On the Development of Hindi-English Code-Switching Speech Recognition Systems and Corpus

Ganji Sreeram

On the Development of Hindi-English Code-Switching Speech Recognition Systems and Corpus

A

thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

by

GANJI SREERAM

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

June 2020

Certificate

This is to certify that the thesis entitled "**On the Development of Hindi-English Code-Switching Speech Recognition Systems and Corpus** ", submitted by **Ganji Sreeram**, Roll No. 156102028, a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for the submission. The results embodied in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

30th June, 2020
Guwahati.

Dr. Rohit Sinha
Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

To,

My Parents

(***Shri. Bala Krishna Rao and Smt. Jyothi***)

and

My Wife

(***Sri Katyayani***)

Acknowledgements

This thesis would not have been possible without the help and support of several people in various measures. I take this opportunity to express my sincere acknowledgments to all of them.

I express my deepest and most sincere gratitude to my thesis supervisor Prof. Rohit Sinha for his guidance and constant encouragement. His insightful feedbacks have helped me very much in improving the quality of my thesis. I greatly admire his attitude towards research, creative thinking, and enthusiasm for work. It is truly a blessing to have a supervisor who cares so much about my work and who is always there for me in times of need. I shall forever remain indebted to him.

I am grateful to Prof. P K Bora, the chairman of my doctoral committee, for his support and encouragement throughout my Ph.D. period. I am thankful to the other members of my doctoral committee, Dr. Tony Jacob, Dr. Prithwijit Guha, and Dr. S Ranabir Singh, for sparing their time for evaluating my work. I thank Prof. Samudravijaya, Dr. Priyankoo Sarmah, Prof. S Dandapat, and Prof. S R M Prasanna for their valuable suggestions on my work. I would also like to thank all other faculty members of the Department of Electronics and Electrical Engineering, IIT Guwahati, for their care and support. I am also very thankful to all the technical, office, security, canteen, and maintenance staff members of the department for their help when required.

I am beholden to my seniors (Dr. Haris B C, Dr. Syed Shahnawazuddin, Dr. O P Singh, Dr. Nagendra Kumar, Dr. Suman Deb, Dr. Ramesh K Bhukya) and my dear friends (Dr. Bidisha Sharma, Dr. Subhasis Mandal, Himakshi Choudhury, Prabhakar Eedara, Vineeta Das, Saswati Rabha) and all other members in Signal Informatics and EMST Laboratories. I am also thankful to all other friends of the department for their help and support.

A special thanks to my badminton and chit-chat buddies (Dr. Vivek Venugopal, Dr. Jitendra Prajapati, Gargi Baruah, Uddipana Dowarah, Wg. Cdr. G K Thiyagasundaram, Trusna Meher, Dr. Niladri Das, Kamakshi Manjari, Dr. Charudatt Y Kadolkar) for being my stressbusters. I can not imagine my life at IIT Guwahati without these guys. Also, I would like to extend my thanks to other badminton partners (Prof. M K Bhuyan, Dr. S K Nayak, Prof. Ajay K, Dr. Arindam Dey, Dr. Soumitra Nandi), my photography and swimming buddies (Prem Sagar, Dr. Robin George). They helped me in balancing both physical and mental fitness throughout my Ph.D. life. I want to thank all the committee members of the IEEE Student Branch, IIT Guwahati, for providing me a

platform to explore and develop communication skills and leadership qualities.

I wish to acknowledge with gratitude for the partial financial assistance received towards data collection from a project grant No. 11(18)/2012-HCC(TDIL) from the Ministry of Electronics and Information Technology, Govt. of India.

Last but not least, my deepest gratitude goes to my parents and my beloved wife. Without their love, support, and sacrifice, I wouldn't have been able to complete my PhD.

Ganji Sreeram

Abstract

Code-switching refers to the alternate use of two or more languages (or dialects) during the conversation. This phenomenon has been observed in many multilingual communities across the globe. Therefore, handling code-switching by the spoken input systems is very much required for efficient human-machine interaction. However, due to the lack of domain-specific resources, the research in this domain is somewhat limited compared to the monolingual case. This thesis aims to address the acoustic and language modeling challenges in code-switching automatic speech recognition (ASR) tasks. In addition to that, a Hindi-English code-switching corpus has been created towards addressing the data scarcity issue.

The early works on code-switching ASR happen to employ the hybrid framework typically developed for the monolingual case. The created Hindi-English code-switching corpus is first evaluated in the hybrid framework. The hybrid framework comprises of three sub-modules, namely, a pronunciation model, an acoustic model, and a language model. The end-to-end (E2E) framework has recently emerged as a viable alternative to the hybrid systems in the ASR domain. Unlike the hybrid framework, the E2E framework does not require the phonetically labeled training data, and also does not include any explicit pronunciation model. In the case of code-switching ASR, for multiple languages being involved, these attributes become more attractive. Motivated by that, in this thesis, the E2E framework has been explored for developing the code-switching ASR systems.

In the existing code-switching E2E ASR works, the target set is derived by merely combining the character sets of the languages involved. Such systems would suffer from high confusability among the cross-language targets due to the broad acoustic similarity among sound units involved in the code-switching language pairs. To avoid such a confusability, a common phone set covering the underlying languages in code-switching is defined and used as the reduced target set for

training the models. Interestingly, the reduced target set based E2E ASR system outperformed the combined target set one in terms of the target error rate (TER). But, a reverse trend was noted when those target sequences were converted to word sequences, i.e., for computing the word error rate (WER). This degradation in WER is because of the enhanced confusability among the homophones (the words having identical pronunciation but different spellings) within or across the languages involved. For addressing the same, a context-dependent target-to-word (T2W) transduction scheme has also been proposed, which employs an explicit error model and a language model. The proposed T2W transduction scheme is noted to achieve a relative improvement of 22% over the naive transduction scheme in the context of Hindi-English code-switching E2E ASR. Further, to enhance the context information in the code-switching data, a novel textual feature referred to as the code-switching location (CSL) feature and a modified parts-of-speech (POS) tagging scheme have also been proposed. On evaluating these features by incorporated into factored language model (FLM), a significant reduction in perplexity score has been noted. With the use of these FLMs in the proposed T2W transduction scheme, a further improvement in the WER score is achieved. The proposed system outperforms the existing one and yields a TER of 18.1% along with a WER of 29.79% on the created Hindi-English code-switching corpus. Despite the proposed approaches being evaluated for the Hindi-English code-switching case, they are generic enough to be applied for any other code-switching context.

Keywords: Code-switching, speech recognition, language modeling, end-to-end system, factored language model, target-to-word transduction.

