Analysis and Generation of Code-mixed Data

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

Master of Science in
Computational Linguistics
by Research

by

Dama Sravani
2018114020
dama.sravani@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **"Analysis and Generation of Code-mixed Data"** by **Dama Sravani**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Radhika Mamidi

To Dama Suneetha

Acknowledgments

I express my gratitude to my advisor, Dr. Radhika Mamidi, for her constant support and encouragement towards my research interests. She was always open to my ideas and provided guidance in the right direction. Whenever I visited her office with multiple doubts and questions, I always came back with more evident solutions and a more focused mind. I am grateful to her for providing me with the freedom to discuss my concerns and seek her guidance.

I am grateful to my amma for trusting and supporting me in all my decisions without questioning them. Her belief in me has been a great source of inspiration, motivating me to stay strong and responsible. I am also thankful to my mamaiah for patiently listening to all my personal and academic thoughts, and for pushing me beyond my limits every time.

I express my heartfelt thanks to Sukesh Davanthapuram for having immense faith and trust in me. He believed that I had the potential to achieve anything I set my mind to. During times when I doubted myself, he helped me overcome those thoughts and boosted my confidence.

Thank you, Dr. Manish Shrivastava, for providing me with a unique perspective on NLP systems, which allowed me to think more deeply about them, that helped me to delve beyond surface-level analysis. I would also like to thank Dr. Dipti Misra Sharma for teaching me various linguistics courses that helped me build a strong foundation in this field.

I want to extend my thanks to Prashanth Kodali for supporting me during the later stages of my research. The discussions we had proved to be immensely helpful in refining my research ideas. Thank you, Lalitha Kameswari, for providing me with my first research experience. Collaborating with you on our paper has given me hands-on experience on how to approach research problems and the procedures to follow, from the initial stages to submitting a paper. This first paper will always hold a special place in my heart.

Thank you, Manaswini, for always being just one call away. I cannot express enough how grateful I am for your constant support and positive energy, especially during the last phases of my research. You were always my go-to person for any doubt, and I would call you to talk over things, regardless of whether or not you had the context. Thank you, Priya, for also being the person I could talk whatever I have on my mind.

Thank you, Adithya, for always being supportive, both emotionally and technically. I cannot recall a single instance where you said no to any help I asked for. Thank you, Kishan, for giving me a different perspective on my thoughts. I am also grateful for the instances when you inspired me, as they

definitely pushed me to exceed my limits. Nihar, your emotional support has been invaluable to me. Lastly, a special thanks to all my close friends, Sexysalsanskarisamosa, BB and GG, for making my college life more memorable.

Last but definitely not least, thank you, IIIT Hyderabad, for providing me with an environment where I could constantly learn and grow. You have shaped me into a different person, teaching me to approach and learn new things.

Thank you to every one from the bottom of my heart to all the people who helped me in the research journey.

Abstract

Code-mixing (CM) is a common linguistic phenomenon, especially in multilingual communities, where speakers blend two or more languages or dialects in a single sentence or conversation. With English being the second language in India, code-mixing has become a predominant practice, particularly in urban areas. It is common for a Hindi or Telugu speaker to switch to English in the same sentence or utterance. The surge in code-mixing can be attributed to the rise of social media, which has provided a platform for people from diverse linguistic backgrounds to communicate. In such contexts, code-mixing has become an effective way to express oneself using words and phrases from multiple languages.

Code-mixing has been of interest to sociolinguists for a long time. Studying its functional form from a sociolinguistic perspective has become a significant phenomenon along with its linguistic structures. In a multilingual country like India, politicians use language and linguistic mechanisms to establish relationships with people. Our study focuses on using code-mixing in Telugu political speeches to understand the factors responsible for their usage levels in various social settings and communicative contexts. As part of our analysis, we have compiled a set of rules that capture dialectal variations between the Standard and Telangana dialects of Telugu.

Building NLP systems for code-mixing can help analyze sentiment and conversations in social media. It can also facilitate machine translation, helping people who converse in multiple languages to communicate more efficiently. The informal nature of code-mixed text presents a significant challenge for NLP systems. It results in a wide range of language variations that include non-standard abbreviations, contracted spellings, and informal grammatical structures. These challenges collectively result in a scarcity of quality code-mixed data that can be used for building NLP systems.

We propose a hybrid methodology to generate code-mixed text. The proposed hybrid methodology combines rule-based and statistical approaches to convert a monolingual Telugu sentence into a code-mixed sentence in English and Telugu dialects.

Due to the complexity of the proposed hybrid approach, we propose a fine-tuned neural machine translation method to generate high-quality code-mixed sentences using minimal gold-standard corpus. We use filters from the gold corpus to ensure that the synthetic training data for the models is of high quality, resulting in improved performance of the neural machine translation models. Considering the recent success of pre-trained models such as mT5 and mBART, we fine-tuned these models. Moreover,

our approach outperforms the current systems trained on synthetic data for code-mixed generation in Hindi-English. Apart from Hindi-English, the approach performs well when applied to Telugu, a low-resource language, to generate Telugu-English code-mixed sentences. It is crucial to investigate the effectiveness of filtering techniques for generating high-quality code-mixed data, especially in the case of low-resource languages. Moreover, exploring the application of one-shot and zero-shot learning techniques to determine whether the models are trained to generate code-mixed sentences in general or are specific to the languages they are trained on is also a promising future direction.

Contents

Chapter Page

1 Introduction . . . 1
1.1 Language and Natural Language Processing (NLP) . . . 1
1.2 Code-mixing . . . 2
1.2.1 Challenges in code-mixed data . . . 3
1.3 Motivation . . . 3
1.4 Key contributions of Thesis . . . 4
1.5 Thesis Overview . . . 4

2 Related Work . . . 6
2.1 Code-mixing: Previous research works . . . 6
2.2 Sociolinguistic Perspective of Code-mixing . . . 7
2.3 Code-mixed Text Generation . . . 7
2.3.1 Linguistic Theory Based Approaches . . . 7
2.3.2 Language Models based approaches . . . 8
2.3.3 Neural Machine Translation(NMT) approaches . . . 9

3 Political Discourse Analysis: A Case Study of Code Mixing and Code Switching in Political Speeches . . . 10
3.1 Introduction . . . 10
3.2 Dataset and Annotation . . . 11
3.2.1 Dataset collection . . . 11
3.2.2 Annotation . . . 12
3.2.2.1 Guidelines to handle dialectal level code-mixing . . . 12
3.2.2.2 Guidelines to handle language level code-mixing . . . 14
3.2.2.3 Guidelines to handle code-switching . . . 14
3.3 Observations and Results . . . 14
3.4 Conclusions . . . 16

4 Integrating Linguistic Rules and Statistical Methods for Code-Mixed Text Generation . . . 17
4.1 Introduction . . . 17
4.2 Methodology . . . 18
4.2.1 Sentence level code-mixing . . . 18
4.2.2 Word level code-mixing . . . 18
4.2.2.1 Language level: . . . 18
4.2.2.2 Dialectal level code-mixing . . . 19

4.3 Limitations of the approach . . . 20
4.4 Conclusion . . . 20

5 Code-mixed Text generation using Filtering of Synthetic Data in Neural Machine Translation . 21
5.1 Introduction . . . 21
5.2 Architecture . . . 22
5.2.1 Filtering Mechanism . . . 23
5.2.1.1 Regression . . . 23
5.2.1.2 Probabilistic methods . . . 25
5.2.2 Data preparation for Seq2Seq Models: . . . 26
5.2.3 Training Seq2Seq Models . . . 27
5.3 Experiments and results . . . 28
5.3.1 Experimental Setup . . . 28
5.3.2 Results . . . 28
5.4 Conclusion . . . 30

6 Conclusion and Future Work . . . 31
6.1 Summary . . . 31
6.2 Challenges . . . 32
6.3 Future Work . . . 32

Appendix A: . . . 34

Bibliography . . . 41

List of Figures

Figure		Page
2.1	Parse-trees of (a) sentences [2E] and (b) [1S], and (c) of [2CM] according to the ECT .	8
2.2	The CM Generation Process in GCM	8
5.1	Methodology for Code-mixed Text Generation	22
5.2	Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning.	24
5.3	Framework for Multilingual Denoising Pre-training(mBART)	27

List of Tables

Table		Page
3.1	KCR Speech Statistics (E-English, H-Hindi/Urdu)	15
3.2	CBN Speech Statistics	15
5.1	A sample from HINGE Dataset	23
5.2	A sample from Telugu-English Dataset	24
5.3	Evaluation of Regression Models for Hindi-English	25
5.4	Evaluation of Regression Models for Telugu-English	25
5.5	Performance of Hindi-English code-mixed generation models	29
5.6	Performance of Telugu-English code-mixed generation models	29

Chapter 1

Introduction

The objective of this thesis is to investigate the phenomenon of code-mixing from a sociolinguistic perspective and develop a system for generating code-mixed texts in Indian languages. In this chapter, we provide an overview of Language and Natural Language Processing (NLP) and an introduction to the concept of code-mixing. We also discuss the motivation behind this research, the key contributions and the overview of how the thesis is organised.

1.1 Language and Natural Language Processing (NLP)

Language is a crucial tool for communication between individuals and groups, and it is characterized by a complex structure that enables us to convey meaning through the usage of sounds, words, grammar, and syntax. The intricate nature of language makes it possible for us to express a wide range of thoughts and ideas, and it plays an important role in communication and social transfer. Using language, we can express our opinions, share knowledge, and exchange information. However, effective communication requires more than just linguistic competence; it also involves understanding the social and cultural context in which language is used.

Natural Language Processing (NLP) is a research area that aims to enable machines to interpret, understand, and generate human language. However, human language's complex and ambiguous nature presents a significant challenge for NLP researchers. Despite this, NLP has already found numerous applications, such as voice assistants, dialogue and sentiment analysis, machine translation, etc. NLP technology is constantly evolving and becoming more sophisticated. It can potentially transform how humans interact with machines and could lead to the development of even more advanced and intuitive technologies.

1.2 Code-mixing

Code-mixing(CM) is a phenomenon of mixing two or more languages in an utterance of a speech or text [12]. It is a prevalent form of communication in multilingual communities. Users often mix languages to express their thoughts and emotions more effectively or to convey cultural and social identity [54] [29]. With the advent of social media, code-mixing has become even more widespread on platforms like Facebook, Twitter, and Reddit. The extensive use of code-mixing has opened up several interesting research directions in linguistics, computational linguistics, and natural language processing.

There are majorly two different types of code-mixing:

- **Inter-Sentential Code-mixing:** It occurs between two or more sentences. Speakers might alternate between different languages while speaking different sentences.

  Example: I'm going to the super market, *Tum bhi aathe hai kya?*
  Translation: I'm going to the super market, Do you also want to come?

- **Intra-Sentential Code-mixing.** It occurs within a sentence or phrase. The speaker might use words or phrases from different languages with a sentence.

