parameters can yield better results. I would also like to thank Ms. Kovida for guiding me during the earlier days of research. She has taught me how to do smart research.

I would like to conclude by thanking my parents and sister for giving me such a wonderful opportunity to study at IIIT Hyderabad. Without their constant support and encouragement I would not have been the same person that I am today. I consider completion of MS research and thesis as a milestone in my life and I would like to remember it forever by dedicating this thesis to my beloved parents.

Abstract

Code-Mixing (CM), a progeny of multilingualism, is defined as a phenomenon where linguistic units such as phrases, words and morphemes of one language are embedded within an utterance of another language. This phenomenon is often observed in conversations within the bilingual and multilingual user group. Also, Code-Mixed language is extensively used on social media sites like Facebook and Twitter. Though code-mixing is the most natural form of conversation both in speech and text (online chats), the current dialog systems and search engines are not capable of handling this kind of social interaction. Many Popular virtual personal assistants used by a significant amount of smart phone users are unable to handle the mixing and switching between any two languages, which occurs very naturally to any bilingual or multilingual person. So, as mentioned above, it becomes extremely important to understand CM for both information extraction and for the purpose of building dialog systems that are capable of social interaction. In this thesis we take the first step towards understanding CM between two languages - Telugu and English, belonging to two different language families, sharing no ancestry at all. We start by building basic preprocessing tools like Language Identification (LID) models, POS taggers for CM data. We collected Telugu-English code-mixed social-media blog based data. The collected data was annotated by two individual annotators and a Kappa score of 88.91 is reported, which is a sign of reliable data. We experimented with various machine learning algorithms and finally we present a Multiple Layer Perceptron (Neural networks) LID model that uses character n-gram vectors augmented with other handcrafted features. Our system gives an F1-score of 97 for English and 96 for Telugu. The LID module has been used for the task of POS tagging of Telugu-English CM data, which produced an accuracy of 52.37%.

In the second part of the thesis we made an attempt towards understanding code-mixing in dialog (richest and most natural form of language) through automatic recognition of dialog act (speaker's intention) of an utterance. We have experimented with learning algorithms like Support Vector Machines, Naive Bayes, Kth Nearest Neighbor and Hidden Markov Models. Our best system that gives an F1-score of 72.30 is HMM based. Non-availability of annotated conversational code-mixed data poses problems and hinders us from adopting data-driven methods like Neural networks. Manual procurement and annotation of code-mixed data is not only time-consuming but also is labor intensive. Therefore, we also investigate on how knowledge extracted from resource rich languages could be useful in dialog act recognition of code-mixed conversations. We show that a decent accuracy of 66% can be obtained by transforming the code-mixed utterances into a sequence of English words by the use of modules like

LID, transliteration and translation. We propose that in the future, a variant of transfer learning can be used to incorporate this knowledge along with the knowledge obtained from manually annotated code-mixed conversations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the primary things that sets apart human beings from other species is Language. Language has evolved through centuries and will continue to evolve. It is hard to think of ourselves, our societies, and culture without language. Language is a primary means of socialization. It interacts with the social, political and economic power structures. Language and society mutually influence each other. ”Code-Mixing” (CM) is an outcome of this mutual interaction. ”Multilingualism” is one of the factors that claims to be the origin of code-mixing. Multilingualism [1] is the use of more than one language, either by an individual speaker or by a community of speakers. Multilingualism basically arises due to a need to communicate across speech communities [35]. Multilingualism grew due to globalization and wider cultural contact but has been prevalent in the ancient times as well. With the emergence of Internet and Social Media, there is a lot of data available on the web and Information Extraction and Retrieval methods became pivotal. It is observed that a large population on social media - Facebook and Twitter, users code-mix very often [26]. The code-mixed languages manifest themselves on social media adopting the style of social media language(SMS language). The problem is further compounded with crossing the script barriers in code-mixing. There are several other terms that come into the picture while talking about CM, among them ”Code-Switching” and ”Lexical Borrowing” are the ones that appear quite very often. The differences between these terms are not very clear. Some scholars use these terms interchangeably, whereas other choose to differentiate [12][11]. In this thesis, we do not make a differentiation and computationally they are all treated alike, and we use the umbrella term through out this thesis.

In India, CM is widely observed between English and an Indian Language. In this thesis, we focus our study on one such language pair - Telugu-English. Telugu is an agglutinative language which belongs to Dravidian language family. A significant amount of population whose native tongue is Telugu use English in their everyday conversations. Most informal and semi-formal conversations are fabricated in CM fashion. It usually occurs in scenarios where the formal education is received in a language other than the persons native tongue. As most of the formal education is in English - an effect of British colonisation, Telugu-English code-mixing is quite evident from day to day chats to social media posts.

[1] http://shodhganga.inflibnet.ac.in/bitstream/10603/11248/9/09_chapter%202.pdf

A significant amount of research has been done on other language pairs, but we are the first to collect, understand, annotate and make an attempt towards creating tools for English-Telugu language pair. Social media is a very powerful forum where people express their opinions, make statements about the current state of affairs, and a lot of these posts are code-mixed. Therefore, it is very important to be able to understand CM data computationally

Some other rich sources of code-mixed data are every day conversations or textual chats. In the second part of this thesis, we try to understand conversational code-mixed data with the use of some traditional dialog act recognition techniques. We also investigate on how the existing tools and resources can be wisely used to tag utterances with dialog acts, circumventing the ordeal of collecting and annotating a humongous code-mixed conversational data.

1.1 Terminologies and Definitions

Multilingualism: It is a state regarding an individual speaker who uses more than one lanaguage, or a community where two or more languages are used.[2]

Code-Mixing: According to [32] Code-mixing is embedding of linguistic units such as phrases, words and morphemes of one language into an utterance of another language.

Nonce borrowing: Borrowings that do not conditionally follow any phonological or morphsyntactic constraints on their use in the host language[36].

Research in the past states that, the boundaries between borrowing and word level code-mixing are not very clear. Linguists agree that by repeated use and adoption of morphosynctactic of the language on receiving end can shift the status of a word from foreign to loan.

> An example of this is: ”Rail”(from railways) is an English word and ”Railu Bandi” is a telugu word which means train. In this example, ”Railu” is no longer a foreign word.

In this work, we do not differentiate between these instances and phenomena. All of them are considered under the umbrella phenomenon of ”Code-Mixing”.

Though the distribution of language usage in code-mixing is asymmetrical, Code-Mixing is not a random phenomenon. Though there are no strict rules some observations can be made- **Matrix Language** : The dominant language into which certain words are mixed. The grammatical structure of the code-mixed sentence will be that of the Matrix language. **Embedded Language**: The language, whose lexical items are brought into the Matrix language is called the Embedded Language. In the example below, Telugu is the matrix language and English is the embedded language.

> Example : ”Engineering lo em chadivina chivaraku avvalsindi Software engineeree.” [3]
> Translation: Irrespective of whatever we study in Engeneering, one has to become a Software Engineer towards the end.

[2] http://shodhganga.inflibnet.ac.in/bitstream/10603/11248/9/09_chapter%202.pdf
[3] http://chaibisket.com/just-mechanical-things/

1.2 Generic Characteristic of Code-Mixed Telugu-English Data

Let us talk about the characteristics of CM Telugu-English by looking at an example first.

Example: CM Telugu-English Sentence: ***Prathi person lo positivelu negativelu untayi!***
Translation: There are positives and negatives in every person.

As can be seen mixing happens at morphological level, word level and phrase level too. Following are a few more examples:

1. **Morphological level:** The words that are borrowed from English inflect Telugu suffixes that marks case or number. In ”cinemalu” cinema is the root borrowed from English and ”lu” is Telugu morpheme that marks plurality. Similarly, ”Bus” becomes ”Bussu”, this is nativization.

2. **Word level:** A complete word from English gets assimilated into Telugu. An example: ***”Ilanti oka branch unda*** which means ”Is there a branch like this ?”. Here, ”branch” is an English word, and here the sense is not that of a ”tree branch. Branch refers to a department here. This becomes a domain specific word.

3. **Phrase Level:** ***Aina friendship beauty antene oka person ni complete ga accept cheyyatam!***
Translation : The beauty of friendship lies in completely accepting a person. This is a completely code-mixed sentence, that follows the structure of Telugu with English words embedded in it.

4. **Synatactic level:** Here we discuss about Inter-sentential and Intra sentential mixing. All the examples above are instances of intra-sentential mixing. In Telugu-English CM data, there are occurrences where inter-sentential mixing takes place. One such example is, ***”Asal mohatam lekunda adigestham, we will try as hard as we can to eat, but inka entha thinna valla kaakapothe, just give up”***

1.3 Code-Mixing in Social Media Posts Vs Code-Mixing in Dialog

1. Following in example of Hindi-English code-mixed twitter post:

   ”was it a modi wave or just wave? #aapsweep #aapkidilli **”bjp ki halat congress se jyada achchi ni lag rahi”**.

   Sometimes Twitter users give information in one language and prefer their native tongue to express personal opinions.

2. Figures 1.1 and 1.2 are examples of Telugu-English code-mixing on twitter.

3. Table 1.1 shows an example of a Telugu-English Code-mixed Conversation is in table

Figure 1.1 Twitter Post1

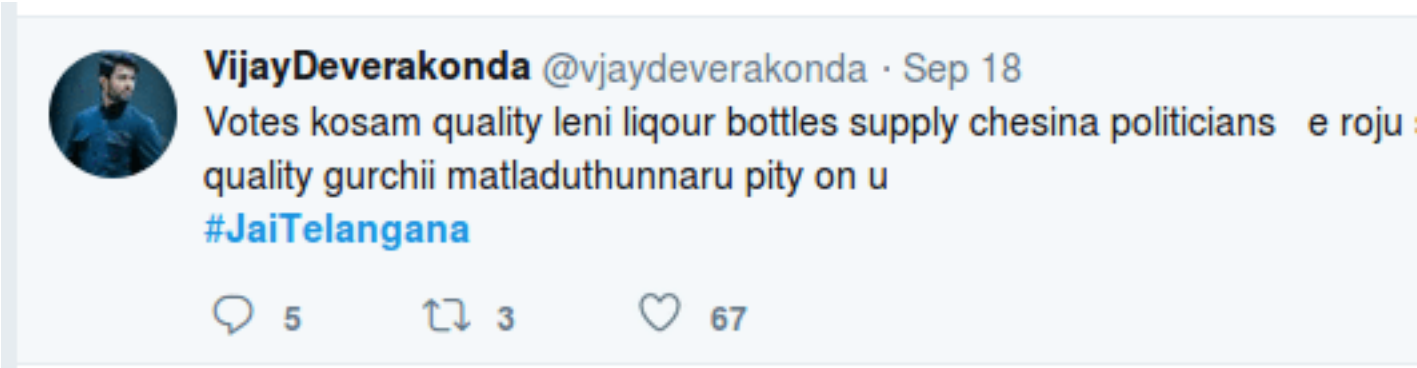

Figure 1.2 Twitter Post2

1.4 Contribution of this thesis

The major contributions of this thesis are:

1. Collecting and Annotating of Code-mixed Telugu-English blog based corpus with language classes.
2. Development of a Neural Network (MLP) based model for the purpose of language identification in code-mixed data.
3. Investigating on how language identification is helpful for the task of tagging code-mixed sentences with Part of Speech tags.
4. Studying Code-Mixing in Dialog:
    - Creation of code-mixed conversational corpus for Library domain using an established research technique called Wizard of Oz.
    - Development of a Dialog Act annotation scheme for the WOz conversations using DAMSL[3] annotation guidelines.

Speaker	Utterance	Translation
A	Hi I am Asklib ela unnavu? nenu mee library assistant ni.	Hi I am Asklib How are you I am your library assistant.
B	Hi I am Divya. I am doing great.	Hi I am Divya. I am doing great.
A	natho intaka mundu matladara leda ide first time aa ?	Did you speak to be earlier or is this the first time?
B	idi second time.	This is the second time.
A	evaina questions unte adagandi.	Ask me if you have any questions.
B	So nenu comedy genre ki related novels kosam search chestunnanu.	I am searching for some novels that belong to comedy genre.
A	okay. meeku ye language lo books kavali ? Hindi, English or Telugu?	okay. In which language do you need these books in?
B	Naku English books Kaavali.	I want books in English.
A	okay.	okay.

Table 1.1 Example of Telugu-English CM Dialog

- Experiments on the WOz conversations using supervised and non-parametric learning methods to recognize dialog act of a code-mixed utterance.
- Investigating how a dialog act annotated corpus from a resource rich language can be adapted to tag code-mixed Telugu-English conversations.