Contents

List of Figures **xvii**

List of Tables **xxi**

List of Acronyms **xxv**

1 Introduction **1**
1.1 Evolution of ASR . . . 3
1.2 ASR Paradigms . . . 5
1.2.1 HMM-based Framework . . . 5
1.2.2 End-to-End Framework . . . 7
1.3 Motivation for Research Work . . . 8
1.4 Contributions of the Work . . . 10
1.5 Organization of the Thesis . . . 11

2 Scope and Literature Review of Code-Switching **13**
2.1 Hindi-English Code-Switching . . . 14
2.2 Existing Works on Code-Switching . . . 16
2.3 Existing Code-Switching Databases . . . 18
2.4 Conclusions . . . 21

3 Hindi-English Code-Switching (HingCoS) Corpus **23**
3.1 HingCoS Text Corpus . . . 25
3.1.1 Steps Involved in Text Normalization: . . . 26
3.2 HingCoS Speech Corpus . . . 27
3.3 Lexical Resources . . . 30
3.4 Statistical Analysis of the HingCoS Corpus . . . 32
3.4.1 Analysis of the HingCoS Text Corpus . . . 32
3.4.2 Analysis of the HingCoS Speech Corpus . . . 36
3.5 Comparison with Existing Code-switching Corpora . . . 37
3.6 Experimental Setups and Evaluations . . . 39
3.6.1 Acoustic and Linguistic Datasets . . . 39
3.6.2 Front-End Features . . . 40
3.6.3 Acoustic Model Training . . . 41
3.6.4 Language Model Training . . . 42
3.6.5 Parameter Tuning . . . 44
3.6.6 Evaluation Results . . . 46
3.7 Conclusions . . . 47

4 Exploration of End-to-End Framework for Code-Switching Speech Recognition 49
4.1 Variants of E2E ASR Frameworks . . . 50
4.1.1 CTC-based E2E Framework . . . 50
4.1.1.1 DBLTSM Network . . . 50
4.1.1.2 CTC Cost Function . . . 52
4.1.2 Attention-based E2E Framework . . . 52
4.2 E2E Hindi-English Code-Switching ASR Systems . . . 54
4.2.1 Combined Target Set Modeling . . . 54
4.2.2 Reduced Target Set Modeling . . . 56
4.2.2.1 Naivety in T2W transduction . . . 58
4.2.2.2 Homophone confusability . . . 58
4.3 Context-Dependent T2W Transduction . . . 60
4.4 Experimental Setup . . . 61
4.4.1 Database Preparation . . . 62
4.4.2 System Description and Parameter Tuning . . . 63
4.4.2.1 Attention-based E2E ASR system . . . 63
4.4.2.2 CTC-based E2E ASR system . . . 64
4.4.2.3 RNN-based LMs . . . 64
4.5 Experimental Results . . . 65
4.5.1 Evaluation of the T2W Transduction . . . 65
4.5.2 Computational Complexity . . . 66
4.6 Conclusions . . . 67

5 Language Modeling of Code-Switching Text Data 69
5.1 Review of Factored Language Model . . . 71
5.1.1 Factored n-gram LM . . . 71
5.1.2 Factored RNNLM . . . 73
5.2 Improved POS Textual Features . . . 74
5.2.1 Incorporating POS Features in Code-Switching LM . . . 75
5.2.2 Incorporating POS Features in Monolingual LM . . . 77
5.3 Proposed Code-Switching Location Textual Features . . . 79
5.3.1 Incorporating CSL Features in Code-Switching LM . . . 79
5.3.2 Incorporating CSL Features in Monolingual LM . . . 80
5.4 Experimental Setup . . . 85
5.4.1 Database Preparation for Exp-1 . . . 85
5.4.2 Database Preparation for Exp-2 . . . 86
5.4.3 Parameter Tuning . . . 87
5.5 Results and Discussion . . . 88
5.5.1 Evaluation by Incorporating Proposed Features in Code-Switching Hindi-English Data . . . 88
5.5.2 Evaluation by Incorporating Proposed Features in Monolingual Hindi Data . . . 89
5.5.3 Studying the Robustness of the Proposed Approach . . . 91
5.6 Revalidation on Mandarin-English Code-Switching Data . . . 92
5.6.1 Evaluation by Incorporating Proposed Features in Code-Switching Mandarin-English Data . . . 94
5.6.2 Evaluation by Incorporating Proposed Features in Monolingual Mandarin Data 95
5.7 Assessment of the Proposed Textual Features in T2W Transduction . . . 96
5.7.1 Evaluation of T2W Transduction with Code-Switching fRNNLM . . . 96

5.7.2 Evaluation of T2W Transduction with Monolingual fRNNLM 97
5.8 Conclusion 98

6 Conclusions **99**
6.1 Summary of the Work 100
6.2 Summary of Contributions 103
6.3 Future Directions 104

A Neural Network Architectures: RNN and LSTM **109**
A.1 RNN Architecture 110
A.2 LSTM Architecture 110

Bibliography **113**

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

List of Figures

1.1 Block diagram representing the structure of a hybrid ASR framework. It is a modular architecture, where the language model, the acoustic model, and the pronunciation model are optimized independently with different objective functions. 5
1.2 Block diagram representing the structure of an E2E ASR framework. All the modules in this framework are jointly optimized with a common objective function. 7

2.1 Distributions of Indian language families as per the 1991 census. The figures in bracket refer to the number of languages (inclusive of mother tongues grouped under them) in each language family. It can be noted that 40.2% of the population are native Hindi speakers. The image is adapted from http://www.ciil-lisindia.net/. Copyright 2011 c Central Institute of Indian Languages. 15

3.1 Illustration of salient text normalizations described in Section 3.1.1 that have been applied to raw Hindi-English data collected from the web sources. Note that, <s> and </s> represents the sentence begin and end markers, respectively. 27
3.2 The call flow of the voice-server used by the volunteers for the session-wise recording of Hindi-English transcripts. Each of the volunteers was given 100 sentences and was asked to record them in 5 difference sessions with each session have 20 sentences. . . 28
3.3 An example of the typical recorded Hindi-English speech utterance marked with native and non-native word/phrase boundaries along with their corresponding language identity labels. The short-hand notations 'Hin' and 'Eng' are used to denote Hindi and English words/phrases, respectively. 29
3.4 The distribution of the vocabulary based on the pronunciation count. Note that, the lexicon contains a total of 8,911 entries out of which 6,616 entries are unique. 32
3.5 Distribution of code-switching instances for varying length of the sentences in the HingCoS text corpus. For the ease in display, the counts in each sentence length group are normalized separately, i.e., the sentence counts in a length group across 8 code-switching instances considered sum up to 100%. 34
3.6 Plot of the average code-switching instances with respect to the length of sentences in HingCoS text corpus. It is to highlight that, in each of the sentences in HingCoS corpus about 20% (approx.) of the words are being code-switched. 35
3.7 Distributions of the English and Hindi Words in the HingCoS text corpus based on their parts of speech (POS) labels. Note that, for the ease in display, the POS tags for Hindi and English words have been normalized separately, i.e., all POS labels corresponding to English/Hindi words sum up to 100%. 35

