

# **DIALOG ACT TAGGING FOR CODE MIXED DATA SET**

*Project Report submitted to  
Rajiv Gandhi University of Knowledge Technologies, Basar  
for the partial fulfillment of the requirements  
for the award of the degree of*

**Bachelor of Technology  
in  
Computer Science & Engineering**

**by**

**GUNDAPU SUNIL**

**ID NO: B121667**

*Under the guidance of*

**Dr. Rahika Mamidi  
Assistant Professor (Ph.D)  
IIIT Hyderabad**



**Department of Computer Science and Engineering  
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES  
BASAR – 504107, TELANGANA  
APRIL 2018**

# **DIALOG ACT TAGGING FOR CODE MIXED DATA SET**

**GUNDAPU SUNIL**

**ID NO: B121667**

*Under the guidance of*

**Dr. Rahika Mamidi**

**Assistant Professor (Ph.D)**

**IIIT Hyderabad**



**Department of Computer Science and Engineering**

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES**

**BASAR – 504107, TELANGANA**

**APRIL 2018**



**Department of Computer Science and Engineering**  
**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,**  
**BASAR**

**CERTIFICATE**

This is to certify that the Project Report entitled '**DIALOG ACT TAGGING FOR CODE MIXED DATA SET**' submitted by **GUNDAPU SUNIL, ID - B121667**, Department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, Basar; for partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering; is a bonafide record of the work and investigations carried out by him/her/them under my supervision and guidance.

**Project Supervisor**  
**Dr. Radhika Mamidi**  
**Assistant Professor (Ph.D)**  
**IIIT Hyderabad**

**External Examiner**

**Head of the Department**  
**Mr. Ranjith Kumar**  
**Assistant Professor**



**Department of Computer Science and Engineering**  
**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,**  
**BASAR**

**DECLARATION**

I/We hereby declare that the work which is being presented in this project entitled, **“DIALOG ACT TAGGING FOR CODE MIXED DATA SET”** submitted to **RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, BASAR** in the partial fulfillment of the requirements for the award of the degree of **BECHLOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING**, is an authentic record of my/our own work carried out under the supervision of **Dr. Rahika Mamidi, Assistant Professor (Ph.D form UOH), International Institute of Informational Technology Hyderabad, Telanagana.** The matter embodied in this project report has not been submitted by me/us for the award of any other degree.

Place:

Date:

**GUNDAPU SUNIL (B121667)**

## **ACKNOWLEDGEMENTS**

I take this opportunity to express my deep and sincere gratitude to my Supervisor Dr. Radhika Mamidi, Assistant Professor, IIIT Hyderabad for her valuable guidance and for giving me the opportunity to work with her. Her vast knowledge in Computational Linguistics especially in Dialog Systems helped me a lot in completing my project. Her constant encouragement, support and generous attitude were a tremendous boost for my work. And I would like to thank J. Divya Sai, for her continuous help and valuable guidance.

I would like to express my sincere gratitude to our HOD Mr. Ranjit Kumar for encouraging us. We also express our sincere thanks to our Project Coordinator Mr. Sujoy Sarkhar for providing guidance during the evaluation processes.

Foremost, I would like to express my sincere gratitude to LTRC Lab, IIIT Hyderabad faculty and staff, for their continuous support and providing all facilities to do this project.

I thank all our seniors, friends, and members of our project group in materializing this report and also for the lively support and encouragement given from time to time. I have great regard, and we wish to extend our warmest thanks to our classmates who offered constant support during six months span of dissertation work.

GUNDAPU SUNIL

## **ABSTRACT**

In a task oriented domain, recognizing the intention of a speaker is important so that the conversation can proceed in the correct direction. This is possible only if there is a way to label the utterance with its proper intent. One such labeling technique is Dialog Act (DA) tagging. The main goal of this thesis is to build a Dialog Act tagger for the Telugu English Code Mixed corpus. Dialogue Act (DA) classification plays a key role in dialogue interpretation, especially in spontaneous conversation analysis. Dialogue acts are defined as the meaning of each utterance at the illocutionary force level.

Code-Mixing (CM) is a very commonly observed mode of communication in a multilingual configuration. The trends of using this newly emerging language have 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.

Almost all standard traditional supervised machine learning approaches to classification have been applied in DA classification, from Support Vector Machines (SVM), Naïve Bayes, NLTK Classifiers, Max Entropy Classifier, Multilayer Perceptron, Conditional Random Field Classifier and Hidden Markov Model (HMM).

# TABLE OF CONTENTS

Chapter	Page No.
Certificate.....	.iii
Declaration.....	iv
Acknowledgements.....	v
Abstract.....	vi
Table of Contents.....	vii
List of Figures and Tables.....	ix
Chapter – 1: Introduction.....	1
1.1 Dialog System.....	1
1.1.1 Components in Dialog System.....	2
1.2 Dialog Acts.....	3
1.3 Objective of the Project.....	5
1.4 Organization of Thesis.....	6
Chapter – 2: Literature Review .....	7
2.1 DA Tagging using N-gram methods.....	7
2.2 DA Tagging using Syntactic features.....	7
2.3 POS Tagging of Hindi-English Code Mixed Text from Social Media.....	7
2.4 Shallow parsing pipeline for Hindi-English code-mixed social media text.....	8
2.5 Dialogue Act Modeling for Automatic Tagging & Recognition of Conversational Speech.....	8
2.6 A Dialog Act Tagger for Telugu.....	8
Chapter – 3: Dialog Corpus , Tag Set and Data Preprocessing.....	9
3.1 ASKLIB Corpus.....	9
3.2 Modified DAMSL Tag Set.....	9
3.3 Data Set of Language Identification.....	11
3.4 Data Preprocessing.....	12
3.5 Pipeline of Dialog Act System.....	13