There is a lot of CM data available on the Internet in the form of sentences but lack of CM conversational data posed a major hurdle to our problem.

1.5 Thesis Organization

1. This thesis is organized into 5 chapters. We begin this chapter by throwing some light on the evolution of language and how language and society influence each other. We discuss some important terminologies. In the latter section, we provide example of mixing of English into Telugu at multiple linguistics levels, that marks the primary characteristics of Telugu-English CM language. Then there is an illustration of code-mixing in social media data and Dialog.

2. In chapter 2, background study and work related to code-mixing is reported from both computational and socio-linguistic perspective. We also discuss how these works are relevant to this thesis.

3. We introduce chapter 3 by explaining why language identification is important in a code-mixed scenario and also mention and discuss some research related to language identification in the

past. The later sections of the chapter describe the data collection and annotation procedures and compare and contrast the datasets used. The experiments discuss the features used, validation techniques, learning algorithms and finally the results. The last section of this chapter is concluded by error analysis. A small section of this chapter also discusses the effects of language identification on POS tagging.

4. In chapter 4, we make an attempt towards understanding code-mixing in dialog using dialog act recognition techniques. The chapter begins by giving a brief outline of what dialog acts are and which part of a dialog system uses them and why it is important to recognize a dialog act in a conversation. In this process, we try to explore if code-mixing introduces any new challenges for the task of dialog act recognition. The final section of this chapter concludes with an investigation on how resource rich language like English can be used and adapted for tagging code-mixed utterances with dialog acts.

5. The final chapter presents the conclusions and contributions of this thesis along with paving a way for future work in this area. Also, we discuss some applications that can be developed as a product of understanding code-mixed data.

Chapter 2

Related Work and Background Study

2.1 Multilingualism in India

India represents six distinct language families spread over a large region and spoken by more than one billion speakers. Linguistic Area is marked by the ”convergence”of linguistic features of various languages spoken in a particular region regardless of the fact that these languages may belong to different families[1]. Another way of describing linguistics area is that , it is a geographical area where genetically unrelated languages share common linguistic features. For example, India is a classic example of linguistic area as the languages of the mainland India belonging to four different language families i.e. Indo Aryan, Dravidian, Austro-Asiatic and Tibeto-Burman share several linguistic traits among themselves. Multilingualism in India was largely a product of close contact between these four language families from the earliest recorded history. In India, every individual is either bilingual or multilingual. This can be attributed to two factors. The first is ”medium of education” and a huge number of books related to science, technology and commerce are in English. Secondly, the need of interaction through trade and travel has demanded a need to know more than one language,

2.2 Nativization of English in India and its effect on Multilingualism

Nativization of language is defined as re-defining a language in one's own cultural and linguistic framework. In this process new words and meanings accumulate to suit the social and cultural requirements[1]. English has been nativized in grammar, semantics and pragmatics acquiring the features of Indian languages. This is well documented in socio-linguistic literature. [5] English remained in India, after its native speakers left the country. It was not transmitted across generations through social interaction but happened through formal education, in this respect, it is like Sanskrit (exclusively used for imparting knowledge).

[1] http://shodhganga.inflibnet.ac.in/bitstream/10603/60793/7/07_chapter%202.pdf

Consolidation of English in the domains of power made a significant difference to the multilingual network [20]. Unlike Sanskrit and Persian, English was not in control of any native elite (based on caste or religion). Nativization of English manifests itself in the form of lexicon, grammar and discourse structure and also as intonation and stress in pronunciation.

2.3 Code-Mixing - A Socio-Linguistic perspective

Over the past few decades socio-linguists have been taking interest in the phenomenon called "Code-Mixing". Code-Mixing and Code-Switching are thoroughly studied socio-linguistic phenomenon of multilingual speech communities. [25].

Conversations, both in the real world and virtual world - social-media and other online forums tend to have a lot of mixing and switching between language. This choice of language is largely influenced by the speakers and their communicative goals [17]. When there are multiple channels of information exchange and communication, the choice of channel depends upon a variety of factors that are usually non deterministic [6]. [7] suggests that in some communities language alteration is also used as a device to signal and imply some pragmatic functions. There are also other reason why mixing or switching of language (code) can take place. Some of them are, marking a topic shift, emphasize on a particular aspect, and many a times it is also used in puns [30]

[47] is one of the earliest works that looked at English and Arabic language use in email communication and claimed that English was more frequently used both when searching the Internet and in formal email communications. They also found that romanized Arab script is used instead of classic one in these chats.

Previous studies on the reasons for facebookers to switch language is 45% due to real lexical need, which resulted in 58.97% of inter-sentential switching and 33.33% of intra-sentential switching [26]. [2], this work deals with English and Romanized Hindi tweets from multilingual Indian users, in here, they show that there is a strong preference for swearing in the dominant language.

[8] presents an annotation scheme for annotating the pragmatic functions of Code-switching in Hindi-English (Hi-En) code-switched tweets based on a linguistic analysis and some initial experiments.

[39] studied the validity and universality of three linguistic constraints "the equivalence of structure", "the free morpheme", and "the size-of-constituent by examining some instances of code switching between the syntactically divergent languages Moroccan Arabic and French.

2.4 Code-Mixing - A Computational Perspective

[42] discusses Part-of-Speech (POS) tagging of Hindi-English Code-Mixed(CM) text from social media content. They propose extensions to the existing approaches, and also present a new feature set which addresses the transliteration problem inherent to social media. They achieve an 84% accuracy

with the new feature set. We show that the context and joint modeling of language detection and POS tag layers do not help in POS tagging.

There are a few latest advancements that happened in code-mixing like development of POS taggers [27], [43] has built a shallow parser pipeline for Hindi-English CM social media data. [37] has developed a Question classification system for Code-Mixed Hindi-English Questions.

[24] used Multilayer Perceptron model to determine the polarity of the sentiment code-mixed Facebook posts. [10] has developed a neural network based entity extraction in Hindi-English code-mixed text. The system generates an F-score of 68.24 and uses distributional word representations as features.

There is not much work done in understanding and analyzing Telugu-English code-mixed data. To the best of our knowledge, there is no work that discusses code-mixing in a dialog scenario, where recognizing the intention of a speaker is of primary importance.

Chapter 3

Language Identification

3.1 Why Language Identification (LID) ?

Language identification is a primary tool that is required to understand code-mixing. The input text is first passed through an *NLTK*[1] tokenizer. The tokenized text is then fed into the language identification module. The language identifier assigns a tag from the predefined tag set to each of the tokens. The output of an LID throws some light at the juncture or types of words where mixing is most likely to happen.

3.2 Literature Survey

According to [31] language identification is a solved problem, but the phenomenon of code-mixing that has spread over the social-media in multilingual societies has leveraged a need for better language identification modules for social-media data.

Among one of the many works done in the area of social media, [15] tried to understand the dominant language in a tweet. Apart from collecting tweets from different European languages, he has also analyzed multilingual blogs.

Furthermore [33] has done word level language identification for Turkish-Dutch online posts from a chat forum. They showed that language models when augmented with context produced better results than lexicons.

[23] shows that word level language identification is most likely to confuse between languages which are linguistically related (e.g., Hindi and Gujarati, Czech and Slovak), for which special disambiguation techniques might be required.

[16] has developed a CRF based word-level language identification system for four pairs of languages namely, English-Spanish (En-Es), English-Nepali (En-Ne), English-Mandarin (En-Cn), and Standard

[1] http://www.nltk.org/

Arabic-Arabic (Ar-Ar) dialects. Highest accuracy of 95.3% has been obtained for English-Nepali (En-Ne) language pair.

[9] used a feed forward neural network for the purpose of language identification. The reported accuracy of the system for Telugu-English is 92.27% .

3.3 Data Collection and Description

Language Identification experiments have been done on two data sets. Following is the description of the two datasets.

3.3.1 Dataset A

The data is annotated with language labels and Part of Speech (POS) tags for 10,207 words and is shared for NLP Tools Contests: POS Tagging Code-Mixed Indian Social Media Text at ICON, 2015. Every word is tagged with one of the labels: T, E, M, N and R.

T class has words which belong to Telugu language.
E class has words which belong to English language.
N class has named entities.
M class has words with English roots and Telugu morphological inflections.
R class has words that do not fall into any of the above classes.

Examples: a. cinemalu (”cinema” + ”lu”(Telugu bound morpheme that marks plurality))
b. cinemallo (”cinema” + ”lu”(Telugu bound morpheme that marks plurality) + ”lo”(Telugu free morpheme which means ”inside”))

Another CM dataset was released at ICON, 2016 for NLP Tools Contest (POS tagging). The released data consists of Facebook and Twitter data for three language pairs namely, Hindi-English, Telugu-English and Bengali English.

3.3.1.1 Data Annotation

The Data shared for NLP Tools Contests: POS Tagging Code-Mixed Indian Social Media Text at ICON, 2015 has 10,207 words and is annotated with the mentioned classes under subsection Dataset A. This data is annotated by two individual annotators at four levels: which are

1. Language of the word,
2. Correct form of the word i.e the citation form of the word as available in the dictionary,
3. The part-of-speech tag of the word,

4. Chunk information of the word.

Table 3.1 provides the statistics of the data collected. An example sentence is shown in Table 3.2. A CRF model has been developed on this, which gave an accuracy of 93%. This work is a part of our publication titled ”Part of Speech Tagging for Code-Mixed English-Telugu Social Media Data”, accepted at CICLING 2016.

#words	10207
#sentences	1335
#words in class T	4342
#words in class E	4515
#words in class R	379
#words in class M	81
#words in class N	890
Average length of a sentence	8

Table 3.1 Statistics of the annotated corpus

Token	Class	Correct Spelling	POS	Chunk
plz	E	please	ADV	B-VP
watch	E	watch	VERB	I-VP
it	E	it	PRON	B-NP
nd	E	and	CONJ	B-CP
share	E	share	VERB	B-VP
chusaka	T	cUsAka	VERB	B-VP
neenu	T	nenu	PRON	B-NP
cheppanavasaram	T	ceVppanavarasaraM	NOUN	B-NP
le	T	lexu	VERB	B-VP
meere	T	mIre	PRON	B-NP
share	E	share	NOUN	B-NP
cheestaru	T	ceswAru	VERB	B-VP

Table 3.2 Five level annotation from ICON, 2015 corpus.

The already annotated CM dataset (Facebook and Twitter) released for ICON, 2016 Tools uses the same classes that were used to annotate CM ICON, 2015 data, but the names of the tags are different. The mapping is E::en, T::te, N::ne, R::univ and there is no mixed class M. Table 3.3 gives the Facebook and Twitter data statistics used in the experiments towards modeling a language identifier for CM data. The ICON2016 CM dataset introduces a new class ”acronyms” (acro) and the class universals (univ) takes care of the punctuations, URLS etc.

Class	Facebook Data	Twitter Data
#English words	3,737	3,205
#Telugu words	2,647	4,053
#Universals	3,222	4,475
#Named entities	392	256
#Acronyms	39	24

Table 3.3 Statistics of ICON2016 Facebook and Twitter Data

3.3.2 Dataset B

ChaiBisket (CB) is a a channel (run by a group of Telugu Speaking youth), where there are articles about a plethora of topics (related to human relations and day to day experiences). This site has both videos and articles. Interestingly, most of their articles are Code-Mixed (Telugu-English). As a first step, a list of code-mixed articles have been chosen manually, then the content on their URLS was crawled using "Beautiful Soup" - Python package for parsing HTML and XML documents [2]. The crawled data was then annotated with six categories, namely English (en), Telugu (te) , Mixed (M - morpheme level mixing), acronyms (acro), named-entities (ne), universals (univ) and unknown words (X). Table 3.4 gives the statistics of the ChaiBisket data used to learn language identification model.

3.3.2.1 Dataset Annotation

Initially, the code-mixed data crawled from ChaiBisket was automatically annotated by running the CRF model which was trained on the ICON, 2015 data, that has an accuracy of 93% and then wrongly tagged tokens were annotated with the right classes by two human annotators. Table 3.5 shows some example from CM data crawled from ChaiBisket website.