3.8 Distribution of the speakers' age along with the gender, is shown as bar-plot. Whereas, distribution of mother tongues (native language) of the speakers in the collected Hindi-English speech data is shown as pie-chart. The Hindi forms the mother tongue of the majority of speakers and is followed by Assamese and Bangla native speakers. 36
3.9 Distribution of the native states of the speakers in the collected speech data. The majority of speakers are from Assam and is followed by Uttar Pradesh. A total of 22 states are covered in the HingCoS speech corpus. The states of India represented in the HingCoS speech corpus are marked in the associated map. 37
3.10 Network architecture for the class based RNNLM. The previous context s_{t-1} and the word w_{t-1} are fed as inputs for modeling the present context s_t at the hidden layer. The output layer is factorized to two parts: c_t for the word classes, and o_t for words conditioned on the classes. The X, Y are weight matrices between input and hidden layer, while the P, Q are weights matrices between hidden and output layers. 43
3.11 Tuning of the context length (N) on the development data. The optimal perplexity score is obtained for N = 5. 45

4.1 Architecture of the CTC-based E2E network. The encoder is a deep network consisting of BLSTM cells. Given a target transcription $\boldsymbol{y}$ and the input feature vector $\boldsymbol{x}$, the network is trained to minimize the CTC cost function as CTC($\boldsymbol{x}$) = log P($\boldsymbol{yjx}$). 51
4.2 Architecture of LAS network. It consists of three modules namely: listener (a BiLSTM encoder), attender (an alignment generator), and speller (an LSTM decoder). 53
4.3 Histogram of different types of homophones present in the HingCoS corpus. It is worth noting that a large number of homophones in the HingCoS corpus belong to the inter-language category. 59
4.4 Flow chart of the proposed context-dependent T2W transduction scheme. $\boldsymbol{f}T_{h_i}\boldsymbol{g}_{i=1}^{n}$ denote a hypothesized target sequence having n segments, set $\boldsymbol{f}W_{i_k}\boldsymbol{g}_{k=1}^{m}$ denote all possible (say m) words returned by that search, and the current partial sentence is denoted by S. 61
4.5 The tuning of parameters for attention-based E2E ASR system. (a) selection of number of epochs, and (b) selection of number of nodes in the encoder. The TER saturates after 300 epochs, while it degrades beyond 256 nodes. 64

5.1 Comparison of possible back-off modeling options in the traditional and the factored n-gram LMs. The factored n-gram LM has a wide range of back-off modeling options in addition to those permitted by the traditional n-gram LM. For the ease of depiction, the context length (N) is considered to be 4. Here, F_t, F_{t-1}, F_{t-2} and F_{t-3} are the feature vectors corresponding to the words w_t, w_{t-1}, w_{t-2} and w_{t-3}, respectively. 72
5.2 Two example topologies among many possible back-off options that can be employed in training the FLMs. Here each word w_t has two features represented by M_t and S_t, respectively. 72
5.3 Architecture of the factored RNNLM. Where x, s, and y denote the input, the hidden and the output layers, respectively. 74
5.4 (a) Flow chart of the existing Hindi-English POS tagger. (b) Flow chart of the proposed approach. The key innovation in the proposed approach has been highlighted, which enables us to extract more accurate POS features of English words embedded in Hindi-English data. Note, for the English words not present in the translated version, the POS tags are derived using the existing approach shown in (a). 76

5.5 Flow chart of the scheme employed to tag the code-switching text data with the proposed CSI feature. In this work, English-to-Hindi translation is done by employing the *Google Translate*, an online machine translation tool. 80
5.6 Flow chart of a novel tagger developed for the CSL tagging of Hindi text data. This flow chart comprises two modules: code-switching (CS) word pair generation and CSL tagging. Note that, this tagger allows us to develop a monolingual Hindi LM that can handle the Hindi-English code-switching data. 84
5.7 Distributions of the English and Hindi Words in the Hindi-English text corpus based on their parts of speech (POS) labels. Note that, for the ease in the display, the POS tags for Hindi and English words are normalized separately, i.e., all POS tags corresponding to English/Hindi words sum up to 100%. 85
5.8 Tuning of the context length (N) on the development data. The optimal PPL score is obtained for N = 5. 87
5.9 Distributions of the English and Mandarin Words in the Mandarin-English text corpus based on their parts of speech (POS) labels. Note that, for the ease in display, the POS tags for Mandarin and English words have been normalized separately, i.e., all POS labels corresponding to English/Mandarin words sum up to 100%. 94
5.10 Assessment of the impact of proposed context-dependent T2W transduction with/without textual features on attention- and CTC-based E2E ASR systems. 97

6.1 Creation of character-level LID tags for the training data towards conditioning the E2E networks to perform LID task on code-switching speech. The 'H/E' denotes the Hindi/English LID tag. The 'b/e' label is appended to the 'H/E' LID tag to mark the begin/end characters. 105
6.2 Visualization of attention mechanism for LID task. For a given Hindi-English code-switching utterance: a) spectrogram labeled with Hindi and English word boundaries for reference purpose. (b) variation of attention weights with respect to time for Hindi and English languages, and (c) alignment produced by the attention network for the input speech and the decoded output LID labels. 106

A.1 Block diagrams of the unfolded network architecture of (a) the RNN, and (ii) the LSTM. 110

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

List of Tables

2.1 Example Hindi-English sentences along with their English translated versions showing the inter-sentential code-switching and the variants of the intra-sentential code-switching. Type-1 and Type-2 variants of intra-sentential code-switching refer to high and low contextual information being carried by the non-native (English) words, respectively. 16

3.1 Quantifying the impact of text normalization of the raw text data as described in Section 3.1.1. The non-characters include punctuation marks, braces, numerals, mathematical symbols, emoticons, etc. 26

3.2 Validation of the text normalization process. In this experiment, two language models (LMs) are created using the raw and the normalized versions of the training set. For performance evaluation, a development and a test datasets which are non-overlapping to the training dataset, are also created. The corresponding perplexities of the LMs trained on raw- and normalized-text are reported. Note that, the out of vocabulary rate of the development and the test datasets with respect to the normalized training dataset turns out to be 16.9% and 18.2%, respectively. 27

3.3 The IRPAbet consisting of 62 labels defined in this work for labeling the sounds in both Indian-English and Hindi languages. For Indian-English, the CMU ARPAbet labels are assigned with Hindi phone set based on perceptual similarity, while a few new labels introduced to cover Indian-English are marked in gray colour. Owing to the Indian accent, many ARPAbet labels get mapped to a single IRPAbet label. . . 31

3.4 Key statistics of the HingCoS text corpus developed in this study. Note that, the code-switching instance refers to the location where switching happens either from Hindi to English or vice-versa. 33