Chapter – 4: Language Identification.....	14
4.1 Introduction.....	15
4.2 Approach.....	16
4.2.1 Conditional Random Fields.....	17
4.3 Experiments and Results .....	18
Chapter – 5: Language Transliteration and Translation.....	19
5.1 Introduction.....	20
5.2 Transliteration.....	20
5.2.1 Rule Based Approach.....	20
5.3 Translation.....	22
5.3.1 Specifying a model.....	22
5.4 Results and Overview.....	22
Chapter – 6: Learning Algorithms.....	23
6.1 Learning Algorithms Introduction.....	23
6.2 Naive Bayes Classifier.....	23
6.2.1 How it works for Dialog Act Tagging??? .....	24
6.3 Maximum Entropy Classifier.....	24
6.4 Multilayer Perceptron.....	25
6.4.1 Algorithm.....	25
6.5 Hidden Markov Model (HMM).....	26
6.5.1 How it works with ASKLIB corpus?? .....	27
6.6 Conditional Random Field Classifier.....	28
Chapter – 7: Results and Conclusion.....	29
7.1 Results before translation.....	29
7.2 Results after translation.....	29
7.3 Conclusion .....	30
7.4 Future Work.....	31
REFERENCES .....	32



## LIST OF FIGURES

Figure No.	Figure Name	Page No.
Figure 1.1	Figure showing Dialog system architecture.....	1
Figure 1.2	An example dialogs with DA tags.....	4
Fig 3.4	Pipeline of the Dialog Act Tagging.....	13
Fig 6.2	Bayes theorem for Naïve classifier.....	23
Fig 6.3	Example matrix representation of given data.....	25

## LIST OF TABLES

<b>Table No</b>	<b>Table Name</b>	<b>Page No.</b>
Table3.1	Gives the information about the number of words, dialogs .....12 and utterances per dialog present in ASKLIB corpus.	
Table3.2	Gives the information about language identification dataset.....13	
Table 5.2.1	Mapping for ambiguous characters.....22	
Table 5.4	Example for translation from code mixed to English.....25	
Table 6.4	TF – IDF formulae of Term frequency and Inverse document frequency.....26	
Table 7.1	Classifiers results before translation into English.....30	
Table 7.2	Classifiers results after translation into English.....30	
Table 7.3	Use of intention recognition in machine translation.....31	

# CHAPTER 1

## INTRODUCTION

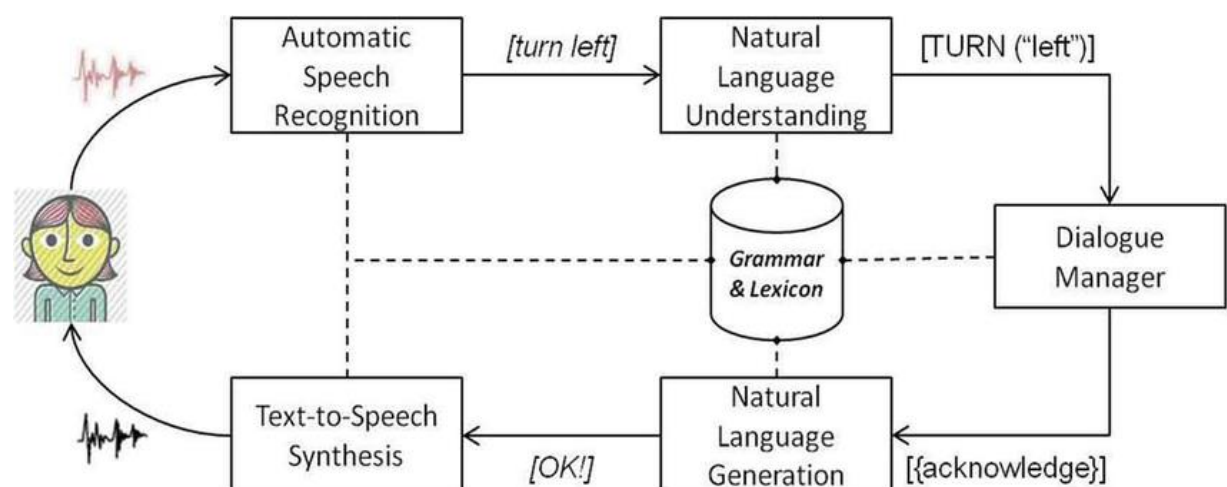
Dialog is the most fundamental and privileged arena of language. It is certainly the first kind of language we most commonly indulge in. The term 'dialog' origins from the Greek word dialog's which means conversation. The study of dialog which is the communication between two or more participants, both spoken (either face-to-face or at a distance via telephone) and written remains an interesting challenge within the field of Computational Linguistics. Dialog has been studied from a variety of perspectives, including linguistic and psychological, but it is relatively recently that this study has extended to the computational.

### 1.1 Dialog System

Conversational agents, programs with which users can engage in conversation using natural language, are an attractive concept because they can serve many purposes, from software interfaces to personal assistants to interactive game characters. Consequently a lot of research has been put into them in the last 50 years, but only with limited success so far.

Dialogue Systems is the most important application area of Computational Pragmatics which is a sub-field of Computational Linguistics. Computational Pragmatics draws on disciplines such as Pragmatics, Artificial Intelligence and Cognitive Science to study the research, development, implementation, and evaluation of dialogue systems or conversational agents.

A dialog system or conversational agent (CA) is a computer system intended to converse with a human, with a coherent structure. The crucial use of a dialog system is to convert simple yet complicated tasks from manual to automate.



**Figure 1.1** Figure showing Dialog system architecture

### 1.1.1 Components in Dialog System

1. The user speaks, the information is recognized by Automatic Speech Recognition and the input is converted to plain text by the system's input recognizer/decoder.
2. The text is analyzed by a natural language understanding unit (NLU), which may include: proper name identification, part of speech tagging, syntactic/semantic parser.
3. The semantic information is analyzed by the dialog manager that maintains the history and the state of the dialog and manages the general flow of the conversation. Usually, the dialog manager contacts one or more task managers that have knowledge of the specific task domain.
4. The dialog manager produces output using natural language generator (NLG). It could be said an NLG system is like a translator that converts a computer based representation into a natural language representation.
5. A text-to-speech (TTS) system converts normal language text into speech.

In recent times, one of the most successful applications of dialogue systems has been with in-car navigation. The machine helps the user not only by giving directions but also in making calls or choosing music. The following dialogue between User (U) and System (S) from CSLI Dialogue System taken from Pon-Barry et al (2004) for in-car devices gives a good idea of the extent of the usefulness of spoken dialogue systems.