  Example: *nenu* school *ki velthunaanu*
  Translation: I'm going to school

Inter-Sentential code-mixing is also referred to as code-switching(CS). Code-mixing is used as a broader term for the usage of different languages or different varieties within an utterance, including code-switching as one of its terms. Despite these differences in meaning, the two terms are frequently used interchangeably in research and academic literature. We will also follow the same convention and use code-mixing to mean both the terms.

Matrix language and embedded language are key terms in code-mixing. The matrix language dominates the mixing, providing structure and grammar, while the embedded language contributes lexical items.

Code-mixing finds implications in both social and linguistic contexts. Socially, it is used to express cultural identity and group membership in multilingual communities. From a linguistic standpoint, it provides valuable insights into how different languages and language varieties interact.

Code-mixing is increasingly applied in various areas of NLP, with machine translation being one of the most important. It is also used in speech recognition, sentiment analysis, and language modelling. In the context of social media platforms, analysing the code-mixed text can offer valuable insights into language usage patterns, sentiment towards specific topics, and community interactions.

1.2.1 Challenges in code-mixed data

Code-mixing, which is the use of multiple languages or language varieties in a single communication, can pose several challenges in processing. Here are a few of them:

- **Co-existence with monolingual data:** Co-existing code-mixed data with monolingual data in social media and other platforms can pose difficulties. The first step involves separating the code-mixed data from the monolingual data. It can be challenging, as it may require specialized tools and techniques to identify and separate the code-mixed instances.

- **Transliteration:** Transliteration can cause spelling differences in code-mixed data because it involves writing a word from one language using the script of another language. It can result in multiple spellings of the same word depending on the transliteration system or language script used.

  For example: The Telugu word *Vellava*, which transliterates to "Did you go?" in English can be written in Roman script as vellava, velava, vellavaa.

- **Grammatical Structures:** Code-mixing may result in sentences that do not follow the grammatical rules of either language, as different languages may have distinct grammatical rules. For example, a speaker may use a noun from one language and a verb from another, creating a sentence that is not grammatically correct in either language.

Apart from these challenges, there are other challenges like cultural misunderstandings and language ambiguity in understanding code-mixed texts.

1.3 Motivation

The motivation for this thesis comes from the observation that code-mixing is not only prevalent in informal communication on social media platforms but also in formal settings like political campaigning. This led to the realization that there is a need to understand code-mixing from a sociolinguistic perspective and develop techniques to analyze and understand code-mixed texts in political discourse across different communicative contexts.

During this research phase, the scarcity of quality code-mixed data, especially for Indian languages, gave rise to the idea of code-mixed text generation. On the other hand, developing rule-based systems for code-mixed text generation is a tedious and challenging task. Rule-based systems require a thorough understanding of the linguistic features of both languages and the rules governing their combination. This process is complicated because code-mixing is highly contextual and dependent on various social and linguistic factors. Therefore, neural machine translation models offer a more flexible and efficient solution for code-mixed text generation, which is the focus of this thesis.

Furthermore, developing neural machine translation systems has also brought about new challenges. Unlike rule-based systems, which require the manual development of linguistic rules and extensive domain knowledge, neural machine translation systems rely on large amounts of parallel data to learn the translation patterns between languages. This challenge motivated the development of architectures that can leverage the limited available data to generate high-quality code-mixed text with minimal human intervention. In this thesis, we aim to address this challenge by proposing novel techniques for generating code-mixed text in Indian languages using neural machine translation models.

1.4 Key contributions of Thesis

The thesis makes important contributions in the area of code-mixed text analysis and generation as follows:

- Analysis of code-mixed text in different political scenarios and determination of the factors that influence varying code-mixing depending on context.
- Creation of a dataset comprising speech-to-text translation of six Telugu speeches, capturing the dialectal differences in Telugu.
- Development of over 50 rules differentiating Standard and Telangana dialects of Telugu Language.
- Building a novel fine-tuned neural machine translation system which uses filtering of synthetic data for generating code-mixed text in Indian languages.
- Developing neural machine translation models for code-mixed text generation that outperform existing models when trained on synthetic data and tested on human-generated data.
- Building a robust code-mixed text generation system that is extensible and requires minimal human intervention to develop code-mixed text in low-resource languages.

1.5 Thesis Overview

The test of the thesis is divided into five chapters as follows:

Chapter 2 of the thesis provides an overview of the existing research work on code-mixing. It discusses the sociolinguistic perspective of code-mixing, various machine translation techniques, and different tasks related to code-mixing with a special focus on Indian languages. It also highlights the progress made in neural machine translation techniques and other related tasks in code-mixing, both in general and in the context of Indian languages. The chapter serves as a foundation for developing the proposed research work in subsequent chapters.

Chapter 3 discusses the work on analyzing code-mixing and code-switching in political speeches as a case study of political discourse analysis. It also illustrates the rules to differentiate between standard and Telangana dialects.

Chapter 4 describes a Hybrid methodology using rule-based and statistical approaches to convert a monolingual sentence to code-mixed text. The shortcomings of this approach are also discussed.

Chapter 5 presents our proposed method, a fine-tuned neural machine translation approach, for generating code-mixed text in Indian languages.

Chapter 6 presents the conclusions drawn from the thesis and outlines the scope for future work in code-mixing research.

Chapter 2

Related Work

In this chapter, we will comprehensively review the previous research conducted in the field of code-mixing, with an emphasis on Indian languages. We will examine the works done in the sociolinguistic perspective of code-mixing and explore various techniques for generating code-mixed texts. Additionally, we will discuss the machine translation techniques, tasks, and progress made in code-mixing, both in general and in Indian languages specifically. This chapter will provide the necessary background and context for the subsequent chapters, where we present our contributions to the field.

2.1 Code-mixing: Previous research works

The initial works of code-mixing were proposed on how languages interact to form code-mixed sentences. They included structural theories [40][37] linguistic constraints [57][7], grammatical constraints [67]. There have also been recent works using the X-bar theory [6].

Code-mixing is a phenomenon that occurs due to multilingualism, [4] examines the framework and discourse functions of code-mixing along with the factors which influence them. To enable further analysis and NLP tasks on code-mixing, [5] provides an annotation scheme for the pragmatic function on Hindi-English code-mixed tweets.

Language identification plays a crucial role in dealing with code-mixed data, as code-mixing involves combining elements from different languages. [17] utilized a dictionary-based approach for word-level language identification in English-Bengali and English-Hindi chat messages. Conditional Random Fields(CRF), Deep-learning and other advanced approaches are used for language identification [61] [35] [44] [25] [73]in Indian languages.

Continuing on this line of research, there have been several studies exploring code-mixing in various NLP tasks, such as POS-tagging [74] [3], named entity recognition [66] [9], sentiment analysis [56] [13] [26] [52], and Offensive language detection [61] [31]. These studies have used a variety of techniques, including neural networks, rule-based methods, and statistical models, to address the challenges posed by code-mixed data.

[30] proposed various code-mixed metrics to interpret the languages ratio and probability of switching between them and the time of switching in code-mixed text. The metrics are M-index, I-index, Language Entropy, Burstiness and Span entropy.

2.2 Sociolinguistic Perspective of Code-mixing

Understanding code-mixing from a sociolinguistic perspective is crucial, as switching languages in conversations is mainly influenced by the speakers' communicative goals and preferences.

According to research conducted by [2], code-mixing can be utilized as a means of conveying social meaning and indicating social relationships. This may include signifying group membership or demonstrating solidarity, among other functions.

[32] examines the social factors that drive code-switching in Malaysian-English bilingual speeches and concludes that the primary motivation for language switching among speakers is for expressing identity.

[75] study looked at how young professionals in Egypt use English and Arabic in online communication. The research shows that English is predominantly used in web usage and formal email correspondence, while a Romanized version of Egyptian Arabic is extensively used in informal email messages and online chats within this particular group.

[1] examines how code mixing and code switching have become more common in written Arabic commercials and concludes that this kind of writing is correlated with particular age groups, businesses, and socio-economic backgrounds of the targeted audience.

2.3 Code-mixed Text Generation

To build accurate downstream models for natural language processing tasks such as sentiment analysis, named entity recognition that process code-mixed data, it is necessary to have high-quality code-mixed text for training and evaluation.

Several workshops [16] [33] have been conducted to encourage research in code-mixed text generation. These workshops have played a significant role in advancing research on code-mixed text generation and have led to the developing of several novel approaches and techniques in this field.

2.3.1 Linguistic Theory Based Approaches

[55] proposed Equivalence Constraint Theory(ECT), which says that code-switching happens when there is a functional equivalence between the source language and the target language in terms of the meaning, pragmatics, or discourse function of a sentence. [58] proposed a computational technique for

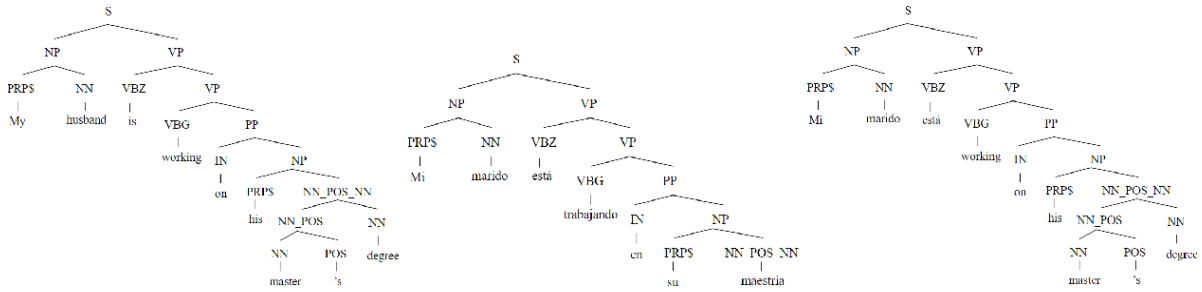

Figure 2.1: Parse-trees of (a) sentences [2E] and (b) [1S], and (c) of [2CM] according to the ECT

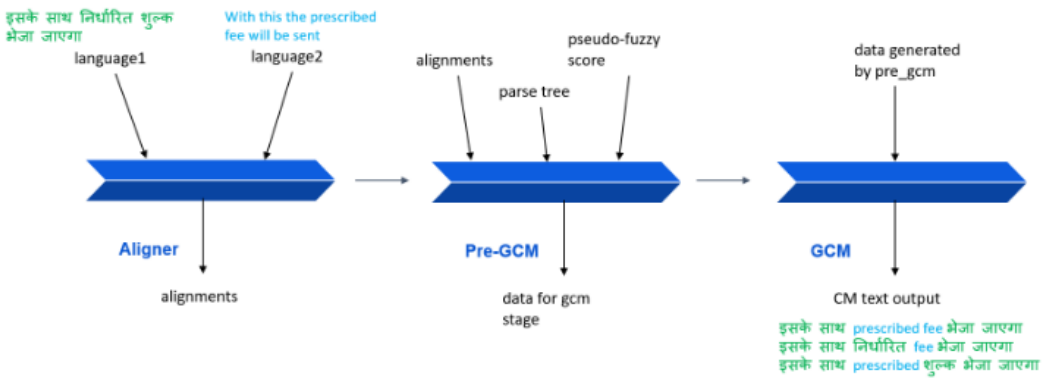

Figure 2.2: The CM Generation Process in GCM

creating grammatically valid artificial code-mixed (CM) data based on the ECT. The paper also depicts that the perplexity of the language model is not reduced when trained on random code-mixed sentences.