3.5 Top 10 most frequent code-switching word pairs (Hindi-to-English and vice versa) that occur in HingCoS text corpus along with their corresponding log-likelyhood probabilities. 33

3.6 The details of the HingCoS speech corpus developed in this study. Note that, the locations where switching happens either from Hindi to English or vice-versa are referred to as the code-switching instances. 36

3.7 Contrastive comparison of existing code-switching speech databases reported in the literature including the HingCoS speech corpus described in this work. The developed HingCoS corpus consists of 25 hours of read speech data from 101 speakers and is definitely one among the biggest code-switching corpora reported so far in the literature. 38

3.8 Contrastive comparison of existing code-switching text databases reported in the literature including the HingCoS text corpus described in this work. 38

3.9 The parameters used for training the DNN and TDNN based hybrid models that are employed in this study. Note that the default parameters set by the employed toolkit are used for training the DNN model, while the parameters for TDNN model are set such that the computational complexity does not exceed the compute resources available at our end. 45
3.10 Recognition performances in terms of perplexity for n-gram LM and RNNLM trained on the Hindi-English training dataset and evaluated on Hindi-English development and test sets. 46
3.11 Evaluation of Hindi-English code-switching speech corpus in context of ASR task. The performance results in terms of percentage word error rate (%WER) along with the 95% confidence interval in brackets are reported for the test set comprising of 1976 sentences as defined in Section 3.6.1. Along with the proposed IRPAbet phone set-based systems, the evaluation has also been done for the systems trained by combining both the Hindi and English phone sets defined in Table 3.3. 47

4.1 The character sets of Hindi (68) and English (26) languages that are used as targets in building the conventional E2E Hindi-English code-switching ASR system. 55
4.2 Evaluation of the E2E ASR system trained on the combined target set for the Hindi-English code-switching task. The WER is computed after the transduction of the hypothesized targets. The percentage of invalid words generated indicate the naiveness of the transduction scheme employed. 55
4.3 Two sample decoded output sequences of the attention-based E2E ASR system developed using a combined target set for the Hindi-English code-switching task. The English translations of the sentences are given in the braces. The errors obtained in the hypothesized character sequences have been highlighted. Note that the symbol '_' is used to mark the word boundaries. The invalid words produced by the transduction process are labeled as *hunki*. 56
4.4 Two sample decoded output sequences of the attention-based E2E ASR system trained on the reduced target set for the Hindi-English code-switching task. For contrast purposes, the sentences are kept the same as considered in Table 4.3. Note that the symbol '_' is used to mark the word boundaries. The invalid words produced by the transduction process are labeled as *hunki*. 57
4.5 Evaluation of the E2E ASR system trained on reduced target set for Hindi-English code-switching task. 57
4.6 Examples of different types of homophones present in the Hindi-English code-switching data. In the HingCoS corpus, there are a total of 96 homophones. 58
4.7 Demonstration of T2W transduction achieved by the naive and the proposed schemes for the reduced target set case. The hypothesized target segments and words in error are shown in the 'red' colour. For the proposed scheme, the highlighted LM score corresponds to the 1-best output. 62
4.8 Salient statistics of text and speech components of the HingCoS corpus. The CS count refers to the number of code-switching instances in the data. 63
4.9 Quality assessment of T2W transduction in terms of %WERs along with 95% confidence interval, obtained for attention- and CTC-based E2E ASR systems developed using reduced and combined target sets. In Naive method, neither an error model (EM) nor a language model (LM) is involved. 65

4.10 The details of the memory usage and the computational time for training the E2E ASR systems using both the reduced and combined target sets. Note that, the reduced target set takes less memory and computational time when compared to the combined target set. This behaviour is attributed to the 34% relative reduction in the target labels that need to be modeled in the former case. 67

5.1 An example Hindi sentence along with a few of its Hindi-English code-switched variants for highlighting the increased word sequence variability due to code-switching. 70
5.2 Example sentences highlighting the broad syntactic rules being followed in Hindi-English code-switching. For the given example Hindi sentence, the English translation is presented below it. 71
5.3 Incorporation of POS information in the modeling of Hindi-English sentences. (a) The existing approach, i.e., employing the English POS tagger to handle English words in Hindi-English sentences while using the Hindi POS tagger for the remaining. On account of the lack of contextual information, the English POS tagger outputs the "Noun" tag for most of the English words in Hindi-English sentences. (b) The proposed approach, i.e., before employing English POS tagger, convert Hindi-English training sentences into English using a machine translator. Note that the POS tag "list" refers to the list of markers and includes surrounding punctuation (as mentioned in Penn Treebank POS Tags). 77
5.4 Typical Hindi-English code-switching sentences in the database and their respective Hindi and English translated versions. Case 1 and Case 2 refer to the presence and absence of one-to-one topological matching between Hindi-English sentence and the corresponding translated Hindi version, respectively. 78
5.5 An example sentence pair highlighting the fact that the POS features extracted for Hindi text remain mostly valid for Hindi-English code-switching text too. The English translation for the given Hindi sentence is "common man suffers more". 78
5.6 The proposed CSL tagged output for an example Hindi-English code-switching sentence. The English translation of example sentence is given in bracket for reference. 80
5.7 The details of the vocabulary size and the word count of the training, development and evaluation datasets created for evaluating the proposed features under two experimental conditions. The Exp-1 explores training LMs on code-switching data while the Exp-2 explores augmenting monolingual LM to deal with code-switching data. Note that the Dev-1 and Eval-1 datasets are used for parameter tuning and evaluation of both Exp-1 and Exp-2 conditions 86
5.8 Evaluation of Exp-1 setup, i.e., the incorporation of proposed textual features in the modeling of code-switching Hindi-English data. Recognition performances in terms of PPL for various FLMs trained on Hindi-English Train-1 dataset incorporating different combinations of features when evaluated on Hindi-English Eval-1 dataset. Note that POS_E, and POS_P refers to the POS features of Hindi-English code-switching data obtained by employing the existing approach and the proposed approach, respectively. The best performances obtained are highlighted. 89