1U: What music do you have?  
2S: I have rock, soul ...  
3U: What rock songs?  
4S: Just a second: you have to turn left at the next intersection.  
5U: OK. 6S: I have songs by U2, Cake, Green Day, Jet..etc.  
7U: Play something by them.  
8S: OK. Playing xxx.  
9U: Call John  
10S: Do you mean John Smith or John Barry?  
11U: John Smith 12S: Calling...  
13U: I need somewhere for lunch.  
14S: There is a highly rated vegetarian restaurant near here.  
15U: What other cuisines?  
16S: There are Italian, Japanese, Malaysian, Nepali, and Thai restaurants close by.

The basic dialog systems are only capable of limited domain specific conversations. Such dialog systems are based only on what is needed to fill slots and only a particular pattern is followed while responding the questions. In order to be useful for more than just form filling applications a dialog system must be able to do things like decide when a user has asked a question, made a proposal, rejected a suggestion and need to be able to ask clarification

questions and suggest plans. So the conversational agent needs sophisticated models of interpretation and generation. For better interpretation, the dialog system must understand the actual intent of what the user says. This can be done with the help of Dialog acts.

## 1.2 Dialog Acts

To make dialog managers and core components of the Natural Language Understanding and Natural Language Generation more manageable and reusable a description of human dialogs is essential.

A dialog or conversation consists of a sequence of turns, each of which consists of one or more utterance. According to an utterance in a dialog is a kind of action being performed by the speaker? Austin called these kinds of actions as speech acts.

Later on, suggested that all speech acts can be classified into three types.

1. **Locutionary act:** The utterance of a sentence with determinate sense and reference.
2. **Illocutionary act:** The making of a statement, offer, promise, etc. in uttering a sentence, by virtue of the conventional force associated with it (or its explicit performative paraphrase).

Illocutionary acts again classified into five classes.

- **Assertives:** Commit the Speaker to the truth of some proposition.

Examples: assert, predict, conclude, deduce, guess, hypothesize, stating, claiming, reporting, announcing suggest stands for the illocutionary point of assertive, which is to commit the speaker to something's being the case.

- **Directives:** Attempts to bring about some effect through the action of the Hearer.

Examples: order, command, request, ask, beg, plead, pray, entreat, invite, permit, advise, ordering, requesting, and demanding. And in special case: interrogatives

- **Commissives:** Commit Speaker to some future action.

Examples: promising, offering to do something, pledge, vow, swear, guarantee.

- **Expressives:** The expression of some psychological state.

Examples: thanking, apologizing, congratulating, condole, deplore, welcome.

- **Declarations:** Speech acts whose successful performance brings about the correspondence between the propositional content and reality (resigning, sentencing, dismissing).

Examples: pronouncing two people man and wife, christening a ship, terminating someone's employment, adjourning a meeting, appointing someone chairman.

- 3. Perlocutionary act:** The bringing about of effects on the audience by means of uttering the sentence, such effects being special to the circumstance of utterance.

A **Dialog Act** is a specialized speech act. For example, Question is a speech act, but Question\_on\_hotel is a dialog act. Dialog acts are different in different dialog systems. The process of understanding and generating the dialogs is known as dialog modeling. In dialog modeling, to understand the dialogs, speaker's intent must be recognized. The recognition of the speaker's intent is done with the help of Dialog Acts.

Dialog Acts is a tagset that classifies utterances based on pragmatic, semantic and syntactic features. The dialog acts represent the meaning of an utterance in the context of a dialog, where the context is divided into several types, with both global and local views: linguistic, semantic, physical, social and cognitive. A human conversation with its corresponding dialog acts is given

Speaker	Dialog in Telugu with gloss and its English Translation		DA tag
విద్యార్థి student Student	:	నమస్కారం సర్. hello sir Hello sir.	GREETINGS
లైబ్రేరియన్ Librarian Librarian	:	నమస్కారం. hello Hello.	
విద్యార్థి student Student	:	ఎన్ని రోజులలో ఈ పుస్తకం ఇవ్వేయాలి? how_many days_in this book give_should By when this book has to be returned?	
లైబ్రేరియన్ Librarian Librarian	:	ఒక్క నెలలో. one month_in In one month.	ANSWER
విద్యార్థి student Student	:	ధన్యవాదములు. thanks Thank you.	

**Figure 1.2** an example dialog with DA tags

The NLP problem that this thesis focuses on is the recognition of so called dialogue acts by conversational agents. Dialogue acts are the actions that speakers perform through their speech such as giving or requesting certain pieces of information and expressing emotions like gratefulness and regret. They are vital to understanding what the speaker's communicative goals and intentions are. There has been a decent amount of research on the topic of dialogue act recognition, but in general the accuracy of the developed systems is not at a satisfactory level yet.

### **1.3 Objective of the Project**

Recognizing the intention of a speaker is important so that the conversation can proceed in the correct direction. This is possible only if there is a way of labeling the utterance with its proper intent, called as “Dialog Act Tagging”. The intention of this project is to tagging a dialog act to Code – Mixed Data, by recognizing the intention of that dialogue.

### **1.4 Organization of Thesis**

The Remaining Chapters as follows;

CHAPTER 2 Describes the literature review of the project.

CHAPTER 3 Describes about data set, data preprocessing and pipeline of the project

CHAPTER4 Describes language identification.

CHAPTER 5 Describes transliteration and translation.

CHAPTER 6 Describes learning algorithms for dialog act tagging

CHAPTER 7 Describes results, conclusion and future work

## **CHAPTER 2**

### **LITERATURE REVIEW**

Earlier, research in DA tagging was limited to linguistic domain, but now with the help of statistics, machine learning and pattern matching, automated DA tagging with various DA recognition approaches have come into existence. Out of all the statistical approaches, the use of n-grams for DA tagging gave better accuracy.

#### **2.1 DA Tagging using N-gram methods**

In this project they proposed model is Dialog act tagging using N-gram methods. Some of the DA tagging methods using n-gram cues include:

- Word based DA tagging in proved that Dialogue acts can be inferred from their constituent words to an accuracy of around 50% using a very simple unigram model, implying that better performance should be possible using a more involved N-gram Markov model.
- Considers n-gram dialog act tagging method by taking n-grams of order 1 to 3.
- On the other hand, used n-grams with predictivity criterion for DA tagging which shows that instead of considering all n-grams, to take only those which surpass the threshold. But pointed out that the threshold concept works only if the corpus is large enough.