All tokens	28594
English tokens	11950
Telugu tokens	12100
Universals(numbers,punctuation)	3788
Named entities	591
Acronyms	80
Mixed	41
Other	43

Table 3.4 Statistics of ChaiBisket Data

[2]https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Class	example
English	procedure_en, relatives_en, comments_en
Telugu	bathuku_te, anipinchesthaaru_te
Universals	?_univ, 3_univ
Named entities	Bookmyshow_ne, English_ne
Acronyms	TV_acro, BTech_acro
Mixed	cinemaalani_M, scenelu_M, directga_M
Other	Woh_X, hai_X, zindagi_X

Table 3.5 Class wise examples from Chaibisket Data

3.3.3 Similarities and Differences between datasets A and B

Dataset B follows an article writing style. Majority of the data is in the form of posts. Unlike social media, these posts (articles) are managed by the owners of the website ChaiBisket prior to posting on the website. All these posts are written using an English alphabet keyboard, therefore, all the Telugu words are romanized. Following is an example from ChaiBisket under the title ”9 Annoying Things Godavari People Are Tired Of Hearing!”

> ***Maa Slang Meeku elaga Comedy ga anipistundo mee slang kuda maku alage comedy ga anpistundi. Inta chinna logic ela miss ayyava ra baji rao***.

Whereas, Dataset A is more closer to the language used on social media. Unlike B, A is rich with popular Internet slang like ”LOL”, ”ROFL”, ”FYI”, etc. Apart from these acronyms, B also possesses characteristics of SMS language like usage of ”plz” instead of ”please” and ”sry” instead of ”sorry”. Spelling errors is another important feature of data scraped from Facebook public pages. Following is an example from a Facebook public page post”

> ***plz watch it nd share chusaka nenu cheppanavasarm le meere share chestharuuu***

3.4 Experiments

Two sets of experiments were performed. In the first one, samples from dataset B were used for both training and testing. Whereas, in the second set of experiments, dataset A was used as the training data and tested on B and then vice-versa. Four supervised learning algorithms were used through the experiments. The subsequent sections describe the learning algorithms used.

3.4.1 SET 1 Experiments and Results

Four learning algorithms were chosen to perform the experiments. The reasons for choosing these out of a variety of learning algorithms known are discussed in the following subsections under the respective headings.

3.4.1.1 Support Vector Machines

Baldwin and Lui, 2010 state, SVM is the best learning algorithm across domains. SVM work better on smaller datasets. We choose SVM, as our dataset is relatively smaller. In the case of language identification for code-mixed Telugu-English data, the overlap among the target classes is marginal. They are known to perform better in the case of lower class overlap.

Support Vector Machines are also effective in high dimensional spaces. Each data item is a point represented in n-dimensional space, where n is the number of features. In this experiment, the feature vector is a bag of character n-grams (bigrams and trigrams) and is augmented with other boolean features like presence of postpositions, emoticons, alphanumeric strings etc. The size of the vector is 4,125. SVMs are effective in high dimensional space also they perform feature extraction inherently. This is extremely important as we are not pruning any features. They are also memory efficient as they use only a few data points (support vectors) in the decision function.

Another reason to use SVM is, sparse character vector for words. For, each word there are only a few non-zero entries. SVMs work well with dense concepts and sparse instances. In this experiment we use sklearn's [3] LinearSVC (linear support vector classification) with a "linear" kernel implemented in liblinear((no kernel transformations) to ensure good scalability.

3.4.1.2 Decision Trees

Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. This is the best way to explore data and model non-linearities. This is a non-parametric method, as there are no assumptions on space distribution and classifier structure. It will segregate the data items based on all values of variables (features) and identifies the variable, which creates the best homogeneous sets of points . There are many algorithms to find the best splitter variable like Information gain, Gini and Entropy. Gini is the default metric used for this experiment. The feature vector is the same as the one used for SVM. We use sklearn to implement Decision Trees.

3.4.1.3 Neural Networks - Multiple Layer Perceptron(MLP)

MLP does not fall into the group of traditional machine learning algorithms. MLP or Neural networks do not make any assumption regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration when compared to other probability

[3] http://scikit-learn.org/stable/

based models. As in the case of previous experiments, a bag of character n-grams (bigrams, trigrams) is used, augmented with seven other class specific features. We used *keras*[4] to implement the neural network. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. The architecture of the model can be seen in Figure 3.6 . Here "vec" means the bag of n-grams vector and we augment 7 class specific features to the vector. Multiple layer perceptron with just one hidden layer is equivalent to a logistic regressor. This also gives us an idea if the data is linearly separable or not.

```
model = Sequential()
model.add(Dense(32,input_dim = len(vec)+7))
model.add(Activation('relu')
model.add(Dense(7,input_dim = 32))
model.add(Activation('softmax'))
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics = ['accuracy'])
model.fit(train_data,train_labels,nb_epoch=100,batch_size= 128)
test_predict = model.predict(test_data,batch_size= 128)
```

Table 3.6 Multiple Layer Perceptron Architecture

3.4.1.4 Conditional Random Fields

Conditional Random Fields is a popular probabilistic method used for structured prediction. This method takes the context and sequence into account while doing the prediction. These are hard to train, but are more accurate. It uses a template of features, according to which it learns the model based on the training data. Unlike HMMs(Hidden Markov Models) CRFs are discriminative models based on conditional probability distribution, therefore, resulting the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Assumptions of HMM are:

1. The next state is only dependent on the previous state. (first order HMM)
2. Current observation is independent of previous observations.

CRF also overcomes the label bias problem of Maximum Entropy Markov Models by using a global normalizer. CRF has been implemented using CRF++ [5]. Following are the features used to train a CRF :

1. Lexical Features
    - Word
2. Sub-Lexical Features

[4] https://keras.io/
[5] https://taku910.github.io/crfpp/

- suffix and prefix character strings
- infix character strings
- prefix, suffix character strings of neighboring words

3. Class Specific Features and Others

    - Presence of postpositions
    - Length of the word
    - Presence of emoticons
    - Presence in the lexicon
    - Checking for number patterns
    - Checking for alphanumeric patterns
    - Checking for Upper Case

The following template gave the best results for CRF:

CPS	CI	WL	NPS	NW	PSP
3,2,1	3	no	3	2	no

1. If CPS = k, k length suffix and prefix strings of the current word are taken as a feature.

2. If CPS = k1, k2,.. then k1, k2,.. length suffix and prefix strings of the current word are taken as a feature.

3. If NPS = k, k length suffix and prefix strings of neighboring words are taken as a feature.

4. If NPS = k1, k2,.. then k1, k2,.. length suffix and prefix strings of neighboring words are taken as feature.

5. If CI=k, k length infix string form the current word is taken as a feature.

The class-wise results of these experiments are reported in Table 3.7, 3.8, 3.9 and 3.10 respectively. In the first set of experiments, both the training and testing data are obtained from ChaiBisket. The data statistics are shown in Table 3.4 and a K fold cross validation (K=3) technique has been adopted to perform the experiments.

Class	Precision	Recall	F1-score
English	93.00	98.67	95.67
Telugu	94.67	96.67	95.67
Mixed	50.00	7.67	**11.67**
Universals	100.00	99.67	99.67
Named Entities	50.00	13.67	21.34
Acronyms	66.67	19.34	29.34
Others(X)	14.34	35.00	11.67

Table 3.7 SVM Results

Class	Precision	Recall	F1-score
English	96.67	95.67	96.34
Telugu	90.67	98.67	94.67
Mixed	20.67	7.67	7.00
Universals	100.00	99.00	99.00
Named Entities	44.34	16.00	23.34
Acronyms	54.34	18.34	27.34
Others(X)	22.34	37.00	**26.67**

Table 3.8 Decision Tree Results

Class	Precision	Recall	F1-score
English	96.00	98.34	**97.00**
Telugu	95.00	98.34	**96.67**
Mixed	16.67	6.67	9.67
Universals	100.00	99.67	**99.67**
Named Entities	51.67	24.67	**32.67**
Acronyms	50.67	22.00	**29.67**
Others(X)	12.00	35.00	15.67

Table 3.9 MLP Results

Class	Precision	Recall	F1-score
English	96.34	98.67	**97.34**
Telugu	94.00	99.00	**96.67**
Mixed	0.00	0.00	0.00
Universals	100	99	**99.67**
Named Entities	77.67	8.34	15.34
Acronyms	83.34	6.00	11.67
Others(X)	0.00	0.00	0.00

Table 3.10 CRF Results

3.4.2 Set 2 Experiments and Results

Though MLP performs the best for most of the classes, the performance of CRF is comparable too. Therefore, both MLP and CRF are used in the second set of experiments. The primary aim of this experiment is to use the two data sets A and B as Training and Testing alternatively and observe a class specific performance difference if any and investigate the reasons for the same. The SET 2 of experiments is again divided into two sub-experiments. The training data and testing data used for the first part are sampled from ChaiBisket and ICON, 2016 Facebook data and Twitter data respectively. In the second part, ChaiBisket Data is used for testing and the ICON, 2016 FB and Twitter data is used for Training.

Class	Precision		Recall		F1-Score	
	CRF	MLP	CRF	MLP	CRF	MLP
English	0.68	0.66	0.88	0.86	**0.77**	0.75
Telugu	0.56	0.63	0.96	0.94	0.70	**0.75**
Universals	0.98	0.99	0.15	0.23	0.26	**0.37**
Named Entities	0.19	0.19	0.12	0.31	0.15	**0.23**

Table 3.11 Training: CB, Testing: ICON,2016_TWT

Class	Precision		Recall		F1-Score	
	CRF	MLP	CRF	MLP	CRF	MLP
English	0.70	0.72	0.89	0.91	0.79	**0.81**
Telugu	0.56	0.64	0.94	0.92	0.70	**0.76**
Universals	0.97	0.99	0.20	0.31	0.33	**0.48**
Named Entities	0.35	0.35	0.17	0.34	0.23	**0.35**
Acronyms	0.12	0.06	0.03	0.05	0.04	0.06

Table 3.12 Training: CB, Testing: ICON,2016_FB

Class	Precision	Recall	F1-score
English	0.96	0.91	**0.94**
Telugu	0.97	0.92	**0.95**
Universals	0.66	1.00	**0.80**
Named Entities	0.57	0.08	**0.14**

Table 3.13 Training Data : ICON, 2016 Code-Mixed Facebook Data ; Testing Data:Code-Mixed Data from ChaiBisket

Class	Precision	Recall	F1-score
English	0.97	0.96	**0.97**
Telugu	0.94	0.99	**0.96**
Universals	0.98	0.99	**0.99**
Named Entities	0.49	0.24	**0.32**

Table 3.14 Training Data : ICON, 2016 Code-Mixed Twitter Data ; Testing Data:Code-Mixed Data from ChaiBisket

3.4.3 Discussion and Conclusion

1. In the first fold of experiments, Conditional Random fields have performed the best for the classes English, Telugu and Universals, but performed poorly for other language classes like ”Mixed”, ”Named Entities” and ”Acronyms”.

2. On the other side, the performance of MLP is comparable(almost equal) to that of CRF w.r.t to ”English”, ”Telugu” and ”Universals” and MLP also performs well for the other classes mentioned in the taxonomy for language identification in this work.

3. There are some cases where CRF has recognized a class right but MLP has failed. This is an example where CRFs sequence modelling power comes into the scene. Consider an example:

   In our data set we have a word **mike** which has occured as an English word more than a Telugu word. In a sentence like the following naaku hero evaro telidu mike teliyali, MLP identifies this as English whereas CRF identifies rightly as Telugu.

4. In the first part of second fold of experiments, code-mixed Chaibisket data is used for training and two individual test sets - Facebook Telugu-English code-mixed data released at ICON-2016 and Twitter Telugu-English code-mixed data released at ICON-2016 has been used for Testing. It is evident from the previous experiments that CRF and MLP apart from giving the best results are also comparable. Therefore, we tabulate the Precision, Recall and F1-scores of both the learning algorithms. Tables 3.12 and Table 3.11 shows the results ans once again MLP wins the race.

Class	Precision	Recall	F1-score
English	0.78	0.86	**0.82**
Telugu	0.75	0.91	**0.82**
Universals	0.82	0.61	**0.70**
Named Entities	0.32	0.22	**0.26**

Table 3.15 Training Data : ICON2016 Code-Mixed Facebook Data ; Testing Data: ICON2016 Code-Mixed Twitter Data

5. The MLP model performed better on ”Mixed” , ”named entities” and ”Acronyms”. As we have not included any class specific handcrafted features for these classes, CRF fails to recognize, where as MLP does, as it internally learns which features are important for a particular class.

6. In the second part of set II experiments, we use two individual training sets- Facebook Telugu-English code-mixed data released at ICON-2016 and Twitter Telugu-English code-mixed data released at ICON-2016. Testing is performed on the ChaiBisket data. For this part CRF has been used instead of MLP as it more memory efficient and our main focus is only on four classes namely English, Telugu, Universals and Named Entities. The definition of these classes is uniform over all the datasets used for this work. Tables 3.13 and 3.15 show the F1-score scores.