5.9 Evaluation of Exp-2 setup i.e., the incorporation of proposed textual features in the modeling of monolingual Hindi data. The performances (in terms of PPL) of the POS and the proposed CSL features in training the Hindi RNNLM in the context of Hindi-English code-switching task. Since the Hindi RNNLMs are being tuned on Hindi-English Dev-1 dataset, those performances are for reference purpose only. Also, the performances on 5-gram LM are given for contrast purposes. The best performances obtained are highlighted. 90
5.10 Typical Hindi-English code-switching sentences in the database and their respective English translated and retranslated versions. For simulating the translation errors, the English translation of a Hindi-English sentence produced by the Google Translate tool is retranslated to the Hindi language and then again translated back to English. Thus, two logically similar English translations of the given Hindi-English sentence are produced. 92
5.11 Assessment of errors in translation for the proposed POS tagging approach. For simulating the errors during machine translation, varying percent of direct-English translated versions of Hindi-English sentences in the Train-1 are retranslated to Hindi and then translated back to English. The sentence error rates of such retranslations are computed with respect to the direct-English translations of Hindi-English sentences and those measure the effectiveness of the employed process in introducing the errors in the direct-English translation. The missed word rate quantifies the amount of the missed English words in the translated data. 92
5.12 The details of the vocabulary size and the word count of the training, development and evaluation datasets created for re-evaluating the proposed features under two different experimental conditions. 93
5.13 Evaluation of Exp-1 setup i.e., the incorporation of proposed textual features in the modeling of code-switching Mandarin-English data. PPL scores for various FLMs trained on Mandarin-English training dataset incorporating different combinations of features when evaluated on Mandarin-English Eval-1 dataset. The best performances obtained are highlighted. 95
5.14 Evaluation of Exp-2 setup i.e., the incorporation of proposed textual features in the modeling of monolingual Mandarin data. PPL scores of the POS and the CSL features in training the Mandarin RNNLM in context of Mandarin-English code-switching task. Since, the Mandarin RNNLMs are being tuned on Mandarin-English Dev-2 dataset, those performances are for reference purpose only. Also, the performances on 5-gram LM are given for contrast purposes. 95
5.15 Assessment of textual feature augmented Hindi-English code-switching FLMs on the proposed T2W transduction scheme for E2E ASR systems trained on reduced/combined target set. The recognition performances are reported in terms of percentage word error rate (% WER) along with the 95% confidence interval in brackets 96
5.16 Assessment of textual feature augmented monolingual Hindi FLMs on the proposed T2W transduction scheme for E2E ASR systems trained on reduced/combined target set. The recognition performances are reported in terms of percentage word error rate (% WER) along with the 95% confidence interval in brackets 98

List of Acronyms

AM	acoustic model
ANN	artificial neural network
ASR	automatic speech recognition
BLSTM	bidirectional long short term memory
BPTT	back propagation through time
CSL	code-switching location
CTC	connectionist temporal classification
DBLSTM	deep bidirectional long short term memory
DNN	deep neural network
E2E	end-to-end
EM	error model
EMST	Electro-Medical and Speech Technology
FBANK	filter-bank
FDNN	feed-forward neural network
FLM	factored language model
fMLLR	feature-space maximum likelihood linear regression
fRNNLM	factored recurrent neural network based language model
GA	genetic algorithm
GMM	Gaussian mixture model
HMI	human-machine interaction
HMM	hidden Markov model
HTK	hidden Markov model toolkit
IITG	Indian Institute of Technology Guwahati

IRPAbet	Indian real pronunciation alphabets
LAS	listen, attend and spell
LDA	linear discriminant analysis
LID	language identification
LM	language model
LPC	linear prediction coefficient
LSTM	long short term memory
MER	mixed error rate
MFCC	mel-frequency cepstral coefficient
MLLT	maximum likelihood linear transform
MSR	Microsoft Research
MTL	multi-task learning
MT	machine translation
NLP	natural language processing
OOV	out-of-vocabulary
PBCM	phonetically balanced code-mixed
PDP	parallel distributed processing
PM	pronunciation model
POS	parts-of-speech
PPL	perplexity
RNNLM	recurrent neural network based language model
RNN	recurrent neural network
RNN-T	recurrent neural network transducer
SAT	speaker adaptive training
SGMM	subspace Gaussian mixture model
T2W	target-to-word
TDNN	time delay deep neural network
TER	target error rate
T-matrix	total variability matrix

UBM	universal background model
VAD	voice activity detection
WER	word error rate
WFST	weighted finite state transducer
WSJ	Wall Street Journal

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

1

Introduction

Contents

1.1 Evolution of ASR 3
1.2 ASR Paradigms 5
1.3 Motivation for Research Work 8
1.4 Contributions of the Work 10
1.5 Organization of the Thesis 11

Automatic speech recognition (ASR) refers to the task of speech to text conversion by machines [1]. The main objective of an ASR system is to output the spoken word tokens given the input utterances [2]. Over the past few decades, the scope of ASR has spread across a wide range of applications that involve human-machine interaction (HMI). A few examples of those applications include speech-based information retrieval, speech-based web searches, reading tutors, language learning tools, and entertainment. With the growth of technology, the ASR systems are deployed on mobile phones, automobiles, televisions, etc., and thus play an essential role in our day-to-day lives. The speech input of the user accessing the ASR system contains not only the message but also the information about age, gender, health, social/regional association, and emotional state of the speaker [3]. Besides these inter- and intra-speaker acoustic variabilities, a speech signal is also affected by other factors such as variability in the environment (channel and background), and linguistics (vocabulary size, context mismatch, and code-switching). Therefore, in all the applications mentioned above, the significant variabilities caused by these factors make the task of ASR more challenging. As a result of that, the typical ASR systems happen to incorporate different adaptation and normalization techniques to achieve robust recognition performance. The effective handling of each of the variabilities mentioned above in speech signals is an independent research problem. Most of the existing ASR systems are developed and optimized for monolingual speech recognition. There is still a considerable gap between human and machine speech recognition performance [4, 5], particularly when the spoken utterances involve multiple languages, i.e., code-switching.

In this work, we focus on developing the ASR system that can handle code-switching. Code-switching refers to switching or mixing of the non-native (foreign) language words/phrases into the native language sentences while conversating [6–8]. With urbanization and globalization, code-switching has become a common phenomenon in multilingual communities across the world. The recent works [9, 10] have highlighted that code-switching is also observed in the textual chats, comments, and messages posted on social media sites like Facebook, Twitter, WhatsApp, YouTube, etc. Therefore, there is a greater need to handle code-switching by the spoken and textual input systems for effective HMI. The thesis begins with a brief review of the evolution of ASR. We then highlight the challenges in developing the ASR systems for code-switching data. Following that, we present the contributions and outline of the thesis.