#### **2.2 DA Tagging using Syntactic features**

- Proved that both the syntactic relations as well as the semantic VerbNet-based relations included in the utterances can be extracted and added to the feature sets for the recognition task.
- Explicitly proposed global syntactic features derived from automatic parsing of the sentence which will help in DA tagging of the sentence.

#### **2.3 POS Tagging of Hindi-English Code Mixed Text from Social Media**

--- Royal Sequiera, Monojit Choudhury, Kalika Bali

Part-of-Speech (POS) tagging of Hindi-English Code-Mixed(CM) text from social media content. They are proposed extensions to the already existing approaches. For this part of speech tagging, they are also present a new feature set which addresses the transliteration problem inherent in social media. We achieve 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.



## **2.4 Shallow parsing pipeline for hindi-english code-mixed social media text**

**---- P. Bansal, M. Srivastava, R. Mamidi, and D. M. Sharma**

In this study, the problem of shallow parsing of Hindi-English code-mixed social media text (CSMT) has been addressed. They have annotated the data, developed a language identifier, a normalizer, a part-of-speech tagger and a shallow parser. For this project, first they are attempt shallow parsing on CSMT. The pipeline developed has been made available to the research community with the goal of enabling better text analysis of Hindi English CSMT.

## **2.5 Dialogue Act Modeling for Automatic Tagging & Recognition of Conversational Speech**

**--- Klaus Ries, Daniel Jurafsky, Rachel Martin**

In this paper they describe a statistical approach for modeling dialogue acts in Conversational speech, i.e., speech act like units such as STATEMENT, QUESTION, BACKCHANNEL, AGREEMENT, DISAGREEMENT, and APOLOGY. This model detects and predicts dialogue acts based on lexical, collocation, and prosodic cues, as well as on the discourse coherence of the dialogue act sequence. The dialogue model is based on treating the discourse structure of a conversation as a hidden Markov model and the individual dialogue acts as observations emanating from the model states. Constraints on the likely sequence of dialogue acts are modeled via a dialogue act n-gram.

Models are trained and evaluated using a large hand-labeled database of 1,155 conversations from the Switchboard corpus of spontaneous human-to-human telephone speech. We achieved good dialogue act labeling accuracy (65% based on errorful, automatically recognized words and prosody, and 71% based on word transcripts, compared to a chance baseline accuracy of 35% and human accuracy of 84%) and a small reduction in word recognition error.

## **2.6 A Dialog Act Tagger for Telugu**

**-- Dowlagar Suman and Radhika Mamidi**

The main goal of this thesis is to build a Dialog Act tagger for the Telugu corpus. This work focuses on discussing various n-gram DA tagging techniques so as to tag the Telugu data. In first they proposed a method that uses only n-gram karakas with back-off as n-gram language modeling technique at n-gram level and Memory Based Learning at utterance level. After that, implemented another model, that use syntactic features such as anaphora resolution, conjunct identification and using modifier modified relationship rules we automatically extract n-gram karakas. Then they apply language modeling (LM) with Hidden Markov Model (HMM) method for DA tagging.

## **CHAPTER 3**

### **DIALOG CORPUS, TAG SET AND PREPROCESSING**

#### **3.1 ASKLIB Corpus**

At present, there is no available corpus related to task oriented Telugu English code mixed dialogs. Our work started with the construction and acquisition of the dialogs in code mixed data. The corpus is developed by us using various observations, techniques and interactions.

The focus was on task oriented, domain dependent dialogs with 'Library' as the domain. We named the corpus as ASKLIB. ASKLIB consists of nearly 225 dialogs that took place between students and the librarian. This corpus is also collected by frequently visiting different libraries and observing how people interact with the librarian. The data acquisition was also done through the Wizard of Oz technique. 27 active participants were told to assume the scenario of a library and were asked to write a few generic 2 party conversations. After the corpus acquisition, we observed that the dialogs pertaining to the library domain could be broadly classified into mainly 4 types, viz. ISSUE, REISSUE, RETURN, ENQUIRY. By consider these four tags, we are again broadly classifies the tag set. Person's interaction with a librarian can result in issue, reissue or return of a book or any enquiry related to a book/the library.

#### **3.2 Modified DAMSL Tag Set**

The DA tag set is based on DAMSL (Dialogue Act Markup in Several Layers) and some domain dependent tags.

#### **DAMSL**

A major problem with speech act theory is that it attempts to capture the utterance's purpose with one label. This is a problem because utterances can simultaneously respond, promise, request and inform. DAMSL addresses this problem by allowing multiple labels in multiple layers to be applied to the utterance. Thus an utterance might simultaneously perform actions such as responding to a question, confirming understanding, promising to perform an action and informing. DAMSL is one of the domain independent tag sets used for DA Tagging. DAMSL was initially designed to be universal.

Its annotation scheme is composed off our levels (or dimensions): communicative status, information level, forward looking functions and backward looking functions. The communicative status states whether the utterance is not interpretable, abandoned or it is a self-talk. This feature is not used for most of the utterances. The information level provides an abstract characterization of the content of the utterance.

It is composed of four categories: task, task-management, communication-management and other-level. In the forward looking functions the annotators are allowed to look ahead in the dialog to determine the effect an utterance has on dialog. The backward

looking functions show the relationship between the current utterance and the previous dialogue acts, such as accepting a proposal or answering the question. DAMSL is composed of 42 DA classes.

DAMSL tags are high-level and designed to be applicable to various types of dialogs. The idea is that for a particular domain, these classes could be further subdivided into acts that are relevant to the domain, the common level of abstraction across domains, would allow researchers to share data in a way that would not be possible if everyone developed their own scheme. The focus of DAMSL has primarily been on task-oriented dialogs, where the participants are focused on accomplishing a specific task.

Tag List	Description of Tags	Example Utterance
ASSERT	Speaker makes a claim with respect to library timings	Now the time is 6.30pm.
INFO_REQUEST	Utterance is bound to provide answer related to issue of book	Do you want to issue this book?
ISSUE	Utterance intent is to issue the book	Issue this book to me
COMMIT	Committing to perform the action in future.	Ok sir, I will come tomorrow.
GREETINGS_REPLY	Utterance is replying to a greet so as to maintain the conversation	I am fine.
GREETINGS	Utterance states that the conversation is started	Good Morning sir.
ACTION_DIR	Utterance intent is to make hearer, perform an action	Give me the id card and the book.
GREETINGS_EOC	Utterance states that the conversation is completed	Thank you
REISSUE_ASSERT	Speaker makes a claim while returning the book	I am extending the book's due date.
ACCEPT	Utterance shows speakers agreement to the proposal or claim	Ok sir.
ACCEPT_ACKNOWLEDGE	Utterance indicates speaker has understood.	Ok sir, I will return this book in a month.