[71] proposed a Matrix Language Frame, which says that when bilinguals mix languages, they use a dominant language or ”matrix language” as a structure into which they integrate words or phrases from the non-dominant language. This framework has been refined in subsequent publications by Myers-Scotton and other linguists.

[60] proposed GCM, an open-source toolkit which generates multiple code-mixed sentences for a given pair of parallel monolingual sentences. The possible code-mixed sentences are generated using the Equivalence Constraint and Matrix Language theories.

2.3.2 Language Models based approaches

[20] [21] [45] used language models for generating code-mixed text. These works have utilized various techniques, such as combining two monolingual language models using a probabilistic method and pre-training the language model using synthetic text for generating code-mixed text.

2.3.3 Neural Machine Translation(NMT) approaches

Machine translation is the process of using algorithms to translate text or speech from one language to another automatically. The task of code-mixed text generation can be posed as a machine translation task, where a monolingual sentence is translated into a text containing one or more languages.

The paper by Chang et al.[15] proposed a generative adversarial network (GAN) for generating code-switched sentences from monolingual sentences. The paper by Gupta et al. [28] proposed a semi-supervised approach using pre-trained encoders for generating code-mixed text.

[36] uses curriculum training to generate code-mixed Hindi-English data. In the curriculum training, the pre-models are fine-tuned by training them on synthetic data and then on gold code-mixed data. This architecture has achieved a BLEU score of 12.67 and was placed first in the overall ranking of CALCS[16] shared task.

[22] has explored mBART pre-trained multilingual sequence-to-sequence model to generate Hindi-English text. This method illustrates the improvement in performance by converting the Hindi roman script to Devanagari script and concatenating Hindi and English sentences for training.

Chapter 3

Political Discourse Analysis: A Case Study of Code Mixing and Code Switching in Political Speeches

.

3.1 Introduction

In a linguistically diverse country like India, political leaders utilize various methods to establish relationships with people from different language backgrounds. Over time, various communication modes have emerged to enable political leaders to connect with people. By speaking to people in their native language or language variety, politicians can create a sense of approachability and relatability, leading to stronger connections with the public. Code-mixing and code-switching can also be utilized strategically to connect with specific communities and build political support.

[24] researched specific speech events to examine the relationship between speakers' linguistic choices. [23] found that local dialect carried great prestige. A person's native speech is regarded as an integral part of his family background, a sign of his local identity. However, when interacting with members of other communities and with tourists, the residents would use the standard dialect.

[38] conducted a study about various interpersonal speech choices in election campaign speeches, including the usage patterns of nouns, pronouns, kinship terms, rhetorical questions, etc. There are a few more studies ([53], [34], [39]) which analyze the deeper intention behind the choice of words and phrases using the famous Speech Act theory by [63] and the Sociocognitive model by [72].

However, to our knowledge, this work is the first of its kind, which analyses code-mixing and code-switching together in political discourse along with dialectal level code-mixing analysis. We aim to understand these phenomena as a speech choice and its effect on the audience in politics.

The matrix language we have chosen to study these phenomena is Telugu, a South-Central Dravidian language predominantly spoken in India's Southern parts, especially in Andhra Pradesh and Telangana. There are many regional dialects and sub-dialects in Telugu, but the three major dialects are the Coastal Andhra dialect, the Telangana dialect which has a significant influence of Urdu/Dakhni, and the Ray-

alaseema dialect. The variety spoken by the educated class from the interior districts of Andhra area was modernised and elevated to 'Standard Telugu' status in 1969 and since then has been widely used in textbooks, newspapers and other formal communication. It is also referred to as Modern Standard Telugu (MST) in [41].

Even though Telugu is one of India's largest spoken languages, with more than 80 million speakers, there is a severe dearth of resources in Telugu, which makes it hard for NLP research. For our purpose, we did not find any corpus for Code Mixing and Switching in Telugu. Hence, we created a corpus of political speeches in Telugu consisting of 1134 sentences and about 10000 words. Since our work is closely associated with a set of rules and statistical observations corresponding to those rules, the corpus size was sufficient to give us good results. We will further examine the factors responsible for varying levels of code-mixing and code-switching in these speeches.

3.2 Dataset and Annotation

3.2.1 Dataset collection

Even with the advent of social and print media, in-person modes of communication such as campaigns and political speeches remain the most preferred ways of communicating with the general public for politicians. They try to ensure that the audience feels connected to them, thereby increasing their potential votes. This is done strategically and persuasively.

We chose our speakers as Mr K Chandrasekhar Rao (KCR), the Chief Minister of Telangana and Mr Chandra Babu Naidu(CBN), former Chief Minister of Andhra Pradesh. KCR is the founder of the Telangana Rashtra Samiti (TRS) party and is widely regarded as the face of the Telangana movement for a separate state in 2014. CBN is the leader of the Telugu Desam Party. They use a variety of dialects and languages such as Telangana Telugu, Modern Standard Telugu, Urdu, English and Hindi in their speeches.

We chose a total of 6 speeches of both the speakers in three different social settings and communicative contexts to analyse the levels of code-mixing and code-switching as follows:

1. **Public Meetings in Telangana**: KCR's speech was during the Telangana movement, meant for the creation of a new state. He addressed the pathetic situation of Telangana residents and also discussed the plan and policies for the new state.CBN's speech is during the Telangana elections in 2018. The audience were residents of Telangana. We will refer to this as communicative event 1.

2. **Felicitating Dr. Venkaiah Naidu when honoured as Vice President**: KCR and CBN's speeches were with MLAs and other parliament members of their respective states,*viz*. They spoke about Dr Venkaiah Naidu's great qualities and praised him for his service to the nation and attaining one of its highest positions. We will refer to this as communicative event 2.

3. **Capital Development**: In these speeches, both the speakers were talking about developments of capitals. In the KCR speech, the audience were Government officials and local politicians of Telangana. CBN addressed the collectors of Andhra Pradesh. We will refer to this one as communicative event 3.

Though the speeches were available on YouTube, none of the existing off-the-shelf speech to text systems could serve to capture the speech effectively along with the dialectal variations in the language. Therefore, we manually transcribed the speeches in the WX format [19] and verified the transcription with the help of native speakers. The duration of the speeches is 100 minutes for each speaker, and after transcription, it consists of 1134 sentences. The total word count is around 10000.

3.2.2 Annotation

All speeches are annotated for the usage of code-mixing and code-switching at the word level. Each speech is annotated for Dialectal level code-mixing(DCM), Language level code-mixing (LCM) and code-Switching(CS). We will further examine how code-mixing and code-switching will vary in different social settings and communicative contexts for pragmatic reasons.

3.2.2.1 Guidelines to handle dialectal level code-mixing

The subjects of our study use Telangana dialect and MST more often compared.

To our knowledge, there has been no exhaustive set of observations differentiating these two varieties Telangana from MST. We took some observations from the book by [10] and [62]. Few more observations are drawn from texts of [14]. Also, we compiled a few more observations from a TV news program named *Teenmar news* which uses Telangana dialect. After removing duplicates, we categorised the observations and segregated them into three categories: Vowel rule (V), Consonant rule (C) and the other rules which apply to syllables (S). The rules in each of these three categories are further classified as *Addition, Deletion or Replacement*, based on the kind of operation performed.

We came up with over 50 tags for these observations capturing the pattern differences between the Standard and Telangana dialects. All the observations are listed in appendix(A). If a word follows any of these observations, then it is marked as 1 under the category DCM. Else, it is marked as 0. In this chapter, we present a few observations which are prominent in our data. The writing convention followed is:

[Standard dialect word] - [Telangana dialect word]:

1. **Vowel rules**

    - **Deletion**: In Telangana dialect, vowels are dropped at the end of some words. For example:

        nenu - nen

- **Replacement**:
  - Long vowels are replaced with short vowels.

    vaswAru - vaswaru

    ceswAru - ceswaru
  - In Telangana, verbs ending with *i* are replaces with *e*

    ceyAli - ceyAle

2. **Consonant rules**

- **Addition**:
  - In Telangana dialect g is added at the start for few words. This phenomenon is more prevalent deixial terms. Some words or phrases in a language need additional information to be fully understood. For instance, English pronouns are examples of such words since their meaning relies on the situation or context they are used in.

    ippuDu - gippuDu

    Ayana - gAyana

    Anthe - ganthe
  - For certain verbs, n is added either at the start of in between the syllable

    ceVppAlA ? - ceVppAlna ?

    padukune - pandukune
- **Deletion**: In some words *v* is dropped at the beginning of the word. This occurs in nouns, pronouns and verbs

  vAna - Ana

  vAllu - Allu
- **Replacement**: Voiced consonants are replaced with voiceless consonants in some words.

  pedda + kAleV- peddagAleV -

  cAlu - jAlu

  peTTAru - beTTAru

3. **Syllable rules**

- **Deletion**: Dropping of the syllable which precedes the /d/ sound. In some cases, after the dropping, the preceding vowel is lengthened. This is mostly observed in terms associated with spatial deixis.

  ikkaDa - IDa
- **Replacement**: For the verbs in past tense, The second last syllable's long vowel gets replaced with *in/i/shortening of vowel/ina*. These are further sub-categorised based on gender, number and person.

cesAru - jeSinru
cesAvA - jeSinavA
cesAru - jeSiru

3.2.2.2 Guidelines to handle language level code-mixing

In this chapter, language level code-mixing is said to occur when two or more languages or language varieties are used at a morphological level. To be more precise, it occurred when English root words were suffixed with Telugu plural markers, and morphological suffixes in one word or English/Hindi words are used.

pArtIlu - party + lu
kAlejIlo - College + lo
rejiyanga - region + ga

If a word follows these observations, then it is marked as 1. Else, it is marked as 0 for language level code-mixing.

3.2.2.3 Guidelines to handle code-switching

All the language variations at the sentence level, i.e. if the sentence or phrase with more than one word is in a different language, then it is considered under code-switching. Here as our speeches are in Telugu, sentences or phrases in languages other than Telugu come under this category. All the words in these sentences/phrases are marked as 1.

mIru ganaka commitement won
tIskunte, Yes sir come on let us move annAru

In the above sentence, all the words in the phrase *Yes sir come on let us move* are marked as 1 under the category code-switching.

3.3 Observations and Results

After annotating based on these guidelines, the results are tabulated as follows.