7. In the final experiment, both training and testing are social media data. Training is done ICON2016-Facebook data and Testing is done on ICON2016-Twitter data. The results are tabulated in Table 3.15.

8. In set II of experiments, the language identification module has given best results when trained on social-media data and tested on Chaibisket, on contrary training on Chaibisket Data and testing on social media was not that effective in relation to the former. Some reason for this are:

    - Spelling errors in the social media data. Example: ”Interested” is written as ”intrstd”.These mistakes are very rare in the chaiBisket Data. As social media data covers a lot of variation in this context, the character level n-grams succeed in capturing the character sequences well.

    - The F1-score for universals is low in Table 3.12 and 3.11 . It has been observed that the gold annotation for many classes in the ICON2026 Twitter and Facebook data has been flawed. Our model trained on ChaiBisket data correctly annotates the tags that have been wrongly tagged by human annotators. Now we have a rightly tagged social media code-mixed gold corpus.

9. The overall F1-score of the MLP language identification system we propose for code-mixed(Telugu-English) data is **97 for English, 96.67 for Telugu, 9.67 for Mixed, 99.67 for Universals, 32.67 for named entities, 29.67 for Acronyms and 15.67 for Other class.**

3.5 Affect of Language Identification on POS tagging of Code-Mixed Data

We used the developed language identification module for a POS tagging experiment where we combine POS taggers of individual monolingual languages. The accuracy of the POS tagger for Code-Mixed Telugu-English is 52.37%. On experimenting with different templates of features and using a backward search feature selection algorithm we have identified that, the language information of a particular word is not of any significant help in POS tagging. POS tagging problem is more dependent on the class context of the words around it, rather than the language they belong to.

Chapter 4

Dialog Act tagging for Code-Mixed Conversations: Conversational Data Collection, Annotation Scheme, and Experiments

4.1 Introduction and Motivation

The phenomenon of Code-Mixing can be studied by analyzing a variety of applications. Apart from being a commonly used spoken form in multilingual settings, CM also manifests itself on social media sites in the form of posts, comments, replies to the comments and most importantly chat conversations. Most informal and semi-formal conversations are fabricated in CM fashion. The richest form of code-mixing is observed in day-to-day conversations. Code-mixed conversations primarily occur among people who share a similar language background, for example, in scenarios where the formal education is received in a language other than the persons native tongue. Previous studies show that the reasons for facebookers to switch language is 45% due to real lexical need, which further resulted in 58.97% of inter-sentential switching and 33.33% of intra-sentential switching [26]. Popularly used personal assistants like Siri, Cortana, Alexa etc., currently do not handle this case of language switching in the course of a conversation.

The ability to understand language remarks humanity and intelligence, and conversation or dialog is the most fundamental and privileged arena of language. Dialog that happens in a bilingual setting is rich with inter and intra sentential mixing. The term ”Dialog” has its roots in the Greek word, which means conversation. Dialog has been studied through various lenses like linguistics and Cognition. Recently, it has attracted the field of computation too. In this work, an attempt is made towards understanding the phenomenon of code-mixing by adopting a computational approach to handle an aspect of Dialog. To understand this better, one needs to know the typical architecture of a Dialog System .

4.2 Dialog System and its Components

Task-oriented Dialog systems are designed for a particular task and are set up to have short conversations to get information from the user to help them complete the task. These include the digital assistants

that are now on every smart phone (Siri, Cortana, Alexa, Google Now/Home, etc.) whose dialog agents can give travel directions, control home appliances, find restaurants, or help make phone calls or send texts. On the other hand Chatbots[28] are also dialog systems, designed for extended conversations, set up to mimic the unstructured conversations or chats characteristic of human-human interaction. Figure 4.1 depicts a generic dialog system architecture with following modules [1] .

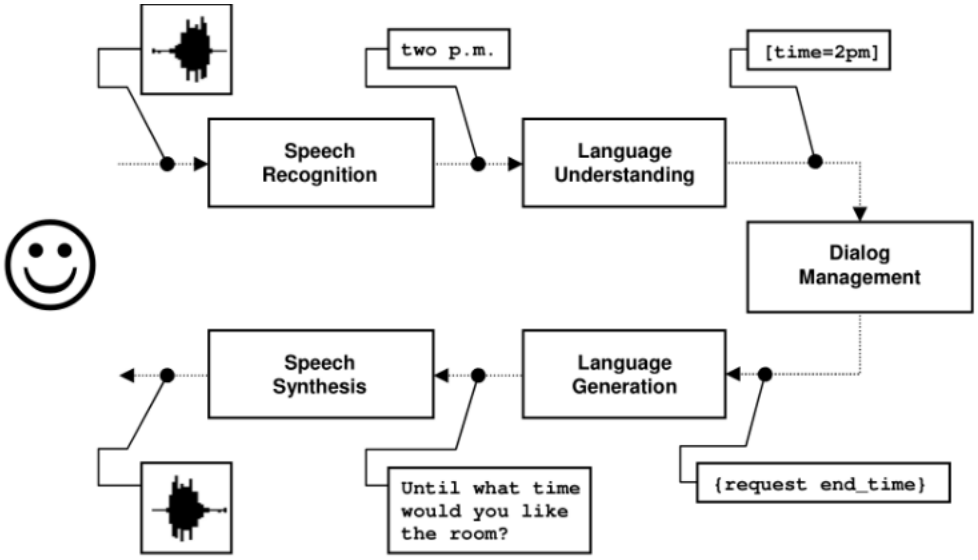

Figure 4.1 Typical Dialog System Architecture

Speech Recognition: An automatic speech recognition module that takes speech as input, decodes it and coverts it into text.

Natural Language Understanding (NLU): Most natural language understanding systems share some common components. This module needs a lexicon of the language and a parser and grammar rules to break sentences into an internal representation. This module can also include other tools like Named entity recognizer, intention recognizer etc.

Dialog Management : A Dialog Manager handles the state and flow of a conversation. Input to Dialog Manager is an internal representation of a natural language utterance.

Natural Language Generation (NLG) : Natural Language generation is a task of producing natural language (text) from a machine representation system such as a knowledge base or a logical form, obtained from Dialog Manager and NLU.

[1] https://dialrc.org/

Speech Synthesis: The natural language text obtained from NLG is converted to Speech, using a Text to Speech conversion module.

Most of the task specific dialog systems that are available today respond by filling the slots or completing a pattern. This method is called template based. They are not intelligent enough or capable of having social interaction. A first step towards social interaction is to recognize and understand the intent of the user and respond accordingly. Dialog Acts can be used for this purpose of intention recognition in a task oriented dialog system.

4.3 Evolution of Dialog Acts

The term "Dialog Act" is related to the term "Speech Act". In linguistics, a speech act is an utterance defined in terms of a speaker's intention and the effect it has on a listener. Essentially, it has a performative function in the conversation. Speech Act theory is a sub-field of pragmatics. Pragmatics stresses on how words can also be used to carry out actions apart from conveying information.

Speech Act theory was introduced by J.L Austin, in the year 1975. According to Austin, an utterance has three layers of interpretation. He classifies all the utterances into three categories- locutionary, illocutionary, or perlocutionary acts. Following is a brief description of these acts as summarized by Seyyed Mohammad [2].

1. **Locutionary Act** : Locutionary act is the act of saying literal meaning of a sentence. The meaning of the sentence that is conveyed through the words and structure of the sentence (arrangement of words).

2. **Illocutionary Act**: Illocutionary acts are performed through the communicative force of an utterance. This encapsulates the pragmatic meaning of the utterance. It is the act that is performed by saying the utterance. For example , through the utterance, "It is raining outside!", apart from giving information, the speaker wants the hearer to use an umbrella if the hearer wants to go outside, or, the speaker want the hearer not to go outside and stay inside the room. An intention of the utterance might be to make a statement, an offer, an explanation, or for some other communicative purpose.

3. **Perlocutionary Act** : . Perlocutionary act is the hearer's reaction toward the speaker's utterance.

   All the above mentioned acts are illustrated in Table 4.1

Searle in his work [41] in 1979, has introduced the concept of speech acts (assertives, directives, commissives, expressives and declaratives) [3], that come under Austins illocutionary acts. Searle proposed that all *speech acts*[3] can be classified into one of these five classes. Following is a brief description of the acts :

[2]http://hariku23.blogspot.in/2015/01/locutionary-illocutionary-and.html

[3]http://www.coli.uni-saarland.de/projects/milca/courses/dialogue/html/node66.html

Utterance : I have substantial amount of money in my savings account	
Locution (utteracne)	I have substantial amount of money in my savings account (conveying information)
Illocution (meaning)	an act of offering the hearer to ask for money or a dinner treat, depending on the context.
Perlocution (reaction)	the hearer asks for some money, or asks for a dinner treat.

Table 4.1 An Example of Austin's Speech Acts

1. **Assertives:** They commit the speaker to something being the case. - suggesting, putting forward, swearing, boasting, concluding.
   Example: No one can dance better than Michael Jackson

2. **Directives:** They make an attempt towards making the addressee perform an action - asking, ordering, requesting, inviting, advising, begging.
   Example: Could you please pass the salt?

3. **Commisives:** The Speaker commits to do something in the future - promising, planning, vowing, betting, opposing.
   Example: I am going to attend the meeting tomorrow.

4. **Expressives:** They express how the speaker feels about the state or situation he or she is in - welcoming, deploring, aplogizing.
   Example: I am really sorry about yesterday's dinner.

5. **Declaratives:** The state of the world is changed in an immediate way. - depends on the authority of speaker.
   Example: You are fired!

Speech act is a generic term. Dialog Act is a specialized speech act [4]. For example, A dialog act includes both the semantic and communicative aspect of an utterance. To elaborate, the semantic aspect of an utterance in a Dialog tries to figure out "What the utterance is about?"- What objects, events or situations it relates to. The communicative aspect refers to what purpose the information in the utterance serves the dialog as a whole - whether it is to warn the addressee, or direct him towards doing something in the future or to explain about the information present in the utterance.

The semantic component corresponds to the information that obtains a certain place in the state of the addressee, and the communicative component describes the precise place that the information obtains once the utterance is understood correctly. This computational approach to dialog is called "Information-State" or "Context-switch".[13]

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

[13] defines **dialog act as a unit in the semantic description of communicative behavior produced by a sender and directed at an addressee, specifying how the behavior is intended to influence the context through understanding of the behavior**

Dialog modeling is a process of understanding and generating dialogs. Recognizing speaker's intention plays a crucial role in understanding utterances and Dialog Acts are helpful for this purpose. A Dialog Act tag-set classifies the utterances based on syntactic (surface features), semantic and pragmatic (context) features. In a dialog, there can be instances where even though the utterance contains the same words, the context in which they occur can mean different things. Following is an illustration of such case:

Example 1

CASE I:
Speaker A: nenu cinemaki vellali.
Translation: I have to go to the movies.
Speaker B: aa table mida cash undi tisko.
Translation: There is money on that table. Take it.

CASE II:
Speaker A: reyy market ki vastava ?
Translation: Hey, Do you want to come to the market?
Speaker B: nenu cinemaki vellali.
Translation: I have to go to the movies.

Example 2

CASE I:
Speaker A: Library timings enti?
Translation: What are library timings?
Speaker B: Library timings 10AM to 5PM.
Translation: Library is open from 10 AM to 5 PM.

CASE II:
Speaker A: nenu 6PM ki book return chestha ?
Translation: May I return the book at 6PM ?
Speaker B: Library timings 10 to 5
Translation: Library is open from 10 AM to 5 PM.

In case I of first example, the intention of speaker A, apart from conveying information is to request or ask for money, which is understood by Speaker B. In the second case, Speaker B utters the same sentence but the primary intention is to convey that he cannot go to the market with Speaker A. Similarly, in case I of second example, speaker B is answering a question, whereas in the case II, the same utterance is spoken by Speaker B in a different context, the intention (communicative force) of the utterance is to say a "NO", but that cannot be gathered from mere words ignoring the context. Hence recognition of intent is necessary for understanding and further processing of the utterance in a dialog.

4.4 Conversational Data Collection and Annotation

4.4.1 Wizard of Oz (WOz)

To the best of my knowledge there is no code-mixed conversational data publicly available for any language pair. Therefore, the work started with construction and procurement of dialog data. English-Telugu conversational data was collected through WOz experiment. A **Wizard of Oz** experiment is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated by an unseen human being. 28 participants were involved in this including students and faculty (library staff) of an educational institute. Out of these 3 people assumed the role of a virtual library assistant (a computer system) and the remaining 25 interacted with the wizards with various queries, resulting in 25 conversations. To ensure diversity within the domain, participants were not provided with specific tasks and were free to ask any questions pertaining to library.