**Table3.1** gives the information about the number of words, dialogs and utterances per dialog present in ASKLIB corpus. The information the percentage of tag in the corpus, description of each tag with an example (written originally in Telugu) translated to English.

### 3.3 Data Set of Language Identification

Word	Pos Tag	Language
1992	G_X	univ(Universal)
chance	G_N	en(English)
andhariki	PSP	te(Telugu)
radhika	G_N	ne(Named Entity)

**Table3.2** Gives the information about the example of language identification dataset

In this dataset, we have approximately 33,000 code mixed words. These words are manually annotated by POS tag and 4 language identification classes.

The classes, we are used for language identification following,

1. Telugu(te)
2. English(en)
3. Universal(univ)
4. Named Entity Recognition(ne)

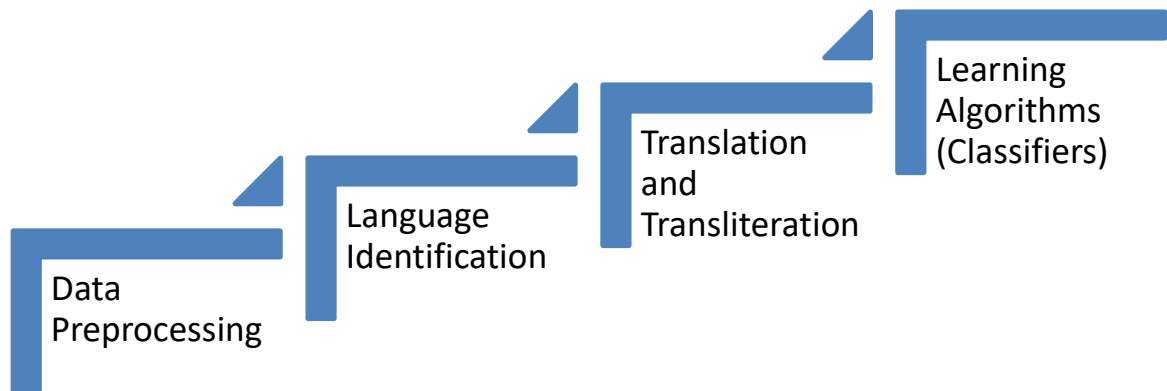
### 3.3 Data Preprocessing

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.

#### Example:

- **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.

### 3.4 Pipeline of Dialog Act System:



**Fig 3.4** Pipeline of the Dialog Act Tagging

## CHAPTER 4

### LANGUAGE IDENTIFICATION

#### 4.1 Introduction

Language Identification is the process of dividing words into one of the mentioned classes. A Conditional Random Fields (CRF) model is implemented for this purpose. The features are to be derived from the word itself, as any other tool might not function on this data at this stage. We mainly consider character sequences of the word as features for the system. The challenging part is when a word might belong to two different classes.

##### Example:

Input: plz watch it nd share chusaka nenu cheppanavasarm le meere share chestharuuu

Output: plz-en watch-en it-en nd-en share-en chusaka-en nenu-en cheppanavasarm-en  
le-te mere\_en share-en chestharuuu-en

#### 4.2 Approach

Language Identification is the process of dividing words into one of the mentioned classes. Initially we are taking this problem same as the pos tagging problem. Hidden Markov Model (HMM) is used for that. But comparatively another which Conditional Random Field gives more accuracy rather than the HMM. A CRF model is implemented for this purpose with the following features:

- lexical feature:
  - word
- sub-lexical features:
  - Prefix, suffix character strings
  - n-grams of word
  - prefix, Suffix character strings of neighboring words
- other features:
  - length of the word
  - neighboring words
  - pos tag of word
  - pos tag of previous word

### **4.2.1 Conditional Random Fields**

Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting structured data, such as sequences, trees and lattices. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences. Conditional random fields are arbitrary undirected graphical models trained to maximize the conditional probability of the desired outputs given the corresponding inputs. The primary advantage of CRFs over hidden Markov models (HMM) is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Additionally, CRFs avoid the label bias problem, a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models. CRFs outperform both MEMMs and HMMs on a number of real-world tasks in many fields.

### **4.3 Experiments and Results**

Language Identification: Using different combinations and variations of these features, various templates were experimented upon these CRF algorithm.

#### **Template File Features:**

previous word|current word|next word|pos tag|Is it number|Is it special character|suffix|prefix

#### **Results:**

Precision: 0.887695092

Recall: 0.899182781

Accuracy: 0.888173248

## CHAPTER – 5

### LANGUAGE TRANSLITERATION AND TRANSLATION

#### 5.1 Introduction

In addition to the dearth of code-mixed data, the problem of learning reliable word embedding's further compounded with spelling variations due to Romanization of Telugu words. In my approach, we investigate if translation could be of any use. The output from LID system is used for this purpose. At this stage, we have two options; either translates everything in to English or everything into Telugu. We investigate either ways. We have used Indictrans for transliteration and Google translate for lexical (individual word) translation.

#### 5.2 Transliteration

Transliteration simply means conversion of a text from one script to another. Machine transliteration is the computer automated process of transcribing a character or word from one language script to another. Machine transliteration can play an important role in natural language application such as information retrieval and machine translation, especially for handling proper nouns and technical terms, cross-language applications, data mining and information retrieval system.

For language transliteration we are using “INDIC-TRANS”. After language identification, we are going to transliterate it into Telugu language and script too. The tool aims on adding a state-of-the-art transliteration module for cross transliterations among all Indian languages including English and Urdu. The module currently supports the almost all Indian languages. And this tool gives more accurate results.