1.1 Evolution of ASR

The technology behind speech recognition has been in development for over a century. But, until mid 20^{th} century, the pace of the research in ASR domain was slow [11]. The first speech recognition system *Audrey* was developed by Davis, Biddulph, and Balashek at Bell Laboratories in 1952 [12]. It was built for isolated digit recognition for a single speaker, using the formant frequencies estimated from the vowel regions of each digit. Later, Olson and Belar of RCA Laboratories built a system to recognize ten syllables of a single speaker [13] and, Forgie and Forgie of MIT Lincoln Lab built a speaker-independent ten vowel recognizer [14]. Other recognition systems of the 1960s are a vowel recognizer by Suzuki and Nakata of the Radio Research Lab in Tokyo [15], digit recognizer of NEC Laboratories [16]. Also, in 1962, IBM demonstrated *Shoebox*, a system that could recognize digits and arithmetic commands such as plus and total. In 1976, Carnegie Mellon University developed a speech recognizer *Harpy*, which could recognize over 1000 words. With the introduction of the hidden Markov model (HMM) [17] into speech recognition, the research in this area has seen tremendous growth. An HMM consists of two stochastic processes, namely, an invisible process of hidden states and a visible process of observation symbols. The hidden states form a Markov chain, and the probability distribution of the observed symbol depends on the underlying state. Given a set of labeled utterances that can represent the majority of variations of the acoustic units such as phonemes and words, usually, the Baum-Welch algorithm [18] is employed to estimate the parameters of the corresponding model. The estimation of parameters is referred to as the training or the learning phase. During the testing or the decoding phase, the resulting model provides the maximum-likelihood estimate [19] that helps in recognizing the unknown utterance, using the Bayes' decision theory [20]. In the early 1970s, Jim Baker from CMU first applied HMMs to speech recognition tasks. Since then, the HMMs have been explored extensively and emerged as the dominant technique for acoustic modeling in ASR systems [21, 22].

In the 1950s, though the idea of artificial neural network (ANN) was first introduced, initially, it could not produce remarkable outcomes [23]. In the 1980s, with the advent of the parallel distributed processing (PDP) model and an efficient training approach called error backpropagation, the interest in mimicking the human neural processing mechanism was re-initiated. Thus, the multi-layer perceptron was introduced, which has the capability of approximating any function to arbitrary

precision, provided no limitation in the processing complexity was imposed. In the early attempts of employing ANNs in speech recognition, the research is mainly on simple tasks like recognizing a few phonemes or a few isolated words [24–27]. However, as the practical ASR systems inevitably require handling of temporal variations in the spoken utterance, the ANNs in their original form have not proven to be extensible. Therefore, the research is then focused on integrating the ANNs with HMM to take advantage of temporal modeling capability of the HMM [25, 28, 29].

After successfully addressing the isolated word recognition, the focus of the research is shifted towards the large vocabulary continuous speech recognition (LVCSR) tasks involving several thousands of words. As systems became more sophisticated, i.e., including several thousands of models and millions of parameters to be learned, a well-structured baseline software system was very much essential for further research and development to incorporate new concepts and algorithms. In the 1990s, significant progress was made in the development of software tools that facilitated researchers all over the world to continue research in the ASR field. In the year 1993, Steve Young and his team developed a system called the hidden Markov model toolkit (HTK) [30], which happened to be one of the most widely adopted toolkits for research in ASR domain. Later, in the year 2011, Daniel Povey and his team have developed another popular ASR toolkit referred to as Kaldi tookit [31]. This toolkit facilitated to incorporate various deep neural network (DNN) architectures into the structure of HMM. In the literature, this framework is referred to as the hybrid framework.

The recent advancements in compute power and the introduction of Python programming language [32] have provided a platform to the researchers for exploring new DNN architectures to train very large data sets. In 2014, Graves [33] proposed an end-to-end (E2E) framework for developing an ASR system that directly transcribes audio data with text, without requiring an intermediate phonetic representation. In 2017, Vincent Renkens developed a toolkit on TensorFlow [34] platform known as Nabu toolkit [35], that supports E2E ASR system training. Following that, Watanabe has developed another E2E speech processing toolkit referred to as Espnet toolkit [36] on PyTorch [37], a library designed to enable rapid research in machine learning. Since then, there has been growing research interest in E2E ASR as it simplifies the training process. Motivated by that, in this thesis, we explore the E2E framework for developing the code-switching ASR systems. For contrast purposes, the hybrid framework is also employed for developing the ASR systems.

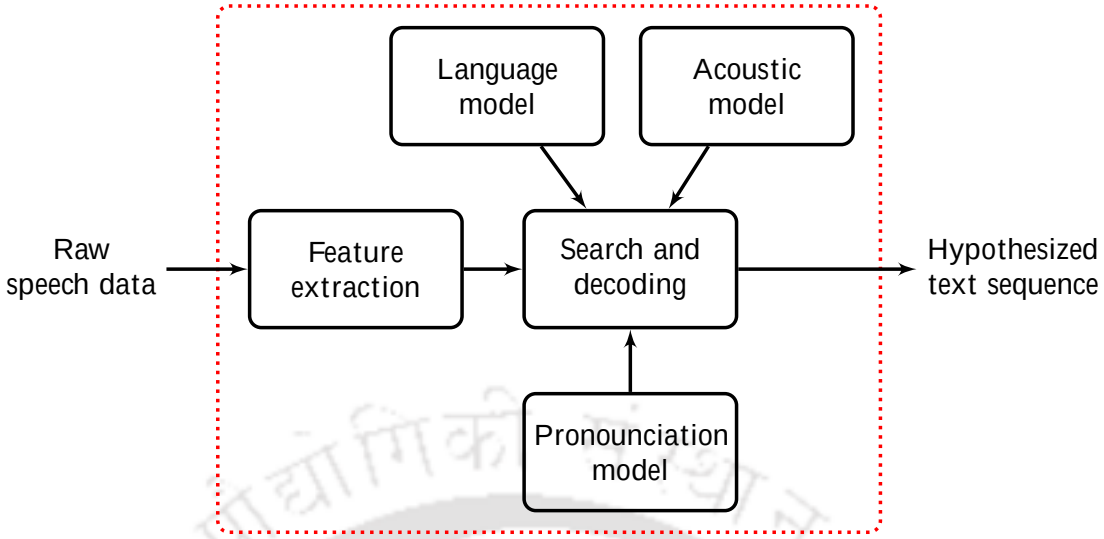


Figure 1.1: Block diagram representing the structure of a hybrid ASR framework. It is a modular architecture, where the language model, the acoustic model, and the pronunciation model are optimized independently with di erent objective functions.

1.2 ASR Paradigms

Over the years, the LVCSR systems have made great progress. Based on the key technologies employed, the ASR paradigms can be broadly divided into two categories. The salient attributes of those are discussed below.

1.2.1 HMM-based Framework

The HMM-based ASR systems employ the modular framework that comprises three sub-modules, namely, an acoustic model (AM), a pronunciation model (PM), and a language model (LM). A block diagram representing the structure of an HMM-based ASR system is shown in Figure 1.1. The input raw speech data is fist chopped into short-duration frames, AND then converted into a suitable parametric representation in the feature extraction module. The chosen parametric representation is intended to capture the relevant information in speech signal while removing the redundancies to achieve a compact representation. These short-time parametric representations are referred to as the acoustic feature vectors. The Mel-filterbank (FBANK) energy, the Mel-frequency cepstral coefficient (MFCC) [38], the linear prediction coefficient (LPC) [39], and perceptual LPC (PLPC) [40] based front-end parameterization of the speech signals are the most commonly used acoustic features for ASR task.