Initially, they were asked to use Telugu, but most of the participants started mixing English. When asked to stick to a single language (Telugu), the response time of the participant increased, and the conversation lost its naturalness. So, this strict imposition was removed and instead we asked them to speak to the system in a natural way as how they would pose the question to another peer who knows Telugu. Along with *code-mixing*[5], some participants also did *code-switching* [6]. So, in a given turn of a speaker, there can be three possibilities: An utterance is either completely in English, Telugu, or is code-mixed. Resembling cross-scripting observed in social media like Facebook, Twitter and massive usage of English keyboards in Romanizing native languages, this data is also collected in similar cross-scripted manner via a chat interface. Table 4.2 depicts an excerpt from a conversation collected through WOz. A total of 25 conversations were collected and Table 4.3 shows statistics of the data.

[5]Code-Mixing is the embedding of linguistic units such as phrases, words and morphemes of one language into an utterance of another language.

[6]Code-Switching is juxtaposition within the same speech exchange of passages of speech belonging to two different grammatical systems or sub-systems

Speaker	Utterance	Translation
SYS:	Hi nenu mee Library Assistant	I am your Library Assistant
	meeku ela help cheya galanu ?	How can I help you ?
USER:	Hello	Hello
	linear algebra books section ekada undi ?	Where is linear algebra books section ?
SYS:	Linear Algebra books meeku section 3.2 lo dorukutundi	You will find Linear Algebra books in section 3.2
	meeku ye book kavali	Which book do you want ?
USER:	Naaku Linear Algebra by Russel norvig third edition kavali.	I want the third edition of Linear Algebra by Russel norvig

Table 4.2 Sample Data

Total no. of conversations	25
Total no. of participants per conversation	2
Total no. of utterances	856
Total no. of unique utterances	636
Total no. of words	4,270
Total no. of unique words	1,147
Average no. of turns/conversation	21
Average no. of utterances/conversation	34
Average no. of words/conversation	171
Average no. of words/utterance	5
Average no. of utterances/turn	2-3
English Utterances	172
Telugu Utterances	93
Code-mixed Utterances	352

Table 4.3 Data Statistics

4.4.2 Dialog Act Annotation

4.4.2.1 Dialog Act Markup in Several Layers (DAMSL)

Over the years many researchers have noticed that speech-act tries to capture, the purpose(s) of an utterance with a single label. This is problematic, as an utterance might simultaneously perform actions such as responding to a question, confirming, understanding, promising to perform an action and informing [3]. DAMSL is a tagging scheme, that addresses this problem by allowing multiple tags at multiple layers. The classes in the DAMSL tag-set are high-level and are designed to be applicable to various types of dialogs independent of domain. These classes can be subdivided into finer sub-classes that are relevant to the domain.

The scheme has three layers - Forward Communicative Functions, Backward Communicative Functions and Utterance features. The Forward Communicative Functions consist of a typology in a similar style as the actions of traditional speech act theory. The Backward Communicative Functions indicate how the current utterance relates to the previous dialog, such as accepting a proposal, confirming un-

derstanding or answering a question(grounding utterances). Utterance features encode how the content of the utterance is relevant in the conversation. DAMSL-SWBD tag-set consists of a total of 42 tags. Figure 4.2 shows basic DAMSL architecture.

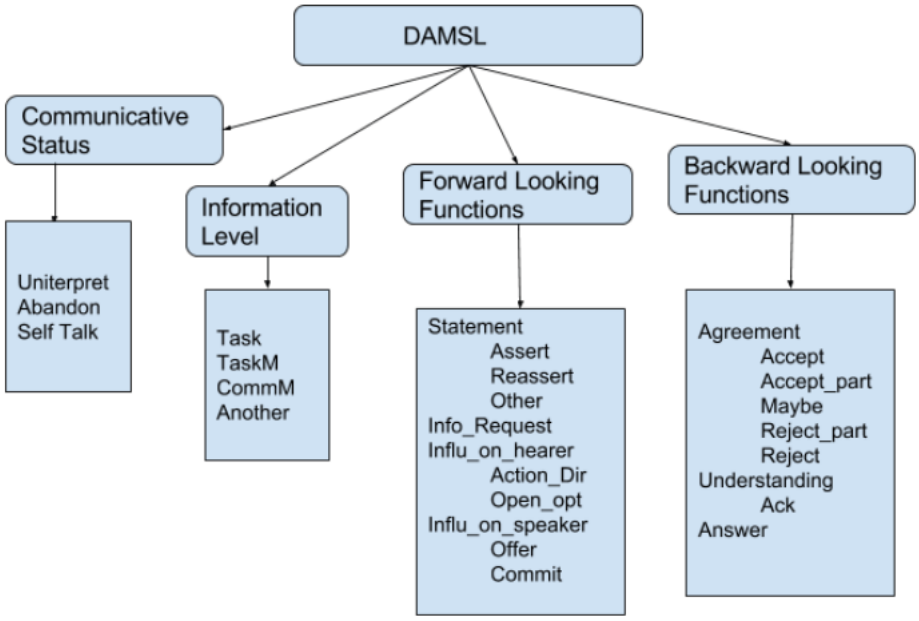

Figure 4.2 DAMSL architecture

4.4.3 Annotation Schema

DAMSL-SWBD tag-set consists a total of 42 tags. The 42 tags are clustered into smaller groups, under original coarser DAMSL classes. These coarser tags are used to annotate the collected Telugu-English Code-Mixed Library domain corpus. The chosen tag-set is completely domain independent and can be extended to various domains. The concept of forward and backward layered communicative functions has been adopted, and each utterance has been annotated at three layers - Forward Communicative Function, Backward Communication Function and Task. The original DAMSL [3] tag-set has been tailored, as all the tags are not used in annotating the data procures through Wizard of Oz. Table 4.4 shows the taxonomy of the Dialog Acts used for annotation. The ”TASK” tag, in the informational level of DAMSL tagging scheme, is here extended to have 9 tasks out of which 8 are domain dependent. We observed that the utterances pertaining to the library domain could be broadly related to 8 tasks. Table 4.5 gives the description of these tasks.

The collected 25 conversations were annotated by 2 annotators at three levels as described in table 4.4. Every utterance is tagged at all these three levels. The annotators were provided with the DAMSL *annotation guidelines* [7]. The present tailored DAMSL tag-set used for annotating code-mixed Telugu-English library domain corpus has 9 Forward communicative tags and 6 Backward communicative tags.

[7] https://www.cs.rochester.edu/research/speech/damsl/RevisedManual/

Forward Communicative Function	Backward Communicative Function	Information Level
Assert	Accept	CM
Info_Request	Reject	Task0
Action_Directive	Maybe	Task1
Open_Option(OO)	Hold	Task2
Offer	Ack	Task3
Commit	Answer	Task4
Explicit_Performative		Task5
Greeting		Task6
Greeting_EOC(end of conversation)		Task7
		Task8
		Other

Table 4.4 Tailored DAMSL Tag-set

TASK	Description
TASK0	Goal Elicitation
TASK1	Book Enquiry
TASK2	Library Enquiry
TASK3	Regarding Issue
TASK4	Regarding Reissue
TASK5	Regarding Return
TASK6	Regarding Reserve
TASK7	Regarding Membership
TASK8	Miscellaneous

Table 4.5 Task Tag Description.

A NULL tag is included in both Backward and Forward Looking Functions, which marks the absence of a relation between the current speakers utterance and the previous speakers utterance and in cases of grounding utterances. For example an utterance like okay is tagged as NULL in the Forward Looking layer. Cohens kappa was calculated and an inter-annotator agreement of 87.19 has been obtained, which is a sign of reliable data.

Table 4.6 shows an excerpt from one conversation out of the 25 conversations collected through WOz. Table 4.7, gives the class wise Dialog Act statistics of the code-mixed conversational data.

4.5 Automatic Dialog Act Recognition

The major goal of this work is to find answers to questions like, whether the phenomenon of code-mixing affects dialog act recognition ? If so, what are the challenges and how can they be approached?

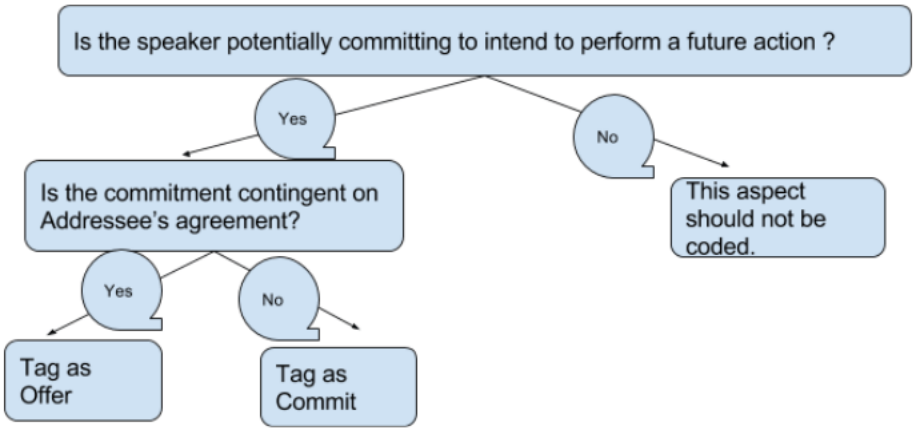

Figure 4.3 Decision Tree for influencing speaker's future action

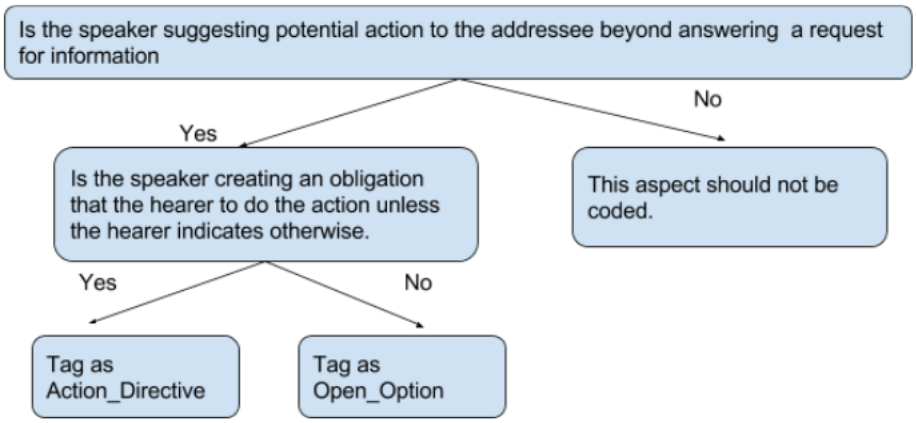

Figure 4.4 Decision Tree for influencing addressee's future action

Can the existing tools and resources serve the purpose of automatic dialog act recognition for code-mixed conversations? As a first step towards finding answers to these questions, our work began with some background study of the already existing automatic dialog act recognition techniques.

4.5.1 Literature Survey

Many annotation schemes have been developed from projects like DAMSL, a domain independent DA annotation schema, [3], Maptask [4] and Verbmobil [14]. The previous work in this domain is performed on corpora that is not fully representative enough of the casual day-to-day conversations in a multilingual environment.

Automatic Dialog act recognition has been done for human-human speech conversations [38]. Many experiments have been done on the famous [8]SWBD (switchboard) corpus which is annotated with

[8]http://compprag.christopherpotts.net/swda.html#tags

Speaker	Utterance	FWD:BWD:TSK
SYS:	Hi nenu mee Library Assistant	GREETING:NULL:CM
	meeku ela help cheya galanu ?	OFFER:NULL:TASK0
USER:	Hello	GREETING:ACK:CM
	linear algebra books section ekada undi ?	INFO_REQUEST:ACCEPT:TASK1
SYS:	Linear Algebra books meeku section 3.2 lo dorukutundi	ASSERT:ANSWER:TASK1
	meeku ye book kavali	INFO_REQUEST:NULL:TASK1
USER:	Naaku Linear Algebra by Russel norvig third edition kavali.	ASSERT:ANSWER:TASK1

Table 4.6 Sample Conversational Data

DAMSL tags. Some papers [38, 44] have included acoustic features along with the textual ones and have obtained good performance with supervised learning methods like Naive Bayes, KNN and SVMs.