In communicative event 1, as they were addressing Telangana residents, relatively higher levels of Telangana dialect are observed in speeches by both the speakers to get more *connection with the audience*. However, KCR has used more Telangana dialect in his speech than CBN. KCR was fighting for a separate Telangana state. CBN speech was during the Telangana elections in 2018. His ideology

Speech	No.of Words	Dialectal-level code-mixing	Language-level code-mixing	Code-Switching
1	2153	19.9%	E- 5.4% H- 0.8%	E- 0.1% H- 17.4%
2	1137	12%	E - 3.1% H - 0%	E - 2.1% H - 0%
2	2654	15%	E - 7.01% H - 0%	E - 39.93 H - 0%

Table 3.1: KCR Speech Statistics (E-English, H-Hindi/Urdu)

Speech	No.of Words	Dialectal Level Code-mixing	Languages-level code-mixing	Code-switching
1	1357	8.91%	E - 4.64% H - 0%	E - 1.76% H - 0%
2	1960	3.82%	E - 4.7% H - 0%	E - 4.33% H - 0%
3	984	2.7%	E - 7.01% H - 0 %	E - 39.63% H - 0%

Table 3.2: CBN Speech Statistics

doesn't align with KCR. In addition to connection with the audience, *ideologies of the speaker* also impact the levels of code-mixing and code-switching. KCR also uses high levels of code-switching in Hindi for establishing a better connection with the audience as the Telangana dialect is influenced by Hindi/Urdu.

In communicative event 2, KCN and CBN addressed MLAs and other parliament members of Telangana and Andhra Pradesh. In CBN's speech, the usage of MST can be due to the absence of Telangana residents. However, in KCR speech, most of them are Telangana residents, yet lesser levels of Telangana dialect are observed. So, *context of the speech* also determines the levels of code-mixing and code-switching. In this communicative event, as they were addressing a national topic, MST, lesser language level code-mixing and lower code-switching levels are observed.

In communicative event 3, English usage is high in both speeches than other speeches as the meeting is about capitals and all government officials may not be aware of the local language. In KCR speech, local politicians are also part of the meeting, so Telangana dialect usage is prominent. Whereas in CBN speech, very high levels of English is used as the meeting is only with collectors.

3.4 Conclusions

In this chapter, we looked at the phenomenon of code-mixing/code-switching between dialects of Telugu, MST and languages like English and Hindi/Urdu for different communicative contexts. The audience, ideologies of the speaker and context of the speech impacted the speakers linguistic choices.

Chapter 4

Integrating Linguistic Rules and Statistical Methods for Code-Mixed Text Generation

4.1 Introduction

Machine Translation is the process of automatically converting text or speech from one language to another language. The three primary categories of machine translation are rule-based, statistical, and neural. Rule-based systems rely on a set of linguistic rules to produce translations, while statistical systems utilize large amounts of bilingual text to identify patterns and probabilities. Neural machine translation systems employ deep learning to generate translations.

[8] proposed a Rule-based Machine Translation System (RBMT) for translating Hindi to Sanskrit using the prominent linguistic characteristics of both languages. [42] have built Statistical Machine Translation (SMT) systems for languages belonging to Indo-Aryan and Dravidian families.

[47] proposed a method for translating from English to Telugu using a rule-based approach that maps English prepositions to Telugu postpositions. There have several other dictionary-based hybrid approaches also [49] [48] [59]. [65] [70] conducted a linguistic study to carefully select the appropriate translation rules to improve translating accuracy of the Tunisian Dialect to Modern Standard Arabic.

Code-mixing is typically characterized by using multiple languages or language codes within a conversation. However, it is worth noting that code-mixing can also involve using different dialects or variations of the same language in addition to different languages. Although many studies have focused on code-mixing involving different languages or language codes, relatively less research has been done on code-mixing between different dialects of the same language. Only a few studies have explored this type of code-mixing [46].

It is the first attempt to convert a monolingual Telugu sentence into a code-mixed Telugu-English sentence with dialectal variations. In order to overcome the limitations posed by the lack of comprehensive reference databases for dialect-level transfer, a hybrid approach that combines rule-based and statistical methods is proposed as a viable solution.

4.2 Methodology

This section will cover a detailed explanation of converting a Telugu sentence into a code-mixed Telugu-English text and subsequently incorporate dialectal-level code-mixing into the sentence. A pipeline is followed to reach the final translation, which will be demonstrated with an example sentence.

Telugu text: *repu nenu kalASAlakivelli akkaDa cAlA yerpAtulu ceyAli. anxuke, ippuDu wonxaragA padukuntunAnu. sare Exe mari, SuBarawri.*

English Translation: Tomorrow I have to go to the college and make a lot of arrangements. So, I am sleeping early. Okay then, Good night.

4.2.1 Sentence level code-mixing

As mentioned in chapter 1, [24] refers code-switching (CS) as switching of languages or codes between sentences or phrases, while code-mixing involves the mixing of languages or codes within a sentence or phrase. Code-mixing is used as a general term that encompasses both phenomena. Each sentence in the sample can be translated into English. However, to make it a code-mixed sentence, a statistical-based approach can be used to select the most appropriate sentence to be translated into English.

The most common way for a Telugu speaker would be to use English in the last sentence. The resulting code-mixed sentence would appear as follows:

Code-mixed text: *repu nenu kalASAlaki velli akkaDa cAlA yerpAtulu ceyAli. anxuke, ippuDu wonxaragA padukuntunAnu.* ***Okay then, Good night****.*

4.2.2 Word level code-mixing

4.2.2.1 Language level:

In a code-mixed sentence, there are two or more languages. When people mix languages, they usually have a primary language they use more often, called the matrix language, and another language they mix in, called the embedded language. Over time, these words can become a regular part of the language they are borrowed into.

To achieve this computationally, we can create a dictionary that contains Telugu words and their corresponding English equivalents. We can also create a separate list of commonly used English words to replace Telugu words and replace them accordingly within the text to incorporate language-level code-mixing in the text.

In the given sentence, the speaker tends to replace certain words with their more commonly used English counterparts, such as:

- anxuke - so

- kalASAla - college. As vibhakti is attached to the word, when this Telugu word is replaced by English, it takes a new form. *kAlejIki* - Collage + ki

The resulting sentence with language-level code-mixing:

Code-mixed text: *repu nenu **kAlejIki** velli akkaDa cAlA yerpAtulu ceyAli. **So**, ippuDu wonxaragA padukuntunAnu. Okay then, Good night.*

4.2.2.2 Dialectal level code-mixing

Code-mixing can also occur at the dialect level, as demonstrated in the preceding chapter, where guidelines were given for transforming MST of Telugu into Telangana dialect. These rules can be translated computationally to apply them to words. Some rules are relevant only to deictic words, while others apply to verbs. Therefore, the algorithm for implementing code-mixing at the dialect level includes the following steps:

1. POS-Tagging
2. Diexis Identification
3. Application of rules to each word

For the given sentence, the following rules are applicable:

- On applying the vowel deletion rule:

  nenu - nen

- On Applying the vowel replacement rule:

  ceyAli - ceyAle

- On Applying the consonant addition rules of adding n and adding g:

  padukuntunAnu - pandukuntunAnu
  ippuDu - gippuDu

- On Applying the syllable deletion rule:

  akkaDa - aDa

The code-mixed sentence resulting from applying these changes is as follows, and the resulting dialectal-level code-mixed sentence is:

Code-mixed text: *repu **nen** kAlejIki velli **aDa** cAlA yerpAtulu **ceyAle**. So, **gippuDu** wonxaragA padukuntunAnu. Okay then, Good night.*

4.3 Limitations of the approach

Although this method can generate accurate code-mixed sentences in Telugu-English, it has some limitations that need to be addressed.

- The process of language-level code-mixing, which involves creating dictionaries for word replacements, requires a significant amount of linguistic expertise and manual effort.
- One limitation of this approach is that it only considers a limited set of rules for dialect-level code-mixing. Many other rules may need to be accounted for, which can limit the accuracy of the resulting code-mixed sentences. It is important to explore other approaches and gather more comprehensive sets of rules for more accurate code-mixing.
- This approach may not be applicable to other languages as the rules for dialectal-level code-mixing are specific to each language.
- For dialectal-level code-mixing, a word can undergo changes following multiple rules. A hierarchy should be proposed to determine the order in which applicable rules should be applied to words.

4.4 Conclusion

In conclusion, this chapter proposes a pipeline for code-mixed text generation in Telugu and underscores the importance of data in creating rules, which should be based on a comprehensive dataset. However, only covers some possible interpretations of Telugu code-mixing at the dialect level This chapter is a starting point for understanding the challenges and potential solutions for generating code-mixed text from a machine translation perspective. On the other hand, the GCM toolkit generates possible code-mixed sentences using linguistic theories. In the next chapter, we will explore how we can use this toolkit to build better systems for code-mixed text generation.

Chapter 5

Code-mixed Text generation using Filtering of Synthetic Data in Neural Machine Translation

5.1 Introduction

Developing Natural Language Processing (NLP) systems for code-mixed text or speech has become increasingly challenging due to its informal nature. This is amplified by the vast amount of data available on various platforms, which often coexists with monolingual data. Non-standard spelling and lexical and grammatical structures further add to the complexity of processing such text. These challenges result in a scarcity of code-mixed data for training models, underscoring the importance of code-mixed text generation.

In the previous chapter, we discussed the difficulties of generating code-mixed text using rule-based and statistical methods. This chapter will explore the neural machine translation approach for code-mixed text generation.

In recent times, the architectures based on pre-trained models [50] [18] have become the state-of-art models for monolingual language understanding and generative models. The underlying advantage of these models comes from training these models with huge monolingual data from Wikipedia, books etc.

Equivalence Constraint Theory(ECT)[55] and Matrix Language Frame[71] were proposed to understand the code-mixing in languages. GCM, an open-source toolkit proposed by [60], generates multiple code-mixed sentences for a given pair of parallel monolingual sentences using Equivalence Constraint theory and the Matrix Language theory.

[27] proposed an Unsupervised Self-training approach for sentiment analysis of code-mixed data. To tackle the problem of scarce annotated code-mixed data, this approach used minimal data to start fine-tuning mBART and then used pseudo labels obtained by zero-shot transfer for further training.

[36] [22] various techniques such as curriculum training and pre-trained mBART for multilingual sequence-to-sequence have been explored for code-mixed text generation.

To our knowledge, this is the first work which uses a Filtering mechanism with Neural Machine Translation for code-mixed text generation.

This chapter proposes a fine-tuned Neural Machine Translation system that generates high-quality human-interpretable code-mixed text using minimal gold data. Specifically, we focus on Hindi-English and Telugu-English code-mixed text generation. The code-mixed data created by this system can be further used for downstream tasks such as sentiment analysis, building conversation systems, Hate-speech detection etc.

The main contributions of this chapter are summarized below:

1. Our approach introduces two mechanisms for **Quantitatively filtering** synthetically generated code-mixed texts, leveraging human knowledge.

2. We demonstrate the robustness of our system by applying the generation mechanism to a low-resource language such as Telugu, making it the first Neural Machine Translation model for Telugu-English code-mixed generation. Our extension is possible since we train the models with silver data generated from filters that utilize small amounts of the gold corpus.