##### 5.2.1 Rule Based Approach

A rule-based transliteration system uses character mappings defined between two scripts. There are five types of character mappings possible between two natural language scripts: One-to-One, One-to-Many, Many-to-One, One-to-None and None-to-One. A simple One-to-One character mapping would lead to a more accurate and easy to develop system. However, natural languages are inherently ambiguous. There are many cases of ambiguity in orthographic representation of languages world over. Distribution of different types of character mappings is not generally uniform, though. The distribution is usually skewed; unambiguous characters occur more often than their ambiguous counterparts. Nonetheless, to resolve the ambiguous mappings, heuristics are designed which take the surrounding context of an ambiguous letter into consideration.

- **Benefits**



1. Rule-based transliteration systems between Indic scripts are essentially trivial. Indic scripts have a special property that their phonemes are one-to-one aligned between their Unicode tables. It essentially means that the transliteration between any two Indic scripts can be achieved by merely using their unicode tables.
2. Rule-based systems do not require any kind of training data to develop the system.

#### ▪ Challenges

1. Requires domain expertise: To develop a rule-based system between two natural language scripts a domain expert is required.
2. Missing phonemes in Indic scripts: The major concern with rule-based system for Indic-to-Indic transliteration is the missing phonemes in Indic scripts. For example, in Tamil there are no characters for d, dh, b, bh etc. d is pronounced as t, dh as th, b and bh are pronounced as p. Similarly, in Bengali there is no character for v, it is pronounced as b.
3. Ambiguous Character Mappings: Another major issue with rule-based system for is ambiguous character mappings between two natural language scripts. For example, Tamil has the following ambiguous characters:

Character	Mapping
k	k, kh, g, gh, h
C	c, ch, j, jh, s
T	t, th, d, dh
T	T, Th, D, Dh
p	p, ph, b, bh

**Table 5.2.1** mapping for ambiguous characters

#### Example:

It takes, source language = “english” and target language = “english”

Input: Vijay\_ne inka\_te hindhi\_te kudha\_te unnaaii\_te in\_en our\_en library\_en

Output: Vijay\_ne ఇంకా\_te హిందీ\_te books\_en కుదా\_te ఉన్నాయి\_te in\_en our\_en library\_en.

In the above example, it takes only telugu language(te) words only. Those words transliterate into telugu script. The main advantage of this tool is, If the input text contains repeating words, which raw text generally does, make sure to set **build\_lookup**. As the name indicates this builds lookup for transliterated words and thus avoids repeated transliteration of same words. This saves a lot of time if the input corpus is too big.

## 5.3 Translation

In translation phase, we are going to translate the Telugu words into English language. For this work, we are using Google Cloud Translator API. One of the best API to give more and more accurate results.

### 5.3.1 Specifying a model

By default, when you make a translation request to the Google Cloud Translation API, your text is translated using the Neural Machine Translation (NMT) model. If the NMT model is not supported for the requested language translation pair, then the Phrase-Based Machine Translation (PBMT) model is used.

#### Using the model parameter

You can specify which model to use for translation by using the model query parameter. Specify base to use the PBMT model, and “nmt” to use the NMT model. If you specify the NMT model in your request and the requested language translation pair is not supported for the NMT model, then the PBMT model is used.

#### Example:

**Input:** Vijay ఇంకా హిందీ books కుదా ఉన్నాయి in our library

**output:** Vijay ,Hindi books are in our library

## 5.4 Results and Overview:

The overview of this phase is, first it will take the language identified sentence from previous phase. Next, in that sentence it will find out the Telugu words and convert into Telugu script. Now the sentence contains combination of Telugu script and English script words. This sentence will be given to Google cloud translator, it will convert this sentence into total English language and script with meaningful. For this sentence we are detecting the intention and tagging with the suitable dialog act tag.

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 ?

**Table 5.4** Example for translation from code mixed to English

## CHAPTER – 6

### LEARNING ALGORITHMS

#### 6.1 Learning Algorithms Introduction:

We have explored multiple learning algorithms namely, SVM, KNN, HMM, Naive Bayes, MLP, CRF and have tried to understand how differently code-mixed data needs to be processed as compared to monolingual data. We began with exploring traditional machine learning algorithms for code-mixed data. We model the DA Tagging problem as a sequence labeling problem and therefore, used an HMM.

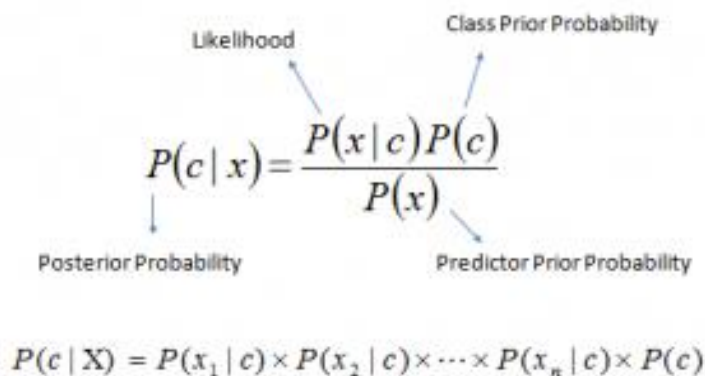
We tried all algorithms in both ways before translation and after translation

#### 6.2 Naive Bayes Classifier:

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability  $P(c|x)$  from  $P(c)$ ,  $P(x)$  and  $P(x|c)$ . Look at the equation below:



The diagram illustrates the Bayes theorem for a Naive classifier. It features the equation  $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$  with arrows pointing from descriptive labels to the corresponding parts of the formula. 'Likelihood' points to  $P(x|c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c|x)$ , and 'Predictor Prior Probability' points to  $P(x)$ . Below this, the expanded formula is shown:  $P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$ .

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability

Likelihood

Class Prior Probability

Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

**Fig 6.2** Bayes theorem for Naïve classifier

### 6.2.1 How it works for Dialog Act Tagging???

I am considering the dialog act tagging problem same as “Text Classification” problem. In this, we are taking each sentence as a document, for that calculate the conditional probability for each class.

#### Input :

- A sentence/utterance in a conversation  $S$
- Fixed set of dialog acts (Considering as classes)  $C = \{c_1, c_2, \dots, c_n\}$
- A training set of  $m$  hand labeled sentences  $(s_1, c_1), (s_2, c_2), \dots, (s_m, c_m)$

#### Output:

- A learned classifier  $Y: S \rightarrow C$
- Gives /Predict a dialog act tag for given utterance.