DA recognition problem has also been looked at as a sequence labeling problem. HMM based approaches using viterbi algorithm were experimented on to obtain the DA tag sequence given the utterance sequence [45]. But, since one turn of a speaker can contain more than one utterance, HMM modelling needs to be carefully done fixing an appropriate window. Neural net models have also been proposed, which work best for large data sets.[40]. Unsupervised approaches such as clustering have also been used previously [21].

Arabic and Spanish are some other languages in which some work on DA annoation and recognition has been published.[18, 40]. Also semi supervised Dialog Act tagging approach has been proposed for Telugu [19].The data set used has very short dialogs, at most two dialog acts per conversation.This short a dialog cannot be used for effective conversational analysis.

A observation that has been common through all the statistical methods adopted is that the use of n-grams as features has always produces decent DA recognition accuracy. DA tagging using word i.e. just unigrams can get an accuracy of 50% according to [22]. This gives an insight that Markovian modeling of n-grams can improve the performance of the system significantly. [48] shows that instead of all n-grams, using only those that pass a learned threshold will be sufficient to obtain a good accuracy, but also points out that this claim is only true in case of a large learning corpus.

Forward Communicative Tags	#count	Backward Communicative Tags	#count
INFO_REQUEST	248	ACK	263
ACTION_DIR	20	HOLD	47
EXPLICIT_PERFORMATIVE	44	ACCEPT	33
OO	14	REJECT	3
OFFER	29	MAYBE	1
GREETING	83	ANSWER	251
ASSERT	286	NULL	258
COMMIT	15		
GREETING_EOC	42		
NULL	75		

Table 4.7 Dialog Act tag wise statistics

[29] has proved that using syntactic features obtained from parsing a sentence using automatic parsers will help in DA recognition.

4.5.2 Our Approach

We model the dialog act recognition problems as a classification problem.As the size of the code-mixed corpus available is small, we began by exploring the traditional n-gram based methods. Three supervised learning algorithms and one non-parametric method has been used as a part of initial experiments.

4.5.2.1 Data-Preprocessing

Utterance Segmentation

A speaker could utter more than one sentence per turn and some utterances could be shorter than a sentence (for example: 'Sare'(Okay) and 'thappakunda'(Sure)). In chat text, multiple punctuation marks could be an indication to a pause.For the task of tokenization, NLTK sentence and word tokenizers have been deployed. In the following example, punctuation is used to signify pause, but sentence tokenizer would separate them into two different sentences.
Before Segmentation: *USER: ikkada issue cheste...mari nenu eppudu collect chesukovali?*
After Segmentation: *USER: ikkada issue cheste mari nenu eppudu collect chesukovali?*
Hence we have manually checked for such instances.

Spelling Errors and abbreviations

The data has a lot of properties that an SMS or chat would have, like usage of short forms, phonetic representation of words, absence of punctuations, elongation of vowels and use of consecutive punctuation(exclamation) marks to express excitement. Following are a few examples from the data:

USER : Ohh ok. Can u check if a book is available ?
USER : lib timings enti?
USER : oh..coooool...
USER : Ohh thankss

These kind of occurrences demand a standardization before any further processing. After the standardization **u** maps to **You**, **thankss** and **cooool** will map to **thanks** and **cool** respectively.

4.5.2.2 Validation Technique

As the size of procured code-mixed data is small and has a high class imbalance, leave-p-out cross-validation has been adopted to avoid biased class distribution in the training data. Experiments were run for P=1, P=2 and P=3.The best results were obtained for P=1 leave-p-out cross-validation. These results have been documented in this thesis.

4.5.2.3 Naive Bayes

For choosing the best dialog act tag, Naive Bayesian interpretation is used.

$$\hat{T} = \underset{T}{argmax} \frac{P(U,T)}{P(U)} \tag{4.1}$$

where $\hat{T}$ is the desirable tag from the tagset T for utterance U.

For, ngrams the above eqaution is modified to

$$\hat{T} = \underset{T}{argmax} \prod_{i=1}^{N} \frac{P(w_i,T)}{P(U)} \tag{4.2}$$

where w_i represents the n-gram sequence and N is equal to the list of n-grams obtained for the utterance U.

A Naive-Bayes maximum likelihood estimation has been first performed using only unigram, bigram and trigram probabilities separately. This experiment was done without incorporating any context information from the previous utterance.

4.5.2.4 Support Vector Machines

Linear Support vector classification with parameter kernel="linear", but implemented in terms of liblinear rather than libsvm has been used to perform the "crammer_singer" multi-class classification. The same leave-p-out cross-validation technique has been followed. The best results were obtained when hyperparameter C was set to 1.

Dialog Act	Unigrams	Unigrams + Bigrams	Unigrams + Bigrams+ Trigrams
GREETING	84.94	85.36	86.69
GREETING_EOC	89.34	89.34	89.34
ASSERT	76.26	76.69	76.38
INFO_REQUEST	75.07	76.88	77.46
OFFER	61.34	63.2	63.2
COMMIT	0.00	0.00	0.00
OO	0.00	16.00	16.00
ACTION_DIRECTIVE	2.28	0.04	0.04
EXPLICIT_PERFORMATIVE	76.19	77.39	76.59
System F1-score for fwd tags	**75.57**	76.52	**76.59**
HOLD	29.18	33.88	34.07
ACK	67.38	67.83	68.06
MAYBE	0.0	0.0	0.0
REJECT	63.24	62.24	62.38
ACCEPT	72.34	71.89	71.01
ANSWER	60.13	61.69	60.83
System F1-score for bwd tags	60.92	**61.64**	61.40

Table 4.8 Naive Bayes Results

4.5.2.5 Kth Nearest Neighbor Alogithm

KNN classification is a memory based learning approach.KNN classifier from *sklearn* [9] has been imported to serve the task of classification. Experiments were run for K=3, 5 and 7 and it has been observed that best results were obtained for K=3 with feature set remaining the same as in previous supervised learning methods used.

4.5.2.6 Hidden Markov Models

Hidden Markov Model (HMM) is a generative sequence to sequence model. This experiment has been performed to examine the relation between consecutive forward looking functions and consecutive backward looking functions. Emission probability of an utterance is calculated as the product of the emission probabilities of individual unigrams that make up the utterance, then with bigrams and trigrams respectively. Transition probabilities between dialog acts are obtained from the training data. The problem at hand is to estimate the optimal sequence of hidden states, given the model parameters (emission and transition probabilities) and observation states.

[9] http://scikit-learn.org/stable/

Dialog Act	Unigrams	Unigrams + Bigrams	Unigrams + Bigrams+ Trigrams
ACTION_DIR	42.54	35.86	35.86
EXPLICIT_PERFORMATIVE	75.34	74.09	74.09
OO	24.00	20.00	20.00
OFFER	77.6	78.0	76.67
INFO_REQUEST	79.68	79.54	77.00
GREETING	84.67	85.68	86.48
ASSERT	79.67	78.60	77.71
COMMIT	10.67	8.00	8.00
GREETING_EOC	89.34	89.34	89.34
System F1-score for fwd tags	**81.36**	80.35	79.20
HOLD	32.84	34.72	35.02
ACK	70.08	71.87	72.05
MAYBE	0.0	0.0	0.0
REJECT	59.0	53.00	53.12
ACCEPT	60.94	59.34	58.34
ANSWER	63.00	70.74	60.05
System F1-score for bwd tags	64.69	**67.97**	67.49

Table 4.9 Support Vector Machines Results

4.5.2.7 Discussion and Future Work

As it is evident from the experimental results, Hidden Markov Models (HMM) with unigrams as features gave the best results with an average F1-score of 72.30 and these results are comparable with SVM. HMM adds contextual information of the previous dialog act tag and therefore performs the best. Following is an example from the collected conversational data in Table 4.12

1. In the above case, the participant is directing the library assistant (wizard) to issue the two books, this is an ACTION_DIRECTIVE, but this information cannot be derived from the surface of the sentence i.e. the constituent words. Context plays a crucial role here. Knowing that the act of previous tag is OFFER and the transition probability between OFFER and ACTION_DIR is significant in the collected data.

2. HMM is precisely a naive Bayes modeling augmented with the information regarding the previous observation state's tag. Both these methods are probability based. Whereas, in our case, there is a combination of features (unigrams + bigrams, unigrams + bigrams + trigrams) and there is no feature pruning step involved. The size of the feature vector 4338. SVM performs better while dealing with large set of features and more number of classes and lesser amount of data as it performs dimensionality reduction and feature extraction inherently. As Naive Bayes and HMM are probability based, sparse data is problematic for these learning algorithms. An other important reason why the combination of HMM and unigrams worked for CM data is that, both the training and testing data are obtained from library domain and a majority of the tasks were about getting

Dialog Act	Unigrams	Unigrams + Bigrams	Unigrams + Bigrams+ Trigrams
ACTION_DIR	15.98	13.34	12.88
EXPLICIT_PERFORMATIVE	63.02	62.50	61.70
OO	25.6	16.0	16.0
OFFER	68.00	73.34	66.67
INFO_REQUEST	66.83	61.00	55.24
GREETING	79.98	73.48	73.23
ASSERT	64.52	60.46	59.35
COMMIT	0.0	0.0	0.0
GREETING_EOC	81.2	73.27	68.60
System F1-score for fwd tags	**68.80**	62.75	59.98
HOLD	36.26	30.74	31.61
ACK	63.14	56.94	55.56
MAYBE	0.0	0.0	0.0
REJECT	0.0	0.0	0.0
ACCEPT	2.67	0.0	0.0
ANSWER	60.26	59.10	58.54
System F1-score for bwd tags	**62.38**	59.74	58.27

Table 4.10 KNN Results

a book issued, returning a book, enquiry about a book etc. Therefore, a lot of domain specific words crawl in and it becomes tad bit easier to identify the DA.

3. Another very important observation is that, from the literature survey, it can be concluded that n-gram based methods give significantly decent DA recognition accuracy. These results are mostly obtained when trained and tested on English corpora. English is a fixed order language, and in the case of code-mixing, most of the sentences follow the structure of Telugu (Matrix language). Telugu is a free word order language. For some classes like Information_Request, unigrams and bigrams, when used together initially increased the F1-score. Furthermore, involving trigrams saw a declination in the performance. n-gram methods are highly dependent on the order of words in a sentence.

4. Code-mixing is not a random phenomenon. Moreover, an utterance can be code-mixed differently by different people and can still mean the same. The free order nature of Indian languages (Telugu or Hindi) combined with the variants in mixing makes it more challenging for n-gram based methods. In the following example, both the sentences translates to " Can I reissue the book after the issue period ends?" and they are expected to me recognized as "INFO_REQUEST", but the n-gram based methods may not completely capture this due to different choice of words and sentence structure.

   An example

Dialog Act	Unigrams	Bigrams	Trigrams
ACTION_DIR	38.95	25.48	0.0
EXPLICIT_PERFORMATIVE	56.62	30.09	17.2
OO	20.26	8.00	12.00
OFFER	67.2	69.2	68.53
INFO_REQUEST	82.76	66.65	51.96
GREETING	86.23	80.82	73.01
ASSERT	77.87	66.43	49.91
COMMIT	10.67	0.0	0.0
GREETING_EOC	58.4	31.06	36.26
System F1-score for fwd tags	**81.81**	64.45	49.80
ACK	59.42	43.67	37.94
MAYBE	0.0	0.0	0.0
ACCEPT	72.94	71.67	69.00
ANSWER	60.98	47.09	43.20
REJECT	63.67	62.89	62.10
HOLD	24.86	24.74	32.85
System F1-score for bwd tags	**62.80**	49.42	46.59

Table 4.11 HMM Results

Speaker	Utterance	Translation	Annotation
Wizard	So only 2 books issue cheyagalatanu.	So, I can issue only two books	OFFER:HOLD:TASK3
Participant	aite naku oka oppenham,oka love story book kavali	In that case, I want Oppenham and a love story book	ACTION_DIR:ACCEPT:TASK3

Table 4.12 Context Specific Example

(a) ante_te reissue_en cheskovacha_te oka_te vela_te maku_te issue_en date_en ipoyaka_te kuda_te use_en cheyalsi_te avasaram_te unte_ ?_univ INFO_REQUEST:NULL:TASK4

(b) issue_en cheskunna_te book_en di_te return_en date_en datesaaka_te kuda_te oka_te vela_te use_en cheyalanukuntunte_te reissue_en cheyinchukune_te chance_en unda_te ?_univ INFO_REQUEST:NULL:TASK4

5. A single turn can contain more that one utterance, so a window of history will give more information on the dependencies than a single previous utterance. Therefore, we believe a windowed HMM would yield a better performance for the DA recognition task at hand. Neural models can be used in the future when there is sufficiently enough data.