3. We created a dataset of 3500 manually annotated Telugu-English code-mixed sentences, where two annotators rated each sentence to ensure the consistency and accuracy of the annotations.

4. Our best model for Hindi-English code-mixed text generation outperforms the [69] architecture when trained on synthetic data and tested on ALL-CS dataset [1].

5.2 Architecture

In this chapter, we aim to build a Hindi-English and Telugu-English code-mixed text generation system using Seq2Seq models. We use a filtering mechanism to train the model with high-quality code-mixed sentences. A huge corpus is filtered through Regression and Probabilistic methods, and the resultant data obtained after filtering is used for training Seq2Seq models.

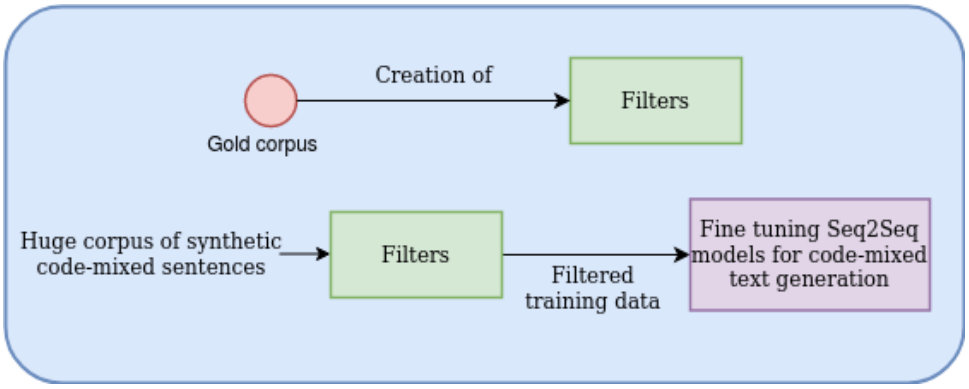

Figure 5.1: Methodology for Code-mixed Text Generation

[1] https://github.com/ishan00/translation-for-code-switching-acl

English	Hindi	Human-generated Hinglish	WAC	WAC rating1	WAC rating2	PAC	PAC rating1	PAC rating2
He propagated the principles of Hinduism.	उसने हिन्दुत्व के नियमों का परचार किया।	['Usne principles of hinduism ka prachaar kiya.', 'Usne principles of hinduism propagate kiye.', 'Usne hinduism ke niyamon ko propagate kiya.', 'He propagated hindutva ke principles.']	usne hinduism ke principles ka prchar kiya.	10	8	usne principles of hinduism ka propagated kiya.	8	4

Table 5.1: A sample from HINGE Dataset

5.2.1 Filtering Mechanism

In this step, we create filters to generate high-quality training data for Seq2Seq models. GCM toolkit generates all the possible code-mixed sentences for parallel monolingual sentences. However, all the code-mixed sentences are not commonly used by humans. These filters leverage human knowledge and select the best samples from synthetically generated code-mixed sentences using the following approaches.

5.2.1.1 Regression

In this method, we train regression models to predict the ratings of code-mixed sentences. Human annotators rate the code-mixed sentences based on their readability and grammatical correctness. The training data preparation is as follows:

Hindi-English: We train Hindi-English regression models using [68] HINGE dataset comprising 4000 Hinglish code mixed sentences. These code-mixed sentences are generated using two rule-based linguistic theories: Word-aligned code-mixing (WAC) and Phrase-aligned code-mixing (PAC) corresponding to the parallel monolingual Hindi and English sentences. Two different annotators rated these code-mixed sentences on a scale of 10.

Telugu-English: Due to the lack of Telugu-English code-mixed datasets evaluated by humans, we created a dataset. The purpose of this dataset is to aid the regression model in filtering out high-quality samples by being aware of all possible code-mixed sentences. To serve this purpose, we randomly pick one code-mixed sentence from a set of code-mixed sentences generated by GCM using a pair of Telugu and English sentences. We have picked 5000 such sentences. An annotator then rated each sentence on a scale of 5 based on readability, grammatical correctness, and semantic correctness. A rating of 5 is given to a sentence if it is the most commonly used and interpretable by humans. Two annotators rated each sentence to ensure the validity and reliability of the dataset. It is to be noted that the set of code-mixed sentences generated by GCM also included monolingual sentences as output. After removing these monolingual sentences, the resulting dataset comprised 3500 rated code-mixed sentences.

For both datasets mentioned above, two annotators annotated each code-mixed sentence. The ratings provided by the annotators were then averaged to obtain the overall rating for each sentence. The selected features for training are as follows:

English	Telugu	Machine generated Telugu-English	Rating 1	Rating 2
Take advantage of this opportunity	ఆంది వచ్చిన అవకాశాన్ని సద్వినియోగం చేసుకోవాలి	this opportunity సద్వినియోగం చేసుకోవాలి	5	4

Table 5.2: A sample from Telugu-English Dataset

- BLEURT scores: BLEURT [64] score, which is a reference-based text generation metric, aids us in capturing the semantic meaning of the code-mixed texts. BLEURT scores are calculated between monolingual and re-generated monolingual sentences from code-mixed texts using Google Translate.
- Code-mixed(CM) metrics: Code-mixed metrics capture the linguistic phenomenon and complexity of the code-mixed sentences. These include CMI, M-index, I-index, Burstiness and Language Entropy. Code-mixed metrics need language tags for their calculation. Language tags are assigned using [11] for Hindi-English, and script-based identification is used for Telugu-English.

Using these input features that capture the semantic and linguistic aspects of code-mixed language, we train regression models to predict the rating of each code-mixed sentence.

- **Linear Regression:** Linear regression is a statistical approach for modelling the relationship between the dependent variable and one or more independent variables. It finds the best line that minimizes the difference between predicted and actual values.

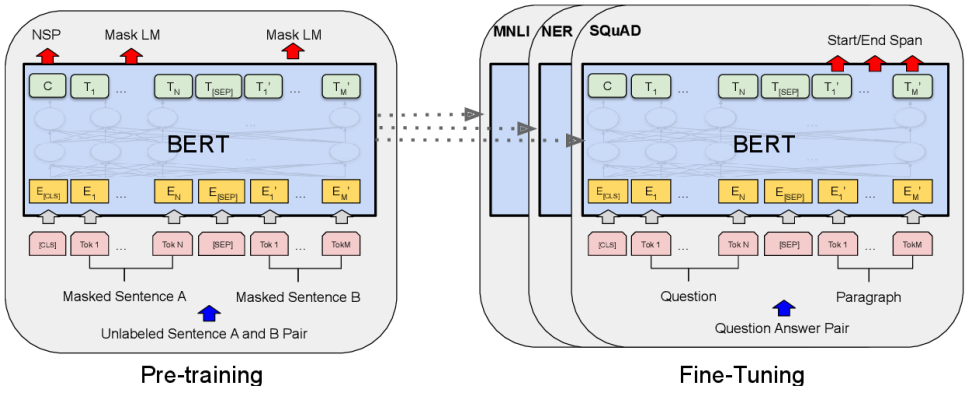

Figure 5.2: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning.

- **Polynomial Regression:** Polynomial regression models the relationship between dependent and independent variables as an nth-degree polynomial. In this regression model, we use 2nd-degree polynomial.
- **BERT Regression:** BERT (Bidirectional Encoder Representations from Transformers), a pre-trained deep learning model, is used as a regression model by adding an output layer. In addition

to the input features, the code-mixed sentence is also passed as input to this regression model. Figure 5.2 illustrates the pre-training and fine-tuning architectures of BERT.

The performance of each regression model is evaluated using various metrics such as the mean absolute error, the root mean squared error and the coefficient of determination(R2 score). The evaluation results for the Hindi-English and Telugu-English code-mixed datasets are presented in Table 5.3 and Table 5.4, respectively. We use the best regression models from both languages as a filter for selecting high-quality code-mixed sentences.

Regression	**MSE**	**RMSE**	**MAE**	**R2_Score**
Linear	2.145	1.464	1.186	0.100
Polynomial(degree-2)	2.141	1.463	1.186	0.101
BERT	2.074	1.440	1.158	0.130

Table 5.3: Evaluation of Regression Models for Hindi-English

Regression	**MSE**	**RMSE**	**MAE**	**R2_Score**
Linear	1.308	1.143	0.947	0.274
Polynomial(degree-2)	1.303	1.141	0.943	0.271
BERT	1.107	1.052	0.826	0.383

Table 5.4: Evaluation of Regression Models for Telugu-English

5.2.1.2 Probabilistic methods

The regression filter relied on the ratings assigned to synthetically generated code-mixed sentences. However, the HINGE dataset also includes human-generated sentences in addition to the synthetically generated ones; the former lack ratings, rendering them unsuitable for regression analysis.

So, the proposed method involves creating a filter that can utilize the information contained within human-generated code-mixed sentences to identify samples that closely resemble them. This is achieved by scoring the samples based on the probabilistic distribution of features observed in human-generated code-mixed sentences.

The same set of features used to train the regression models are utilized for calculating the score in the Probabilistic method.

The score of a code-mixed sentence is calculated as the sum of probabilities of its feature values occurring in the human-generated sentences. The formula used for calculating is as follows:

$$score(CM) = \sum_{f=1}^{n} Prob(f(Value)) \tag{5.1}$$

where: $Prob(f(Value))$ = Probability of a feature value

For instance, if a sentence has a CMI of 50, we calculate the probability of code-mixed sentences with a CMI index of 50 being present in our corpus of human-generated code-mixed sentences.

The probability of a feature value is calculated using the Kernel Density Estimation of the feature. In statistics, **Kernel density estimation (KDE)** is the application of kernel smoothing for probability density estimation. It is a non-parametric method to estimate the probability density function of a random variable based on kernels as weights.

Given a Kernel Density Estimation curve for a feature, the probability for an interval of values can only be obtained. We estimate the probability for a particular value by calculating the probability for the range of values (featureValue-0.01, featueValue+0.01).

As we are utilizing code-mixed content that humans have created, we have opted to utilize the same dataset for filtering in both Hindi-English and Telugu-English.

5.2.2 Data preparation for Seq2Seq Models:

The data for training Seq2Seq models is created using the above filters and synthetically generated code-mixed generated texts.

We use the GCM toolkit to generate a huge corpus of synthetically generated code-mixed sentences. The number of code-mixed sentences varies from each pair of parallel monolingual sentences to the other.

From 72,490 Hindi-English parallel sentences GCM toolkit generated 20,00,000 Hindi-English code-mixed sentences. This is named as GCM-HiEn corpus.

We passed 73,298 Telugu-English parallel sentences to generate 23,37,000 Telugu-English code-mixed sentences. This is named as GCM-TeEn corpus.

The code-mixed sentences for training Seq2Seq models using the above corpora are generated as follows:

1. **Random Sampler:** 40,000 sentences are randomly selected from each GCM-HiEn corpus and GCM-HiEn corpus.

2. **Polynomial Filter:** GCM-HiEn corpus and GCM-HiEn corpus are passed through their respective polynomial regression models, and the highest-rated 40,000 sentences are selected from each corpus.