The AM involves the creation of statistical models for the sub-word units such as phonemes or senones, given the acoustic features. The parameters of AM are typically trained on those acoustic features. The ASR systems usually employ the HMM-based generative models for acoustic modeling. The observations for any particular acoustic unit are assumed to be generated by a finite state machine with state-specific probabilistic distributions. At each time unit, change in the state of the system occurs with a certain probability. An observation is generated with some probability distribution whenever a state is entered. To model the state observation probabilities, a multivariate Gaussian mixture model (GMM) is used. With the introduction of ANNs, there have been attempts made where the state-specific posterior probabilities are generated through the neural networks [25, 29]. Recently, the DNN containing many non-linear hidden layers and a very large output layer trained using the backpropagation algorithm [41] is being used for the LVCSR task. Thus the hybrid DNN-HMM [42, 43] systems are employed for AM training in ASR systems. Pronunciation variation as a term could be used to describe most of the variation present in speech. It is widely assumed that the pronunciation variation is one of the factors which degrades the performance of ASR systems. Towards addressing that issue, the HMM-based framework employs a PM, also known as pronunciation lexicon, which considers the typical pronunciation variations of each of the words present the task vocabulary [44].

In ASR, the LM helps reduce the search space of the sequence of words in the spoken utterances and provide the joint probability of the word sequences. The most commonly used n-gram LM predicts the next word in a sequence using the history of previous words [45]. But, the n-gram LM can not model the long-term dependencies. On the other hand, the recurrent neural network (RNN) based LM (RNNLM), with the presence of the feedback connections, can model both the context and the long-term temporal information efficiently. In the current literature, the RNNLM is extensively explored and is reported to significantly outperform the conventional n-gram LM [46–48]. For the large vocabulary ASR task, the search space increases exponentially with the increase in the vocabulary, and evaluating all possible word sequences is infeasible. For a simple exhaustive search, with a vocabulary size V, and a word sequence of length K, there are V^K different possible word sequences to be considered. Therefore, the HMM-based ASR systems employ the Viterbi decoding algorithm to find the optimal path through a probabilistically scored time/state lattice.

Figure 1.2: Block diagram representing the structure of an E2E ASR framework. All the modules in this framework are jointly optimized with a common objective function.

Though the HMM-based framework is widely used for developing the LVCSR systems, it suffers from a few drawbacks, as listed below.

All the sub-modules mentioned above are trained and optimized independently with different objective functions. Hence, the resulting system can be sub-optimal.

It requires the creation of the phonetically labeled data and a PM, which is time-consuming and requires expert knowledge.

It assumes conditional independence within the HMM and between different modules, to simplify the models construction and training. But, this assumption is not valid for LVCSR.

1.2.2 End-to-End Framework

To address the issues mentioned above in the HMM-based framework, recently, the end-to-end (E2E) framework was proposed and successfully explored in the ASR task [33, 49–51]. Unlike the HMM-based framework, the E2E framework has the following characteristics, which makes it more suitable for LVCSR tasks.

In the E2E framework, all the modules are merged into a single network and trained jointly. The joint modeling enables the network to use a cost function that can be optimized over the final evaluation criteria, thereby resulting in global optimization.

The network is trained with characters as the output targets, given the acoustic features as input. Thus, the E2E ASR framework does not require the phonetically labeled training data and does not include any explicit PM.

A typical E2E ASR system includes the following modules: (i) an encoder, to transform the given input speech sequence into feature sequence, (ii) an aligner, to realize the alignment between

the transformed feature sequence and the language, and (iii) a decoder, to decode the output target sequence. A block diagram representing the structure of a typical E2E ASR system is shown in Figure 1.2. Note that all those modules may not always exist in the E2E framework. The E2E framework itself is a complete structure, and hence it would be difficult to identify which module performs which sub-task. According to [52], depending upon the implementation of soft alignment, the E2E framework can be classified into three types, namely, (i) connectionist temporal classification (CTC) based framework [53], (ii) RNN transducer-based framework [54], and (iii) attention-based framework [33]. In the CTC-based framework, all possible hard alignments are identified, and then soft alignment is performed by aggregating those hard alignments. The CTC decoding assumes that the output labels are independent of each other. The RNN transducer-based framework follows the identical approach as that of the CTC-based framework for identifying the soft alignment except that the output labels are not assumed to be independent. The attention-based framework no longer finds all possible hard alignments. Instead, it employs the sequence-to-sequence modeling with an attention mechanism to directly perform the soft alignment. In this thesis, we explore only the CTC- and attention-based frameworks for developing the code-switching ASR systems. A more detailed discussion on those variants is provided in Chapter 4.

1.3 Motivation for Research Work

Early works on code-switching ASR [55–57] happen to employ the hybrid framework typically developed for monolingual ASR task. As discussed earlier, the hybrid ASR framework has a few shortcomings which can be efficiently addressed by the E2E ASR framework. In the code-switching case, usually two or more languages are involved. As a result of that, the E2E framework becomes more attractive for developing such ASR systems. Motivated by that, recently, the E2E framework has been explored for the code-switching ASR task [58–61]. A detailed literature review of the code-switching ASR works is presented in Chapter 2. In the following, we list a few research challenges posed by the code-switching phenomenon in developing the E2E ASR system, which forms the motivation for this thesis work.

In the existing code-switching E2E ASR works, the target set is derived by simply combining the character sets of the languages involved. It is argued that such systems would suffer

from high confusability among the cross-language targets unless a sufficiently large amount of data is available for training. The possible cause of the confusability lies in the broad acoustic similarity among sound units involved in most of the code-switching language pairs. Also, for the enhanced target set, such systems would exhibit high computational complexity. We hypothesize that one can avoid such a confusability if a common phone set covering the underlying languages in code-switching data is used as the output target, instead of the combined target set.

On the other hand, in the code-switching case, the count of homophones increases significantly due to the presence of two or more languages. Homophones are defined as those words which have identical pronunciation but different spellings. With the use of a common phone set, the problem of homophone confusability gets further enhanced during target-to-word (T2W) transduction for computing the word error rate (WER). For addressing this issue, an efficient T2W transduction scheme that can incorporate context information is required.

Further, we note that the code-switching phenomenon cannot be characterized as a random mixing of words or phrases from two or more languages [62]. The bilingual code-switching phenomena have been noted to follow some broad semantic and syntactic rules [63, 64]. The existing LMs can not capture such context information present in the code-switching sentences. We hypothesize that the context information of the non-native language words can be enhanced if we somehow capture those rules while training the LM. Such an LM, when incorporated into the T2W transduction scheme, is expected to improve the ASR system performance in terms of WER, for code-switching data.

Additionally, unlike the monolingual case, the research in the code-switching domain is somewhat limited. The main reason for the same is the lack of availability of domain-specific resources. Hence, there is a greater need to develop such resources for promoting research in the code-switching domain.