To the best of our knowledge, there are no automated parsers developed for code-mixed Telugu-English data. Including POS information from the POS tagger developed for CM social media text has not improved the accuracy in any of the experiments. One possible reason is that as the accuracy of the

developed POS tagger is 52.37%, the error rate is high. So, incorporation of any linguistic features was out of question. Therefore, in the next section of this chapter, we investigate how the already existing corpora and linguistic tools could be put into best use in order to achieve a DA recognition accuracy that is comparable with the HMM accuracy that we have obtained by annotating twenty five code-mixed conversations. If this is achievable, the humongous and cumbersome task of manual annotation for code-mixed data can be avoided , if not completely can be partially avoided by annotating smaller datasets and adopting semi-supervised methods.

4.6 Learning from Resource Rich Sources

Procuring and manually annotating code-mixed data is very time-consuming, cost-ineffective, and resource-intensive. In this section we perform experiments that learn a dialog act recognizer for code-mixed conversations using a huge conversational corpus of a resource rich language. At this juncture, there are two options available, one is learning from the Telugu (Matrix language) corpus and the second one is to learn from English (Embedded language) corpus. To the best of our knowledge, there is no dialog act annotated Telugu conversational data available . A significant amount of work is done on Dialog Acts in the past, but a majority of it is on English. Therefore, we use the switchboard telephonic conversational corpus that is annotated with SWBD-DAMSL dialog acts.

4.6.1 Some Related Work

Some earlier work has been done in the field of machine translation where the statistical machine translation engines trained on one language pair are used to translate another language. After training a reliable model for high-resource language, the cross-lingual similarities are exploited and the model is adapted to work for a close language with almost zero resources. In [34], they choose Turkish and Azerbaijani. Azerbaijani is a resource poor language with no bilingual corpus. In this work, they outperformed all the existed models and were successful in training a Azerbaijani to English translation Engine. This process is popularly known as ”Adaptation technique” or ”Transfer learning”. It is a vital technique that generalizes models trained for one setting or task to other settings or tasks. [46] presents the details and typology of transfer learning.

4.6.2 Intuition

Intuition behind using the English corpus is that, it has been shown in the past work [22] that, dialog act can be recognized from the constituent words of a sentence to an accuracy of 50%. A basic transliteration and word translation model applied to the code-mixed data can generate a sequence of English words. Therefore, there is no practical hindrance in applying the learned model on transformed code-mixed conversations.

4.6.3 Method

1. SWBD (Switchboard) corpus is a collection of 1155 telephonic conversations; at an average, a conversation has 144-turns and 271 utterances. These are goal-oriented conversations but are domain independent. Altogether, the conversations had 22,4650 utterances. SWBD corpus is tagged with 42 SWBD-DAMSL dialog act tags. Initially, these tags have been collapsed to 14 tags to match our tailored DAMSL tag-set, used to annotate code-mixed conversations.

2. Preprocessing of SWBD corpus:

   - SWBD corpus is a collection of telephonic conversational recordings. The respective transcripts are available which are annotated SWBD-DAMSL dialog acts. As this is speech data, sometimes there can be errors in recording and transcriptions. The data is also annotated with some markers that signal uninterpretable utterances, probably wrong transliterations etc. These type of utterances have been removed from the conversations as a first step and it was observed that this decision did not disturb the conversational flow. Also, transcriptions of non-speech like ”cough”, ”clearing of throat”, has been removed from the data.
   - Secondly, some utterances are tagged with a marker ”+”, which means continuation of previous speaker's utterance. This kind of instances occur a lot in spoken dialog. The second step of preprocessing included stitching of such utterances together and mapping them to the dialog act that they are annotated for.
   - The final step and the most important one included, collapsing the 42 SWBD-DAMSL tags to the basic DAMSL tag-set that we have used to annotated our code-mixed conversations. Some of the changes included, addition of a tag ”SNU” that signaled non-understanding, then ”OFFER”, ”COMMIT” and ”OPEN_OPTION” have been collapsed to ”OO_OF_CC”, as SWBD-DAMLS tag-set does not see them as separate tags.[10]. At present, there are a total of 14 dialog acts.

3. In order to use the models that are trained on this English (resource-rich) corpus, the code-mixed conversations are required to be translated into English. However, there are no linguistic tools that does the translation. The unavailability of parallel corpora or comparable corpora leaves no scope for statistical machine translation. The minimum requirement is to have all the words in English. Therefore, a CM sentence is passed through phases of transliteration and lexical translation (word-to-word) one after the other respectively.

   - **Language Identification:** Given a utterance in CM language, we initially identify the language of each word in the utterance. For language identification, we use the CRF system trained on ChaiBisket data whose F1-score is 97.34 and 96.67 for English and Telugu respectively(CRF is more memory efficient than MLP, and the class wise F1-scores of English

[10]https://web.stanford.edu/ jurafsky/ws97/manual.august1.html

and Telugu are comparable). The details of the LID experiment are discussed in Chapter 3 of this thesis. After language identification, each non-English word is transliterated and then translated using Google Translated API.

- **Transliteration:** *Indictrans* transliterator[11] for Roman to Telugu has been used for transliteration . As a first step, the ML based transliteration happens from Roman to Devanagari script and then Devanagari is transliterated to Telugu script. The accuracy of the transliteration is about 65%.
- **Lexical Translation:** Google API has been used to translate individual words - output of indic transliteration, to English.

The output of this preprocessing module is a sentence with a sequence of English words.The following table shows an example of how a code-mixed sentence looks after preprocessing.

Code-Mixed Sentence	Life lo baga success ayyi parents ki dhooram ga untu , vaalla own life lo munigipoyina mugguru pillalu.
After Language Identification	Life_en lo_te baga_te success_en ayyi_te parents_en ki_te dhooram_te ga_te untu_te ,_univ vaalla_te own_en life_en lo_te munigipoyina_te mugguru_te pillalu_te ._univ
After Translation	Life in the well success is parents to distance Ga stay , Their own life in the submerged three children

Table 4.13 Pre-processing of CM utterance

SVM has given the best dialog act tagging F1-score for both forward and backward tags after HMM. In these experiments we use both SVM and HMM. A Support vector classification with parameter kernel="linear", implemented in libsvm has been used to perform the "crammer_singer" multi-class classification. The architecture of HMM is same as described in the previous experiments. Table 4.14 gives the statistics of the training data and testing data used. Table 4.15 display the class-wise DA F1-scores from the adaptation technique.

4.6.4 Use of Word Embeddings

Another experiment has been performed using MLP ann LSTM. To combat the problem of lack of annotated data for the task at hand, we are using sliding window based splits in the conversations for sequence labeling approaches like HMM and LSTM. The features used for this are simple word_embeddings of dimension 300, obtained from Google's trained *Word2vec*[12] model. For this ex-

[11] http://irshadbhat.github.io/ind-ind/

[12] https://en.wikipedia.org/wiki/Word2vec

Total Utterances	1,81,060	856
Dialog Act	#count SWBD	#count WOZ
GREETING	204	83
GREETING_EOC	2411	42
INFO_REQ	8471	248
ACTION_DIR	708	20
OO_OF_CC	103	58
EXP_PERF	149	44
ASSERT	95469	286
ANSWER	5508	251
ACK	54718	263
ACCEPT	11131	33
REJECT	326	3
MAYBE	9	1
HOLD	525	47
SNU	286	

Table 4.14 Dialog Act statistics for SWBD data and Woz Data

periment, one-leave out validation technique has been used like the earlier experiments. The results are tabulated in Table 4.16 .

4.6.5 Discussion of Results

1. From Table 4.15 it is evident that, the method of adaptation has worked very well ”Greeting”, ”Greeting_EOC”, ”Explicit_Performatives” and ”ACK”. In socio-linguistics, the expressions for these dialog acts are also well known as frozen-expressions. The set of frozen expressions is closed. Therefore, most of the translations are successful, and as the data is code-mixed, expressions like ”Hi”, ”Hello”, ”I am”, ”bye”, ”sorry”, ”thanks” etc, are already part of the CM data. So, there is no error of transliteration and translation for these cases. An example of wrong translation is in Table 4.17

2. The performance of the algorithms for INFO_REQ is not as good as that for the ”Greeting” and ”Explicit_Performatives”. One very straight forward reason is that instances of INFO_REQ are not ”frozen expressions” and there can be a wide variety of natural language utterances. Furthermore, most of the questions in the CM conversational data that fall in to the class of INFO_REQ are domain specific and therefore learning done on a model that belonged to a general model was not of much help. Secondly, the utterances that have been recognized rightly as INFO_REQ either had the question word translated right and an explicit ”?” in the utterance. But, this is not the case with a significant number of cases. The structure of a question in Telugu adds a particle ”aa” or ”a” (which are generally attached to the verb) towards the end of the question, this trait is manifested in the code-mixed utterances also. While doing a word level translation, these characteristics of

Dialog Act	SVM		HMM	
(13 tags)	unigrams	bigrams	unigrams	bigrams
GREETING	95.67	96.23	96.45	97.01
GREETING_EOC	97.80	97.85	98.35	98.62
EXP_PERF	99.13	99.26	99.38	99.67
INFO_REQ	77.67	74.62	78.87	76.25
ACTION_DIR	44.68	40.22	50.23	46.64
ASSERT	55.38	55.23	57.89	56.32
OO_OF_CC	27.82	23.67	30.28	28.94
ACK	90.68	90.32	93.68	91.32
ANSWER	65.23	59.57	76.83	74.32
ACCEPT	88.61	87.65	93.68	92.11
REJECT	98.67	96.32	98.63	97.20
MAYBE	0.0	0.0	0.0	0.0
HOLD	55.67	52.34	54.31	51.67

Table 4.15 Results of Adaptation Method

Dialog Act	MLP	LSTM
Forward_tags	69	32
Backward_tags	65	38

Table 4.16 F1-scores for MLP and LSTM

CM posed a problem and hence the system's f-score declined further. Some examples are shown in 4.17.

3. In the case of ACTION_DIR, the feature of highest importance is verb, firstly as Telugu is agglutinative in nature, in many of the cases, transliteration failed. One such example of a verb is "Iccsyai" (meaning: give me). This being the first problem, the second problem occurred while dealing with sentences like "naaku aa book issue cheyandi." (Nku buk iy cyai). In this case, "cheyandi" translated to make and the occurrence of "issue make" is very rare in an English Corpus. Moreover a lot of ACTION_DIR are pertaining to tasks like "issue", "reissue" and "return". Therefore, the words in these sentences are domain (library) specific. HMM has performed better because it has used the transition probability learned from the English corpus. Transition probability between INFO_REQ and ACTION_DIR is high.

4. ASSERT is again very domain dependent and the number of False Negatives is too high which resulted in low F1-score, due to low Recall. OFFER, COMMIT and Open_Option, for all these classes verb is the important feature and bad transliteration and translation has resulted in poor performance of the algorithms. Most of the Backward communication functions, which are basically grounding utterances are recognized correctly from the method of adaptation combined with HMM.

CM sentence	Original Translation	Obtained Translation	Annotated_Act
malli kaluddam	see you again	coriander let us meet	GREETING_EOC
naatho matladatam ide first time aa	Is this the first time you are talking to me	Me to speak the same first time the	INFO_REQ
Me library lo Karumanchi book unda?	Does your library have the book named "Karumanchi"?	Your library in the Karu good book Be	INFO_REQ
Meeku ye author ki sambandinchina books kavali?	Which author's books would you prefer?	you what author to about books need?	INFO_REQ
aite naku oka oppenham,oka love story book ichesyandi	In that case, give me Oppenham and a love story book.	However, your a Uppenham,a love story book Iccesayandi	ACTION_DIR
Data structures by Karumanchi issue cheyandi.	I would like to get Data Structures by Karumanchi issued.	Data structures by Karu good issue make.	ACTION_DIR.

Table 4.17 Translation Errors from Data

5. To conclude, though a major number of utterances are domain specific. The transition probabilities learned from the English conversational corpus has helped to attain a decent class specific F1-score.

4.6.6 Overall Error Analysis

1. An important point to note here is that, the base structure of the CM sentence in the above example is dominantly Telugu. On plugging the words translated to English, the sentence is just a sequence of English words and therefore, unigrams work the best, as they are word order independent. Because of the different word order, the n-gram based models trained on an English corpus fail on the code-mixed sentences, even after the preprocessing steps like Language Identification, Transliteration and Translation. Incorporation of transliteration with a lower error rate can increase the performance of the system.