3. **BERT Filter:** GCM-HiEn corpus and GCM-HiEn corpus are passed through their respective BERT regression models, and the highest-rated 40,000 sentences are selected from each corpus.

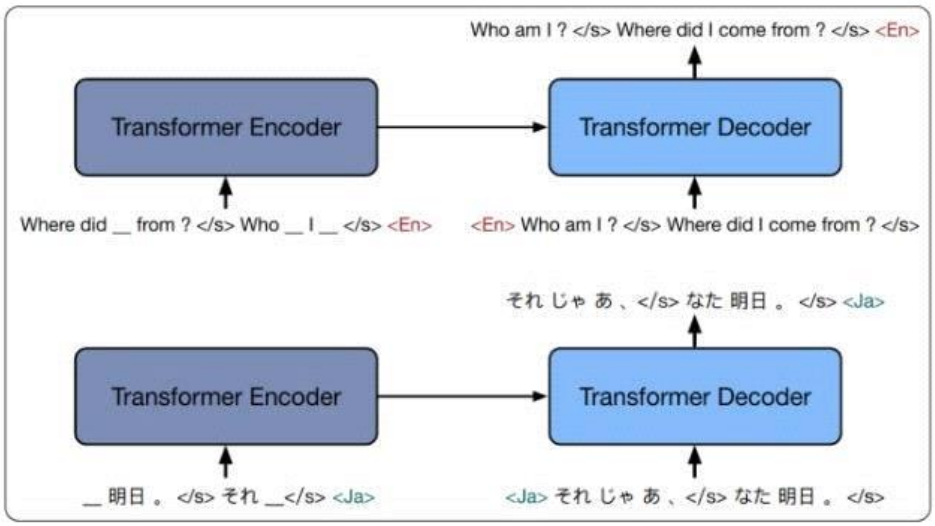

Figure 5.3: Framework for Multilingual Denoising Pre-training(mBART)

4. **Probabilistic Filter:** Scores are calculated for all the code-mixed sentences in GCM-HiEn and GCM-HiEn corpus. 40,000 code-mixed sentences having the highest scores are selected from both corpora.

5.2.3 Training Seq2Seq Models

The filtered data from the above filtering processes are passed through the following Seq2Seq models to generate Hindi-English and Telugu-English code-mixed sentences.

1. **mT5:** mT5 is a multilingual variant of "Text-to-Text Transfer Transformer" (T5), which is pre-trained on new Common Crawl-based dataset comprising 101 languages. This model is specifically designed for multilingual language processing tasks, including machine translation. The capability of this model with multiple languages and the ability to generate text output from text input makes it suitable for generating code-mixed text.

2. **mBART:**[51] proposed mBART, a Seq2Seq denoising auto-encoder pre-trained on monolingual corpora in many languages using the BART architecture. It comprises a shared encoder and language-specific decoders allowing it to transfer knowledge between languages preserving language-specific features. It has achieved state-of-art performance on many cross-lingual tasks, including machine translation. The figure 5.3 shows the architecture diagram for mBART.

5.3 Experiments and results

In this section, we conduct experiments to examine the effectiveness of each filter and its contribution towards generating high-quality code-mixed sentences. The experimental setup is described in detail, followed by a comprehensive analysis of the results obtained.

5.3.1 Experimental Setup

In our experimental setup, we performed fine-tuning of pre-trained language models, namely mT5 and mBART, for code-mixed text generation in Hindi-English and Telugu-English.

The input to these models consists of the concatenation of two corresponding monolingual sentences, and the output is a code-mixed sentence. For each language pair, we fine-tuned each model on four different training datasets created using raw, polynomial, BERT, and probabilistic filters, respectively.

We fine-tuned mT5 and mBART for Hindi-English and Telugu-English code-mixed text generation using appropriate hyperparameters. For mT5, we trained with a batch size of 64 and a learning rate of 2e-3, while for mBART, we used a batch size of 32 and a learning rate of 3e-6. We used the default AdaW optimizer for training both models and selected the hyperparameters to minimize the validation dataset loss.

For Hindi, we used two different datasets for testing: the ALL-CS dataset and the CALCS-2021 [16] shared task validation dataset. The ALL-CS test dataset contains code-mixed sentences and their corresponding Hindi translations, while the English translations for this dataset were generated using Google Translate. On the other hand, the CALCS dataset contains code-mixed sentences which include Hindi words written in Devanagari script.

For Telugu, we used the Sentiment dataset proposed by [43], which contains code-mixed sentences collected from Twitter. We selected 500 code-mixed sentences from this dataset. The monolingual sentences in the dataset were generated manually.

We evaluate the performance of our models using standard metrics such as BLEU scores and ROUGE-L scores and report the results in Table 5.5 and Table 6 5.6 for Hindi-English and Telugu-English, respectively.

5.3.2 Results

The BLEU and ROUGE-L scores demonstrate how **filtering plays a significant role in the model's performance**. All testing datasets are human-generated code-mixed sentences, highlighting how the filters aid in generating code-mixed sentences that closely resemble human-generated ones.

Our models achieved good results on various code-mixed datasets, despite the differences in sampling and characteristics between the training and testing sets. It suggests that our models are robust and can be applied to a wide range of datasets with varying characteristics. The ability to **generalize well across different datasets** highlights the effectiveness of our models.

	mBART				mT5			
	CALCS		ALLCS		CALCS		ALLCS	
	BLEU	**ROUGE-L**	**BLEU**	**ROUGE-L**	**BLEU**	**ROUGE-L**	**BLEU**	**ROUGE-L**
Random Sampler	1.89	18.02	3.14	10.62	2.91	18.40	6.75	9.94
Polynomial Filter	2.84	24.30	7.13	23.5	4.25	21.49	15.04	12.72
BERT Filter	4.9223	**32.46**	13.99	**33.02**	5.41	22.44	15.30	17.67
Probabilistic Filter	4.8422	28.82	17.43	28.80	**6.52**	24.84	**30.02**	20.67

Table 5.5: Performance of Hindi-English code-mixed generation models

	mBART		mT5	
	SentiDataset		SentiDataset	
	BLEU	**ROUGE-L**	**BLEU**	**ROUGE-L**
Random Sampler	4.56	18.54	7.86	20.52
Polynomial Filter	11.46	34.23	12.54	38.44
BERT Filter	10.04	47.30	14.05	39.56
Probabilistic Filter	**12.42**	**49.15**	**21.96**	**53.56**

Table 5.6: Performance of Telugu-English code-mixed generation models

The mBART model with BERT filtering achieved the highest ROUGE scores, while the mT5 model with probabilistic filtering achieved the highest BLEU scores for both the Hindi CALCS validation dataset and the ALL-CS test dataset.

The **Probabilistic filter** uses the feature distribution of human-generated code-mixed sentences to improve the model's performance. This filter was initially created using the Hindi-English dataset but was also applied to the Telugu-English dataset, demonstrating its **language-independent nature**. The good results obtained from Telugu-English code-mixed test generation highlight the power and effectiveness of this filtering mechanism.

In a similar experiment, where the system was trained on synthetic data and tested on the ALL-CS test dataset, the system achieved a BLEU score of 17.73. However, our mT5 model trained with data after applying probabilistic filtering outperformed it, **achieving a much higher score of 30.02**. This significant improvement highlights the importance of using a probabilistic filtering method for code-mixed language translation. It allows for better modelling of the underlying language patterns and improves the system's overall performance.

5.4 Conclusion

In conclusion, our filtering-based neural machine translation approach for code-mixed sentence generation shows promising results on various datasets, including Hindi-English and Telugu-English. The fine-tuning of pre-trained models such as mT5 and mBART has enabled us to generate high-quality code-mixed sentences with minimal gold-standard corpus.

The probabilistic filter is effective and language-independent, as demonstrated by its successful application to the Telugu-English dataset. It can easily be extended to other languages with minimal human effort, unlike other mechanisms.

Chapter 6

Conclusion and Future Work

6.1 Summary

The main focus of this thesis is twofold: firstly, to analyze code-mixed text from a sociolinguistic perspective, and secondly, to develop a method for generating code-mixed text for Indian languages. Analyzing code-mixed text from a sociolinguistic perspective can offer valuable insights into language choice and identity. The goal of developing a method for generating code-mixed text for Indian languages is to enable the creation of high-quality data for developing code-mixed natural language processing (NLP) systems. Code-mixed text can pose a significant challenge for NLP systems to handle, and generating high-quality code-mixed text data can enhance the accuracy and effectiveness of such systems.

To analyze code-mixed text from a sociolinguistic perspective, we examine political speeches in various communicative contexts to explore the level of code-mixing at the language and dialect levels. The analysis reveals that speakers' code-mixing is influenced by factors such as the audience, the speaker's ideologies, and the context of the speech.

As part of the sociolinguistic analysis of code-mixing, this thesis has taken into account various dialects to gain a more comprehensive understanding of the phenomenon. Additionally, 50 rules have been formulated to differentiate between the Modern Standard Dialect and the Telangana Dialect in Telugu.

A hybrid methodology is proposed in this thesis to convert a monolingual Telugu sentence to a code-mixed sentence using the rules that differentiate between the Modern Standard Dialect and Telangana Dialect in Telugu, as well as other rule and dictionary-based methods. The methodology incorporates both language-level and dialectal-level code-mixing, resulting in a more nuanced and accurate representation of code-mixed sentences. However, the proposed hybrid methodology for generating code-mixed text has its own difficulties. The methodology requires significant linguistic knowledge, and constructing rules necessitates comprehensive datasets.

To overcome the limitations of the proposed hybrid methodology, we present a novel filtering-based machine translation system that uses synthetic data to generate high-quality code-mixed text. Specifically, the approach involves fine-tuning the mT5 and mBART models for generating Hindi-English and Telugu-English code-mixed texts, respectively. The probabilistic filter used in this approach is effective and language-independent, as demonstrated by its successful application to the low-resource Telugu language. The system produced promising results on various human-generated datasets, achieving code-mixed text that is as close to human-level quality as possible. The proposed filtering-based machine translation system offers an extensible approach for generating high-quality code-mixed text that can be applied to other languages with minimal human effort.

6.2 Challenges

Despite the recent advances in natural language processing, the handling of code-mixed texts/speech presents significant challenges that need to be addressed.

- **Transcribing Speeches:** Transcribing speeches has been challenging, as off-the-shelf techniques often fail to capture dialectal variants in speech, necessitating manual transcription.
- **Telugu-low resource language:**
  - The lack of resources available to differentiate between the Modern Standard and Telangana dialects made it challenging to create a comprehensive set of rules. Therefore, the rules were generated by meticulously analyzing various books and news channels that specifically use the Telangana dialect.
  - In order to implement the neural-based approach for code-mixed text generation in Telugu, a new dataset had to be created, as no existing datasets were available to create the necessary filters.
  - Lack of toolkits, such as CSNLI, that can convert Telugu words from Roman script to Telugu script.