### 6.3 Maximum Entropy Classifier (conditional exponential classifier)

The **maximum entropy classifier** converts labeled feature sets to vectors using encoding. This encoded vector is then used to calculate weights for each feature that can then be combined to determine the most likely label for a feature set.

For a task like sentiment analysis we can follow the same procedure. We will have as the input a small amount of input data. These will be used to train the Logistic Regression classifier. The most important task then, is to select the proper features which will lead to the best classification. Almost everything in the text sentence can be used as a feature.

For this analysis usually the occurrence of (specific) words is used, or the relative occurrence of words (the word occurrences divided by the total number of words).

Initially, we have to fill in the  $X$  and  $Y$  matrices, which will serve as an input for the gradient descent algorithm and this algorithm will give us the resulting feature vector  $\theta(\phi)$ . With this vector we can determine the class of other text documents.

As always  $Y$  is a vector with  $n$  elements (where  $n$  is the number of sentences). The matrix  $X$  is a  $n$  by  $m$  matrix; here  $m$  is the total number of relevant words in all of the sentences.

Each row of the  $X$  matrix contains all of the data per sentence and each column contains the data per word. If a sentence does not contain a specific word, the corresponding column will contain a zero.

### Example X matrix:

$$X = \begin{bmatrix} & \text{publication} & \text{such} & \text{brilliant} & \text{a} & \dots & \text{beautiful} & \text{edition} & \text{virtually} & \text{unreadable} \\ 1 & 0 & 1/28 & 0 & 2/28 & \dots & 1/28 & 1/28 & 0 & 0 \\ 1 & 0 & 0 & 1/16 & 1/16 & \dots & 0 & 0 & 0 & 0 \\ 1 & 1/30 & 0 & 0 & 2/30 & \dots & 0 & 1/30 & 1/30 & 1/30 \end{bmatrix}$$

**Fig 6.3** Example matrix representation of given data

From this training\_set, we are going to generate a words\_vector. This words\_vector is used to keep track to which column a specific word belongs to. After this words\_vector has been generated, the X matrix and Y vector can filled in.

Once the training set has been converted into the proper format, it can be feed into the train method of the MaxEnt Classifier. Once the training of the MaxEntClassifier is done, it can be used to classify the review in the test set.

## 6.4 Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feed forward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

### 6.4.1 Algorithm:

1. Initially need to load **MLP Classifier**.
2. we create an object called 'mlp' which is a MLP Classifier. We set hidden\_layer\_size to (10) which means we add one hidden layer with 10 neurons. Then we set solver as 'sgd' because we will use Stochastic Gradient Descent as optimizer. Then we set learning\_rate\_init to 0.01, this is a learning rate value (be careful, don't confuse with alpha parameter in MLPClassifier). Then the last, we set 500 as the maximum number of training iteration.
3. Train the model: To train the model, we have to convert the train data into tf-idf vector.

**TFIDF:** short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection.

**Term frequency:** The **term frequency**  $tf(t,d)$ , the simplest choice is to use the raw count of a term in a sentence, i.e. the number of times that term  $t$  occurs in sentence  $d$ . If we denote the raw count by  $f_{t,d}$ , then the simplest  $tf$  scheme is  $tf(t,d) = f_{t,d}$ .

**Inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

Then  $tf-idf$  is calculated as: **Term Frequency \* Inverse document frequency**

TF-IDF is calculated for training data, based on training data, we will train the MLP classifier.

**Recommended TF-IDF weighting schemes**

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Table 6.4 TF – IDF formulae of Term frequency and Inverse document frequency

4. Test the model for the test data.

## 6.5 Hidden Markov Model (HMM):

HMM inherently takes sequence in to account. The way I have formulated the problem is that, an entire conversation - sequence of utterances is a single data point for me. I have modeled the problem as analogous to part of speech tagging problem. So, in our case the observations are utterances and the hidden states are dialog acts. Now calculate the emission and transition probabilities accordingly.

Transition probability:  $P(\text{tag}(\text{prev}) | \text{tag}(\text{current}))$

Emission Probability:  $P(\text{utterance} | \text{tag})$

Emission probability can be calculates as product of individual words given the tag or n-grams also can be considered.

### 6.5.1 How it works with ASKLIB corpus??

We know that the task oriented dialogs have conversational flow which leads to accomplishment of the required task.

This flow can be captured by using the forward backward algorithms. We have used HMM as one of the forward backward algorithms to capture that flow. As HMM works on the principle of the Markov model, i.e. the present state of the sentence is predicted by observing its previous states. We modified it as; the present DA tag of the utterance is predicted by observing the previous 'k' DA tags in the dialog. The Viterbi algorithm is a dynamic programming algorithm and is the most common decoding algorithm used for HMMs, whether for part-of-speech tagging or for speech recognition. We used this for the DA tagging of the sentences.

In ASKLIB corpus there might be situations where the implicit meaning of the speaker is different from what the utterance literally conveys i.e. the words in an utterance might point to a particular DA tag based on the LM output but while considering the context they might point to another DA tag. An example is shown in table 6.5.

DA tagging without context handling	DA TAG
Student: I would like to issue this book	ISSUE
Librarian: This book has already been issued to you. Would you like to reissue the book?	REISSUE_INFO_REQUEST
DA tagging with context handling	DA TAG
Student: I would like to issue this book	REISSUE
Librarian: This book has already been issued to you. Would you like to reissue the book?	REISSUE_INFO_REQUEST

**Table 6.5** An example which proves that context handling is necessary.

After observing the 2 utterances, one can say that the student wants to reissue the book. On the other hand, if we use only the combination of LMs, the information conveyed by the librarian will be missed as LMs modeled here work at utterance level only. Hence without the context i.e. only at LM level, the DA tag given will be ISSUE. But with the help of context viz. with the librarian's utterance we can say that the appropriate DA tag is REISSUE.

$$\hat{Q} = \underset{Q}{\operatorname{argmax}} P(Q)P(S|Q)$$

$$\hat{Q} = \underset{Q}{\operatorname{argmax}} \prod_{i=1}^n P(Q_i|Q_{i-1}) \frac{P(Q_i|S_i)}{P(Q_i)}$$

Viterbi algorithm consists of the emission matrix and the transition matrix. The emission matrix values are obtained from the LM combination method mentioned above and the transition matrix values are obtained from conversation flow of the training data. The Viterbi algorithm is run two times. Firstly, the dialog is given to the Viterbi algorithm and the output with DA tag and its probabilities are taken. During the second run, the dialog is reversed and given to Viterbi algorithm and the corresponding DA tag for utterance and its probabilities are taken. Then the two probabilities obtained for each utterance are compared and the best tag which has the greater probability is given to the utterance.