2. Telugu is an agglutinative and morphologically rich language and when romanized, people tend to separate the morpheme from the root word . For instance, in the above example, root word is dhooram(meaning far) and the adverbial inflection is ga. Dhoormga means away. Due to separation, firstly, ga remains as it is and dhooram is translated to distance. So, the right meaning of the word is not conveyed.

3. When Telugu words in a code-mixed sentence are transliterated and then translated to English, errors can occur at both these levels, where complete loss of original word and insertion of wrong word occur respectively. This might result in not finding corresponding word embeddings. The results of experiments with MLP and LSTM, using word-embedding as features are tabulated in Table 4.16.

A Take Away Message The overall system reaches an accuracy of about 66%. Adoption of this method, only partially caters to the problem of DA tagging and does not relieve was from annotating code-mixed data manually. But, this method can be seen as a variant of "distant-supervision", and knowledge form rich English corpus can be combined with the knowledge of those models trained on relatively less amount of data, in this case the CM conversational data, to achieve better accuracies.

Observation:While doing this task, we also observed that, code-mixing is observed more frequently in utterances pertaining to some Dialog Acts than others. For example, Greetings both at the beginning and ending of the conversations were mostly monolingual (English). Also, utterances that are explicit_performatives (thanking, promising and apologizing), Accept , Reject and acknowledge were in only one language (English or Telugu). The tags that had frequent code-mixing were Assert, Information_request and Action_Directive.

Chapter 5

Conclusions and Future Work

5.1 Summary and Conclusion

Code-mixing (CM) in a language is the embedding of linguistic units such as phrases, words and morphemes of one language within an utterance of another language. This phenomenon is often observed in conversations within the bilingual and multilingual user group. The thesis focuses on two major issues: CM in discourse, on social media network, and CM in dialog. If Language Identification was the major crux of the first part, Dialog Act recognition was for the second part.

In the first part of the thesis, I have studied the problem of identification of the presence of code-mixing, and decoding the underlying information within a code-mixed utterance.
The proposed methodology focused towards identifying the juncture locations using a language identification (LID) module. Various machine learning techniques and feature selection and extraction techniques were explored in the construction of LID. The best performing LID module is based on neural networks, the feature set comprises of lexicons, prefix, suffix, infix, and other character based information. To improve upon decoding the underlying lexical information, we also developed a POS tagger to explore the grammatical structure(syntax) of code-mixed sentences.

In the second part of the thesis, I made an attempt to understand code-mixing in dialog. In a multilingual setting, dialog is a rich source of code-mixing. As a first step I have collected Telugu-English Conversational data through Wizard of Oz technique. An annotation scheme, based on DAMSL was tailored and the collected conversations were annotated. Furthermore, experiments were conducted for Dialog act recognition of utterances in the conversations through supervised and non-parametric learning algorithms using a a bag of n-grams as features. Results were reported explaining why a particular algorithm and feature set performed better than other. Subsequently, error analysis has been done. Later in the thesis, I have used an enormous conversational English corpus annotated with dialog acts to learn a model that can be used to tag code-mixed conversations. This system reaches an accuracy of about 66%. Adoption of this method, only partially caters to the problem of DA recognition and does not

relieve was from annotating code-mixed data manually. We propose that this method can be seen as a variant of Transfer Learning. Knowledge from rich English corpus can be combined with the knowledge of those models trained on relatively less amount of data, in this case the CM conversational data, to achieve better accuracies.

5.2 Future Applications

A refined understanding of the patterns of phrases in CM across a certain language pair would be helpful in the formulation of learning algorithms which will attain the prediction of location and behavior of these conjunctures in spoken utterances and scripts. This would refine the acquisition and retrieval of information across semi-formal and informal platforms. Popular virtual personal assistants, such as even SIRI cannot comprehend the presence of CM within sentences, and thus cannot handle the conversation with switch between the languages which otherwise are supported in isolation. Development of modules which comprehend and generate code-mixed sentences would impel digital personal assistants towards attaining natural interfaces. A good example of this can be understood for the web platforms offering a virtual tutoring over different subjects. The addition of a conversational interface incorporating CM across different language pairs can improve the rigor of learning of different new concepts.

Related Publications

1. **Title : Part-of-Speech Tagging for Code mixed English-Telugu Social media data**
   Authors: Kovida Nelakuditi, jittadivya.sai and Radhika Mamidi
   In the Proceedings of 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016) in Konya, Turkey.

   Abstract: Part-of-Speech Tagging is a primary and an important step for many Natural Language Processing Applications. POS taggers have reported high accuracies on grammatically correct monolingual data. This paper reports work on annotating code mixed English- Telugu data collected from social media site Facebook and creating automatic POS Taggers for this corpus. POS tagging is considered as a classification problem and we use different classifiers like Linear SVMs, CRFs, Multi- nomial Bayes with different combinations of features which capture both context of the word and its internal structure. We also report our work on experimenting with combining monolingual POS taggers for POS tagging of this code mixed English-Telugu data.

2. **Title : "Nee Intention enti?" Towards Dialog Act Recognition in Code-Mixed Conversations**
   Authors: Divya Sai Jitta, Khyathi Raghavi Chandu, Harsha Pamidipalli and Radhika Mamidi
   In the proceedings of 21st International Conference on Asian Language Processing(IALP-2017), Singapore.

   Abstract:Code-Mixing (CM) is a very commonly observed mode of communication in a multilingual configuration. The trends of using this newly emerging language has its effect as a culling option especially in platforms like social media. This becomes particularly important in the context of technology and health, where expressing the upcoming advancements is difficult in native language. Despite the change of such language dynamics, current dialog systems cannot handle a switch between languages across sentences and mixing within a sentence. Everyday conversations are fabricated in this mixed language and analyzing dialog acts in this language is very essential in further advancements of making interaction with personal assistants more natural. The problem is further compounded with crossing the script barriers in code-mixing. In this paper we take the first step towards understanding code-mixing in dialog processing, by recognizing dialog act (intention) of the code-mixed utterance. Considering the dearth of resources in code-mixed languages, we design our current system using only word-level resources such as lan-

guage identification, transliteration and lexical translation. Our best performing system is HMM based with an F-score of 76.67.

Bibliography

[1] A. Abbi. Languages of india and india as a linguistic area, 2012.

[2] P. Agarwal, A. Sharma, J. Grover, M. Sikka, K. Rudra, and M. Choudhury. I may talk in english but gaali toh hindi mein hi denge: A study of english-hindi code-switching and swearing pattern on social networks.

[3] J. Allen and M. Core. Draft of damsl: Dialog act markup in several layers, 1997.

[4] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, et al. The hcrc map task corpus. *Language and speech*, 34(4):351–366, 1991.

[5] E. Annamalai. Nativization of english in india and its effect on multilingualism. *Journal of Language and Politics*, 3(1):151–162, 2004.

[6] P. Auer. The pragmatics of code-switching: A sequential approach. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*, pages 115–135, 1995.

[7] I. M. Barredo. Pragmatic functions of code-switching among basque-spanish bilinguals. *Retrieved on October*, 26:2011, 1997.

[8] R. Begum, K. Bali, M. Choudhury, K. Rudra, and N. Ganguly. Functions of code-switching in tweets: An annotation scheme and some initial experiments. *Politics*, 329437(48854):23421, 2016.

[9] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, and M. Shrivastava. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 48–53. ACM, 2014.

[10] I. A. Bhat, M. Shrivastava, and R. A. Bhat. Code mixed entity extraction in indian languages using neural networks. In *FIRE (Working Notes)*, pages 296–297, 2016.

[11] J.-P. Blom and J. J. Gumperz. Social meaning in linguistic structure: Code-switching in norway. *The bilingualism reader*, pages 111–136, 2000.

[12] J. Bosco. Directions in sociolinguistics: The ethnography of communication, 1973.

[13] H. Bunt. A framework for dialogue act specification. *Proceedings of SIGSEM WG on Representation of Multimodal Semantic Information*, 2005.

[14] J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C. Kowtko, and A. H. Anderson. The reliability of a dialogue structure coding scheme. *Computational linguistics*, 23(1):13–31, 1997.

[15] S. Carter, W. Weerkamp, and M. Tsagkias. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215, 2013.

[16] G. Chittaranjan, Y. Vyas, K. Bali, and M. Choudhury. 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.

[17] D. Crystal. Language and the internet. cambridge university press, 2001.

[18] S. B. Dbabis, H. Ghorbel, L. H. Belguith, and M. Kallel. Automatic dialogue act annotation within arabic debates. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 467–478. Springer, 2015.

[19] S. Dowlagar and R. Mamidi. A semi supervised dialog act tagging for telugu. *ICON*, 2015.

[20] H. R. Dua. Hegemony of english: Future of developing languages in the third world. *Mysore: Yashoda Publications*, 1994.

[21] A. Ezen-Can and K. E. Boyer. Unsupervised classification of student dialogue acts with query-likelihood clustering. In *Educational Data Mining 2013*, 2013.

[22] P. N. Garner, S. R. Browning, R. K. Moore, and M. J. Russell. A theory of word frequencies and its application to dialogue move recognition. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, pages 1880–1883. IEEE, 1996.

[23] S. Gella, K. Bali, and M. Choudhury. "ye word kis lang ka hai bhai?" testing the limits of word level language identification. In *Proceedings of the Eleventh International Conference on Natural Language Processing*, pages 130–139, 2014.

[24] S. Ghosh, S. Ghosh, and D. Das. Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*, 2017.

[25] J. J. Gumperz. *Discourse strategies*, volume 1. Cambridge University Press, 1982.

[26] T. Hidayat. An analysis of code switching used by facebookers, 2008.

[27] A. Jamatia, B. Gambäck, and A. Das. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. Association for Computational Linguistics, 2015.

[28] D. Jurafsky and J. Martin. Dialog systems and chatbots. *Speech and language processing*, 3, 2014.

[29] P. Král and C. Cerisara. Automatic dialogue act recognition with syntactic features. *Language resources and evaluation*, 48(3):419–441, 2014.

[30] Y. Maschler. The language games bilinguals play: Language alternation at language game boundaries. *Language & Communication*, 11(4):263–289, 1991.

[31] P. McNamee. Language identification: a solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101, 2005.

[32] C. Myers-Scotton. *Social motivations for codeswitching: Evidence from Africa*. Oxford University Press, 1995.

[33] D. Nguyen and A. S. Doğruöz. Word level language identification in online multilingual communication. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 857–862, 2013.

[34] P. Passban, Q. Liu, and A. Way. Translating low-resource languages by vocabulary adaptation from close counterparts. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):29, 2017.

[35] D. P. Pattanayak. *Multilingualism in India*. Number 61. Multilingual Matters, 1990.

[36] S. Poplack, D. Sankoff, and C. Miller. The social correlates and linguistic processes of lexical borrowing and assimilation, 1988.

[37] K. C. Raghavi, M. K. Chinnakotla, and M. Shrivastava. Answer ka type kya he?: Learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858. ACM, 2015.

[38] N. Ramachandran. Dialogue act recognition from audio and transcription of human-human conversations. *INTERNATIONAL JOURNAL OF COMPUTER TRENDS & TECHNOLOGY*, 1(4):1946–1950.

[39] R. Redouane. Linguistic constraints on codeswitching and codemixing of bilingual moroccan arabic-french speakers in canada. In *ISB4: Proceedings of the 4th International Symposium on Bilingualism*, pages 1921–1933, 2005.

[40] K. Ries. Hmm and neural network based speech act detection. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 1, pages 497–500. IEEE, 1999.

[41] J. R. Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.

[42] R. Sequiera, M. Choudhury, and K. Bali. Pos tagging of hindi-english code mixed text from social media: Some machine learning experiments. In *12th International Conference on Natural Language Processing*, page 233, 2015.

[43] A. Sharma, S. Gupta, R. Motlani, P. Bansal, M. Srivastava, R. Mamidi, and D. M. Sharma. Shallow parsing pipeline for hindi-english code-mixed social media text. *arXiv preprint arXiv:1604.03136*, 2016.

[44] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[45] D. Surendran and G.-A. Levow. Dialog act tagging with support vector machines and hidden markov models. In *Interspeech*, 2006.

[46] D. Wang and T. F. Zheng. Transfer learning for speech and language processing. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*, pages 1225–1237. IEEE, 2015.

[47] M. Warschauer, G. R. E. Said, and A. G. Zohry. Language choice online: Globalization and identity in egypt. *Journal of Computer-Mediated Communication*, 7(4):0–0, 2002.

[48] N. Webb and M. Ferguson. Automatic extraction of cue phrases for cross-corpus dialogue act classification. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1310–1317. Association for Computational Linguistics, 2010.