6.3 Future Work

- Developing a toolkit for converting sentences from the Modern Standard Telugu dialect to the Telangana dialect is a potential future work that can be undertaken using the established rules.
- As a future direction, we can explore creating seed data for code-mixing that incorporates languages and dialects using the proposed hybrid methodology. Additionally, we can apply the filtering-based neural machine translation method to this seed data to generate code-mixed text that involves both languages and dialects.

- In the context of future research, it is worth exploring the performance of models trained on a combination of different filtering mechanisms for generating code-mixed text. Furthermore, we can investigate the performance of various downstream tasks that utilize the code-mixed text generated from this approach.
- It is crucial to investigate the effectiveness of filtering techniques for generating high-quality code-mixed data, especially in the case of low-resource languages. Moreover, exploring the application of one-shot and zero-shot learning techniques to determine whether the models are trained to generate code-mixed sentences in general or are specific to the languages they are trained on is also a promising future direction.

Appendix A

The following 50 rules have been curated to differentiate between Telangana and Modern Standard Telugu dialects. The writing convention followed is:

[Standard dialect word] - [Telangana dialect word]:

1. **Vowel rules**

   - **Deletion**:
     - [VD1] In Telangana dialect, vowels are dropped at the end of some words. For example:

       nenu - nen
     - [VD2] Vowels are dropped at the end of some words.

       loVpata - loVpta
   - **Addition**:
     - [VA1] The vowel 'aM' is added to some words at the end of a syllable

       wAbElu - wAaMbElu
     - [VA2] In Telangana dialect, words are sometimes changed to make them easier to pronounce by adding vowel 'a' after duplication of the previous consonant.

       unxi - unnaxi

       unxo - unnaxo
   - **Replacement**:
     - [VR1] Long vowels are replaced with short vowels.

       vaswAru - vaswaru

       ceswAru - ceswaru
     - [VR2]In the Telangana dialect, the vowel 'o' in the preposition 'lo' is changed to vowel 'a'.

       gaxi lo - gaxi la
     - [VR3] The vowel 'a' following the consonant 'v' gets rounded and changes to 'o', while the consonant 'v' itself is dropped

vAdu - odu

prajAsvAmyaM - prajAsomyaM

- [VR4] In Telangana dialect, when a verb's root form has a syllable ending with the consonant 'd' and vowel 'aM', the vowel is replaced with 'u'. Moreover, if the preceding syllable ends with the vowel 'a', it is also modified to 'u'.

  ceyadaM - cesudu

  wAgadam - wAgudu

  winadaM - winudu

- [VR5] In Telangana dialect, verbs ending with *i* are replaces with *e*

  ceyAli - ceyAle

- [VR6]Verbs in the negative form undergo a change where the vowel 'u' at the end changes to 'i'.

  undaxu - undaxi

- [VR7] In the Telangana dialect, when the consonant 'y' follows a vowel, the vowel undergoes palatalization and becomes 'e'. Additionally, duplication of the previous consonant is also observed in

  sAmAnyanga - sAmAnnega

  viRayam - viRRem

- [VR8] For the verbs in past tense, the second last syllable's long vowel in verbs can be replaced based on gender, number, and person.
  - [VR8a] The third person plural masculine form of verbs in past tense, the long vowel in the second last syllable gets replaced by 'in' or 'i', and there may be shortening of the vowel as well

    cesAru - cesinru

    cesAru - cesiru
  - [VR8b] In the third person singular masculine, the long vowel in the second last but one syllable of the past tense verb is replaced by "in".

    pAdAdu - pAdindu

    ayyAdu - ayyindu
  - [VR8c] The second person singular/plural and neuter gender masculine the long vowel in the second last but one syllable of the verb in past tence is replaced by 'ina'

    chesAvA - cesinavA
  - [VR8d] The first person singular/plural and neuter gender masculine the long vowel in the second last but one syllable of the verb in past tense is replaced by 'in'

    pettAnu - pettina

    chesAmu - cesinAm

2. **Consonant rules**

- **Deletion:**
  - [CD1] Words starting with the consonant 'v' can sometimes drop the 'v' sound.

    vachindi - achindi

    vAna - Ana
  - [CD2] When a consonant cluster of identical consonants occurs before a short vowel sound, the second consonant is usually simplified or dropped, resulting in a single consonant. This is generally followed by the application of [VD2]

    oVppukoru - oVpkoru

    ekkuva - ekva
  - [CD3] In Telangana dialect, both nouns and verbs can undergo a phonetic change where the consonant 'r' is dropped when it is followed by a plosive in a consonant cluster

    prAnaM - pAnaM

    brawakadaM - bawakadaM
  - [CD4] Three consonant clusters of ndr, ndl, gny, rnm are reduced by dropping the middle consonant

    ndr - nr : *cUdundri - cUdundri*

    ndl - nl : *bandlu - banlu*

    ndl - nl : *gnyanaM - gyanaM*

    rnm - rm : *gavarnmmeVnt - gavarmeVnt*

- **Addition:**
  - [CA1] In Telangana dialect g is added at the start for a few words. This phenomenon is more prevalent in deixial terms.

    ippuDu - gippuDu

    Ayana - gAyana

    Anthe - ganthe
  - [CA2] For certain verbs, n is added either at the start or in between the syllable

    ceVppAlA ? - ceVppAlna ?

    padukune - pandukune
  - [CA3] In Telangana dialect, the consonant 't' is added to certain words

    alAne - atlAne

    ila - gitla [Simultaneouly CA1 is also applied]
- **Replacement**:
  - [CR1] When two words enter into sandhi, the initial voiceless stop of the second word becomes voiced mostly when the last consonant of the first word is a voiced one or when the first word ends in a vowel

    pedda + kAleV - peddagAleV

    tayAru + cesi - tayArujesi

- [CR2] In Telangana dialect, the aforementioned linguistic patterns are applicable not only during sandhi (a combination of two words) but also applicable within a single word.

  peVttAru - beVttAru

  kadA - gadA

- [CR3] Sometimes the initial 'p' of a word becomes 'v' when the word follows another word

  nIllu + pettadam - nIlluvettadam

  lekka + peVtandi - lekkaveVtandi

- [CR4] Assimilation
  - [CR4a] Consonant 'j' with the preceding 'd' sound in certain words

    idi + jeVsinru - ijjeVsinru

  - [CR4b] consonant 's' assimilates with the preceding 'w' or 'd' sound in certain words

    ceswAru - cewwAru

    iswa - iwwa

- [CR5] A phonetic change known as retroflexization occurs where the consonant cluster 'll' is modified to 'dl', after which the consonant 'n' is added to certain words

  illu - indlu

  kallu - kandlu

- [CR6] the consonant 'r' can be replaced with the retroflex lateral approximant 'l', which is known as retroflexion. This typically occurs when 'r' is followed by a vowel

  evaru - evalu

- [CR7] The consonant 's' changes to the consonant 'R' in some words of Telangana dialect

  lesi - leRi

- [CR8] a consonant 'v' can change to 't' when the previous syllable ends with a vowel 'eV' or 'e'.

  tinevallaki - tinetollaki

  tirigeVvAdu - tirigeVtodu

- [CR9] In Telangana dialect, the consonant 'c' is often replaced with 's' at the beginning of a word

  cuduri - suduri

- [CR9] 'v' can change to 'y' when it is preceded by the vowel 'i'.

  ivaala - iyaala

  kattetivi - kattetiyi

- [CR10] The sound change from voiced dental stop 'd' to voiceless dental stop 't', also known as devoicing, occurs when the preceding syllable ends with a nasal sound

Lexanakunda - lexanakunta

gAkunda - gAkunta

- [CR11] 't' changes to 'x' is known as palatalization. This change typically occurs when 't' is followed by the vowel 'i' or 'e'

  enti - enxi

- [CR12] Palatalization of consonant preceding 'y'
  - [CR12a]

    wy - cy : *sawyaM - sacyaM*

  - [CR12b]

    xy - jy : *uxyogaM - ujyogaM*

3. **Syllable rules**

- **Deletion:**
  - [SD1] Dropping of the syllable which precedes the /d/ sound. In some cases, after the dropping, the preceding vowel is lengthened. This is mostly observed in terms associated with spatial deixis.

    ikkaDa - IDa

  - [SD2] When the second syllable of a structure CV (consonant-vowel) has either 'x' or 'r' as a consonant, the second syllable is dropped.

    UrikeV - UkEv

    moVxalu - moVlu

  - [SD3] In certain words, the second syllable of a word with the structure of CV is often dropped when followed by a long vowel in the first syllable

    bAgAneV - bAneV

    lekapotheV - lepotheV

  - [SD4] For certain words having the second syllable of a structure C1 V C2, where C1 and C2 are the same or similar consonants, it is reduced to only C2.

    eVnanwa - eVnwa

    samanjasanga - sanjasanga

  - [SD5] The third syllable of a word is dropped if it is in the shape of CV and the consonant is one of /k g y v h/

    rUpAyalu - rUpAlu

    caxuvukoVni - caxukoVni

  - [SD6] In case of negative verbs last syllable is dropped

    ceyaledu - ceyAle

    rAledu - rAle

  - [SD7] The dropping of the last syllable is not limited to negative verbs but can occur in certain words as well.

unnAnu - unnA

annaya - anna

- **Replacement:**
  - [SR1] Honorific affixes such as 'andi' can undergo a change in the last syllable based on the gender and number of the person being addressed.

    cudu + andi - cudandi

    * Masculine singular - *cudu + ri - cuduri*
    * Feminine singular - *cudu + dri - cudundri* [CA2 is also applied for easy articulation]
    * Plural - *cudu + ru - cuduru*
  - [SR2] For verbs in Present tense, if the last syllable of a word contains the consonant "r" as its onset and the previous syllable ends with the sequence "nnA", then the "nnA" sequence is dropped. This might be accompanied by the drop of the last vowel and sometimes the reduplication of consonants, particularly "r". However, there might be a partial dropping of the syllable too, where only the "n" is retained.

    gunjuwunnAru - gunjuwur

    anukuntunnAru - anukuturru

    kaduwunnarA - kaduwunrA

Related Publications

- **Dama Sravani**, Lalitha Kameswari and Radhika Mamidi. *"Political Discourse Analysis: A Case Study of Code Mixing and Code Switching in Political Speeches"*. Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching (CALCS 2021), June 11, 2021, NAACL

- **Dama Sravani** and Radhika Mamidi *"Enhancing Code-mixed Text generation in Indian Languages using Filtering of Synthetic Data in Neural Machine Translation"* [1]

Other Publications

- Lalitha Kameswari, **Dama Sravani**, Radhika Mamidi. Enhancing Bias Detection in Political News Using Pragmatic Presupposition. The 8th International Workshop on Natural Language Processing for Social Media (ACL Social NLP 2020). 10th July, 2020. USA.

[1] *Under Review*

Bibliography

[1] G. Abuhakema. Code switching and code mixing in arabic written advertisements: Patterns, aspects, and the question of prestige and standardisation. 2013.