This approach helps in complete coverage of the utterance i.e. this approach not only considers previous tags but indirectly it takes next tags to predict the output of the current utterance.

## **6.6 Conditional Random Field Classifier:**

Linear Conditional Random Field is a simple, customizable, and open source implementation of CRF for segmenting/labeling sequential data. CRF is designed for generic purpose and will be applied to a variety of NLP tasks, such as Named Entity Recognition, Information Extraction and Text Chunking.

Both the training file and the test file need to be in a particular format for CRF to work properly. Generally speaking, training and test file must consist of multiple tokens. In addition, a token consists of multiple (but fixed-numbers) columns. The definition of tokens depends on tasks; however, in most of typical cases, they simply correspond to words. Each token must be represented in one line, with the columns separated by white space (spaces or tabular characters). A sequence of token becomes a sentence. To identify the boundary between sentences, an empty line is put. In training data the last column represents a true answer tag which is going to be trained by CRF.

Before going to train the data, we should extract the features from training utterances. Based on these utterances only it will calculate the most likelihood ratio and posterior probability. By training the model it will generate a CRF model file. We can test the test\_data by using this CRF model file.



## CHAPTER – 7

### RESULTS AND CONCLUSION

We have applied different learning algorithms on my Telugu-English code mixed dataset. We are checked the results before and after the translation. Both strategies, gave different results. From all of the algorithms Hidden Markov Model and Multilayer Perceptron produce best results.

#### 7.1 Results before translation:

Learning Algorithm	Translation	Precision	Recall	F - Score
Naïve Bayes Classifier	No	0.512890192	0.441782901	0.592592592593
Max Entropy Classifier	No	0.511987010	0.408917010	0.641975308642
Conditional Random Field	No	0.639120446	0.58686655	0.765432099298
Multilayer Perceptron	No	0.727819801	0.692810193	0.802469136186
Hidden Markov Model	No	-	-	0.771290130912

**Table 7.1** Classifiers results before translation into English

#### 7.2 Results after translation

Learning Algorithm	Translation	Precision	Recall	F - Score
Naïve Bayes Classifier	Yes	0.569101249	0.489011239	0.660194174757
Max Entropy Classifier	Yes	0.478290188	0.404192349	0.563106796117
Conditional Random Field	Yes	0.473727423	0.294912747	0.524271845902
Multilayer Perceptron	Yes	0.589172034	0.510452189	0.662435678922
Hidden Markov Model	Yes	-	-	0.712145670191

**Table 7.2** Classifiers results after translation into English

#### 7.3 Conclusion:

Hidden Markov Model and Multilayer Perceptron MM performs better comparatively, the reason for this being the inclusion of contextual information. The translation has not added significant value to the scores, the reasons for which are discussed here. 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 vectors. There are no strict guidelines that need to be adhered while Romanizing Telugu. This leads to spelling

variations. For example, consider the words: ‘vachaadu’ and ‘vachadu’ (wx form: vaCADu, meaning: ‘he came’). Another most commonly found such variations are the presence and absence of ‘h’. This heavily depends on idiolect as compared to merely geographical or societal factors.

Translation into Telugu was comparatively poor. Romanized Telugu words do not adhere to specified rules, hence introducing errors in translation. While translating to English, words identified as English remain the same, thus eliminating translation error. Lexical translation has been effective in dealing with CM as shown by where translation improved accuracy of their question classification system by 5%. We have not used topic-specific features to maintain the generalizability of the system across domains.

Our best performing system is HMM based which has given an average F-score of 76.67, averaged over Forward and Backward functions.

## 7.4 Future Work

The scope of future directions for this work is as follows,

In this work, we only focused on n-gram dialog act tagging methods. Apart from these, there are several statistical methods developed tagging techniques such as graph based models etc. We will try to observe the behavior of other tagging techniques on our corpus.

DA tagging can be explored in machine translation system. By recognizing the intent of the utterance, one can predict the actual target translation given its DA Tag. An example showing the output of an MT system using DA tagging for intent recognition is given in table 6.1.

Source Language (Telugu)	Pichuka Meedha Brahmastram.
English Translation (Literal)	Using Brahmastram on a sparrow.
English Translation with Intent Recognition	Using excessive force on a weak opponent.

**Table 7.3** Use of intention recognition in machine translation.

## REFERENCES

- [1] S. Dowlagar and R. Mamidi. A semi supervised dialog act tagging for telugu. ICON 2015 : 12th International Conference on Natural Language Processing, 2015.
- [2] T. Hidayat, “An analysis of code switching used by facebookers,” 2008.
- [3] 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. ACM, 2015, pp. 853–858.
- [4] J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Toward conversational human-computer interaction. AI magazine, 22(4):27, 2001.
- [5] A. Bharati, R. Sangal, D. M. Sharma, and L. Bai. Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages. LTRC-TR31, 2006.
- [6] D. Jurafsky and J. H. Martin. Speech & language processing. Pearson Education India, 2000.
- [7] P. Král and C. Cerisara. Dialogue act recognition approaches. Computing and Informatics, 29(2):227–250, 2012.
- [8] P. Liu, Q. Hu, J. Dang, D. Jin, and J. Cao. Dialog act classification in chinese spoken language. In Machine Learning and Cybernetics (ICMLC), 2013 International Conference on, volume 2, pages 516–521. IEEE, 2013.
- [9] A. PVS and G. Karthik. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. Shallow Parsing for South Asian Languages, 21, 2007.
- [10] 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.
- [11] 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, vol. 26, no. 3, pp. 339–373, 2000.
- [12] A. Jamatia, B. Gambhark, and A. Das, “Part-of-speech tagging for code - mixed english-hindi twitter and facebook chat messages.” Association for Computational Linguistics, 2015
- [13] Wikipedia. Dialog system — wikipedia, the free encyclopedia, 2018.