[2] P. Auer and L. Wei, editors. *Handbook of Multilingualism and Multilingual Communication*. De Gruyter Mouton, Berlin, New York, 2007.

[3] U. Barman, J. Wagner, and J. Foster. Part-of-speech tagging of code-mixed social media content: Pipeline, stacking and joint modelling. In *CodeSwitch@EMNLP*, 2016.

[4] R. Bassiouney. Functions of code switching in egypt: Evidence from monologues. 2005.

[5] R. Begum, K. Bali, M. Choudhury, K. Rudra, and N. Ganguly. Functions of code-switching in tweets: An annotation framework and some initial experiments. In *International Conference on Language Resources and Evaluation*, 2016.

[6] H. M. Belazi, E. Rubin, and A. J. Toribio. Code switching and x-bar theory : The functional head constraint. 2008.

[7] S. Berk-Seligson. Linguistic constraints on intrasentential code-switching: A study of spanish/hebrew bilingualism. *Language in Society*, 15:313 – 348, 1986.

[8] N. Bhadwal, P. Agrawal, and V. Madaan. A machine translation system from hindi to sanskrit language using rule based approach. *Scalable Comput. Pract. Exp.*, 21:543–554, 2020.

[9] R. Bhargava, B. V. Tadikonda, and Y. Sharma. Named entity recognition for code mixing in indian languages using hybrid approach. In *Fire*, 2016.

[10] N. Bhaskar. *Bhaskar,*.

[11] I. Bhat, R. A. Bhat, M. Shrivastava, and D. Sharma. Joining hands: Exploiting monolingual treebanks for parsing of code-mixing data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 324–330, 2017.

[12] E. Bokamba. Are there syntactic constraints on code-mixing? *World Englishes*, 8(3):277–292, Nov. 1989.

[13] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, and J. P. McCrae. Corpus creation for sentiment analysis in code-mixed tamil-english text. In *Workshop on Spoken Language Technologies for Under-resourced Languages*, 2020.

[14] I. P. Chakravarthy. *An Annotated Translation of Kalarekhalu A Historical Novel by Ampasayya Naveen*. PhD thesis, The English and Foreign Languages University, Hyderabad, 2016.

[15] C.-T. Chang, S.-P. Chuang, and H. yi Lee. Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In *Interspeech*, 2018.

[16] S. Chen, G. Aguilar, A. Srinivasan, M. Diab, and T. Solorio. Calcs 2021 shared task: Machine translation for code-switched data, 2022.

[17] A. Das and B. Gambäck. Identifying languages at the word level in code-mixed indian social media text. In *ICON*, 2014.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

[19] S. Diwakar, P. Goyal, and R. Gupta. Transliteration among indian languages using wx notation. In *Proceedings of the Conference on Natural Language Processing 2010*, number CONF, pages 147–150. Saarland University Press, 2010.

[20] S. Garg, T. Parekh, and P. Jyothi. Dual language models for code mixed speech recognition. In *Interspeech*, 2017.

[21] S. Garg, T. Parekh, and P. Jyothi. Code-switched language models using dual rnns and same-source pre-training. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

[22] D. Gautam, P. Kodali, K. Gupta, A. Goel, M. Shrivastava, and P. Kumaraguru. CoMeT: Towards code-mixed translation using parallel monolingual sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55, Online, June 2021. Association for Computational Linguistics.

[23] J. J. Gumperz. The sociolinguistic significance of conversational code-switching. *RELC Journal*, 8(2):1–34, 1977.

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

[25] S. Gundapu and R. Mamidi. Word level language identification in english telugu code mixed data. *ArXiv*, abs/2010.04482, 2020.

[26] A. Gupta, S. Menghani, S. K. Rallabandi, and A. W. Black. Unsupervised self-training for sentiment analysis of code-switched data. In *CALCS*, 2021.

[27] A. Gupta, S. Menghani, S. K. Rallabandi, and A. W. Black. Unsupervised self-training for sentiment analysis of code-switched data. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 103–112, Online, June 2021. Association for Computational Linguistics.

[28] D. K. Gupta, A. Ekbal, and P. Bhattacharyya. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings*, 2020.

[29] S. Gupta and P. Pakray. Code-mixing on social media: Strategies and implications. *Languages*, 6(2):43, 2021.

[30] G. A. Guzmán, J. Ricard, J. Serigos, B. E. Bullock, and A. J. Toribio. Metrics for modeling code-switching across corpora. In *Interspeech*, 2017.

[31] B. Haddad, Z. Orabe, A. Al-Abood, and N. Ghneim. Arabic offensive language detection with attention-based deep neural networks. In *OSACT*, 2020.

[32] M. Hadei, V. C. Kumar, and K. S. Jie. Social factors for code-switching-a study of malaysian-english bilingual speakers. *International Journal of Language and Linguistics*, 4:122, 2016.

[33] A. Hegde and S. H. Lakshmaiah. Mucs@mixmt: Indictrans-based machine translation for hinglish text. In *Conference on Machine Translation*, 2022.

[34] B. M. Ilic and M. Radulovic. Commissive and expressive illocutionary acts in political discourse. *Lodz Papers in Pragmatics*, 11(1):19, 2015.

[35] A. Jamatia, A. Das, and B. Gambäck. Deep learning-based language identification in english-hindi-bengali code-mixed social media corpora. *Journal of Intelligent Systems*, 28:399 – 408, 2019.

[36] G. Jawahar, E. M. B. Nagoudi, M. Abdul-Mageed, and L. Lakshmanan, V.S. Exploring text-to-text transformers for English to Hinglish machine translation with synthetic code-mixing. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 36–46, Online, June 2021. Association for Computational Linguistics.

[37] B. B. Kachru. Toward structuring code-mixing: An indian perspective. 1978.

[38] L. Kameswari and R. Mamidi. Political discourse analysis: A case study of 2014 andhra pradesh state assembly election of interpersonal speech choices. In *PACLIC*, 2018.

[39] Z. Kampf and T. Katriel. Political condemnations: Public speech acts and the moralization of discourse. *The handbook of communication in cross-cultural perspective*, 312:324, 2016.

[40] N. Kamwangamalu. Code-mixing across languages: Structure, functions, and constraints. 1989.

[41] B. Krishnamurti, P. Sarma, and K. Civam. *A Basic Course in Modern Telugu*. sole distributors Motilal Banarsidass, Delhi, 1968.

[42] A. Kunchukuttan, A. Mishra, R. Chatterjee, R. M. Shah, and P. Bhattacharyya. Shata-anuvadak: Tackling multiway translation of indian languages. In *International Conference on Language Resources and Evaluation*, 2014.

[43] S. S. V. Kusampudi, P. Sathineni, and R. Mamidi. Sentiment analysis in code-mixed Telugu-English text with unsupervised data normalization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 753–760, Held Online, Sept. 2021. INCOMA Ltd.

[44] P. Lamabam and K. Chakma. A language identification system for code-mixed english-manipuri social media text. *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 79–83, 2016.

[45] Y. Li and P. Fung. Code-switch language model with inversion constraints for mixed language speech recognition. In *International Conference on Computational Linguistics*, 2012.

[46] M. Lichouri and M. Abbas. Machine translation for zero and low-resourced dialects using a new extended version of the dialectal parallel corpus (padic v2.0). In *International Conference on Natural Language and Speech Processing*, 2021.

[47] K. M. Lingam, E. R. Lakshmi, and L. R. Theja. Rule-based machine translation from english to telugu with emphasis on prepositions. *2014 First International Conference on Networks & Soft Computing (ICNSC2014)*, pages 183–187, 2014.

[48] K. M. Lingam, E. Ramalakshmi, and S. Inturi. English to telugu rule based machine translation system: A hybrid approach. *International Journal of Computer Applications*, 101:19–24, 2014.

[49] K. M. Lingam and L. Ravitheja. A hybrid rule-based machine translation system from english to telugu. 2013.

[50] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

[51] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

[52] S. Mandal, S. K. Mahata, and D. Das. Preparing bengali-english code-mixed corpus for sentiment analysis of indian languages. *ArXiv*, abs/1803.04000, 2018.

[53] S. Martinez Guillem. Argumentation, metadiscourse and social cognition: organizing knowledge in political communication. *Discourse & Society*, 20(6):727–746, 2009.

[54] C. Myers-Scotton. Code-switching as a communicative strategy in conversation. *New York, Oxford University Press*, 1993.

[55] C. Myers-Scotton. Social motivations for codeswitching: Evidence from africa. 1994.

[56] B. G. Patra, D. Das, and A. Das. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *ArXiv*, abs/1803.06745, 2018.

[57] C. Pfaff. Constraints on language mixing: Intrasentential code-switching and borrowing in spanish/english. *Language*, 55:291, 1979.

[58] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[59] M. V. Reddy and M. Hanumanthappa. Indic language machine translation tool: English to kannada/telugu. 2013.

[60] M. S. Z. Rizvi, A. Srinivasan, T. Ganu, M. Choudhury, and S. Sitaram. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online, Apr. 2021. Association for Computational Linguistics.

[61] S. Sai and Y. Sharma. Siva@hasoc-dravidian-codemix-fire-2020: Multilingual offensive speech detection in code-mixed and romanized text. In *Fire*, 2020.

[62] J. Sastry. A study of telugu regional and social dialects : a prosodic analysis. 1987.

[63] J. R. Searle, F. Kiefer, M. Bierwisch, et al. *Speech act theory and pragmatics*, volume 10. Springer, 1980.

[64] T. Sellam, D. Das, and A. P. Parikh. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*, 2020.

[65] M. A. Sghaier and M. Zrigui. Rule-based machine translation from tunisian dialect to modern standard arabic. In *International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, 2020.

[66] K. Singh, I. Sen, and P. Kumaraguru. Language identification and named entity recognition in hinglish code mixed tweets. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

[67] R. Singh. Grammatical constraints on code-mixing: Evidence from hindi-english. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 30:33 – 45, 1985.

[68] V. Srivastava and M. K. Singh. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *ArXiv*, abs/2107.03760, 2021.

[69] I. Tarunesh, S. Kumar, and P. Jyothi. From machine translation to code-switching: Generating high-quality code-switched text. *ArXiv*, abs/2107.06483, 2021.

[70] R. Torjmen and K. Haddar. Translation system from tunisian dialect to modern standard arabic. *Concurrency and Computation: Practice and Experience*, 34, 2021.

[71] J. Treffers-Daller. Mixing two languages : French-dutch contact in a comparative perspective. 1994.

[72] T. A. Van Dijk. *Discourse and knowledge: A sociocognitive approach*. Cambridge University Press, 2014.

[73] K. S. S. Varma, A. Chaluvadi, and R. Mamidi. Corpus creation and language identification in low-resource code-mixed telugu-english text. In *Recent Advances in Natural Language Processing*, 2021.

[74] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury. Pos tagging of english-hindi code-mixed social media content. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

[75] M. Warschauer, G. R. E. Said, and A. Zohry. Language choice online: Globalization and identity in egypt. *J. Comput. Mediat. Commun.*, 7:0, 2006.