1.4 Contributions of the Work

Motivated by the issues mentioned above, in this thesis, we aim to develop a robust code-switching E2E ASR system that can handle those challenges, under low-resourced conditions. The salient contributions of the thesis are listed below.

Towards addressing the data scarcity issue, a Hindi-English code-switching database referred to as the *HingCoS* corpus is developed in the Indian context. This corpus consists of code-switching text data having 26k sentences with a total of 0.58 million words. The corpus also contains 25 hours of matching speech data corresponding to 9.2k code-switching sentences covering a vocabulary of 6.5k words. Also, a lexicon covering most of the pronunciation variations for each of the words present in the speech data is created by defining a common phone set. The baseline ASR systems have been developed on the said corpus by employing the hybrid framework.

An E2E framework has been explored for developing an ASR system for Hindi-English code-switching data. Further, a reduced target set is defined for training the Hindi-English code-switching E2E ASR system by exploiting the acoustic similarity. The reduced target set avoids confusability among the cross-lingual targets. Additionally, the reduced target set-based E2E ASR system training takes much less memory and computational time than the existing combined target set case.

For transducing character/phoneme sequences outputted by the E2E ASR system into desired word sequences, a novel context-dependent target-to-word (T2W) transduction scheme is proposed. This scheme employs an explicit error model (EM) along with an LM to provide context information during transduction. The proposed transduction scheme has shown significant improvement in terms of word error rate (WER) compared to the naive approach.

To enhance the LM performance in T2W transduction, a novel textual feature referred to as the code-switching location (CSL) feature is proposed. This feature allows the LM to predict the possible code-switching instances. In addition to that, an improved parts-of-speech (POS) labeling scheme for accurate tagging of non-native words embedded in the code-switching

data has also been proposed. To incorporate these textual features into language modeling, the factored LM (FLM) is employed. Both the proposed features are evaluated by directly incorporating them into code-switching data. Alternately, those textual features have also been assessed by adapting an existing native monolingual LM to handle the code-switching data. The FLMs trained by incorporating the proposed features have shown to outperform the existing LMs in terms of perplexity. When those FLMs are incorporated into the proposed T2W transduction scheme, further improvement in WER is achieved.

It is worth highlighting that, though the proposed approaches are evaluated for the Hindi-English code-switching case, they are generic enough to be applied for any other valid code-switching language pairs across the globe.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. A brief literature review on the code-switching phenomenon and the emergence of Hindi-English code-switching in India is presented in Chapter 2. Also, the details of existing code-switching corpora and state-of-the-art techniques employed for training various systems for code-switching data are described in detail. A few of those techniques are used to contrast the efficacy of the proposed methods pursued in this thesis.

In Chapter 3, the collection of a Hindi-English code-switching text as well as speech database referred to as the HingCoS corpus, and the development of the corresponding lexical resources is described. It elaborates on sources and the protocol used for collecting the corpus, along with the statistical analyses of both the text and speech corpora. It is followed by comparative analyses of both the HingCoS text and speech corpora with the existing code-switching corpora in the literature. The details of the baseline ASR system development on the collected corpus and the obtained experimental results are also reported.

Chapter 4 explores the E2E framework for developing the ASR systems for code-switching data. It presents a detailed description of the CTC- and the attention-based E2E ASR frameworks. Following that, the creation of a reduced target set covering the underlying languages in code-switching data based on the acoustic similarity is presented. In the context of low resourced modeling, this reduced target set avoids the confusability among cross-language targets. Later, the development of

E2E ASR systems on both the existing combined and the proposed reduced target sets is discussed along with the challenges arising in each case, respectively. The proposed context-dependent T2W transduction scheme is also discussed and then evaluated by providing various degrees of context information. The proposed approaches are evaluated on both the CTC- and attention-based E2E ASR frameworks.

A brief discussion on the factored LMs is presented in Chapter 5. Following that, the improved POS tagging scheme and the proposed CSL feature for Hindi-English code-switching text data are discussed in detail. The evaluation of the proposed textual features in contrast with the existing features has been done for two conditions: (i) by directly incorporating the textual features into code-switching data while training the LM, and (ii) by including the textual features into monolingual data and adapt the monolingual LM trained on that data to handle the code-switching. The proposed textual features are also revalidated on Mandarin-English code-switching text data, and the evaluation results are provided. Further, the FLMs trained on the proposed features are incorporated into the proposed T2W transduction scheme and evaluated for the ASR task.

Finally, the thesis is summarized, and the possible future directions of the work are discussed in Chapter 6.

2

Scope and Literature Review of Code-Switching

Contents

2.1 Hindi-English Code-Switching 14
2.2 Existing Works on Code-Switching 16
2.3 Existing Code-Switching Databases 18
2.4 Conclusions 21

In multilingual communities, the speakers often switch or mix between two or more languages or language varieties during communication. In literature, this phenomenon is referred to as code-switching [6,7,64]. The language to which the syntax of a code-switching sentence belongs is referred to as a native language, while that of the embedded foreign words is referred to as a non-native language [8]. Over the decades, due to colonization and other historical factors, many people have migrated from one linguistic region to another for better trade opportunities and livelihood. In such situations, communicating in two or more languages helps people to interact better. It has been observed that, over time, such mixed linguistic communities tend to code-switch to mingle well culturally. In turn, that led to the emergence of code-switching across the globe. In literature [65–70], a few other possible reasons for code-switching are attributed to the lack of appropriate words in the native language, emphasizing specific word/phrase, and showing expertise.

The salient examples of the multilingual communities across the world are as follows. Spanish-English [71] in the United States of America, Arabic-English [72] in Egypt, French-German [73] in Switzerland, Frisian-Dutch [74] in Netherlands, Malay-English [56] and Mandarin-English [75] in Malaysia, Mandarin-Taiwanese [55, 76] in Taiwan, Cantonese-English [77] in Hong Kong, English-isiXhosa, English-isiZulu, and English-Setswana [78] in South-Africa, and Hindi-English [9,79,80] in India. In this work, we study the Hindi-English code-switching phenomenon in the Indian context.

2.1 Hindi-English Code-Switching

India is the second-most populous country in the world and has 23 official languages, including English. After gaining independence from British rule, though the Indian constitution declared Hindi as the primary official language, English usage was continued as a secondary language for its dominance in administration, education, and law [79, 80]. Thereby, communicating with frequent English words/phrases was considered a sign of not only being in power/educated but also more trendy. Over the years, substantial code-switching to English while speaking the native Indian languages has become a common trend across India, particularly by the urban population [81, 82]. Figure 2.1shows the distributions of language families in India according to census 1991. From that Figure, it can be noted that 40.2% of the population are native Hindi speakers, followed by Bengali (8.3%) and Telugu (7.9%). Therefore, one can find frequent code-switching between the Indian

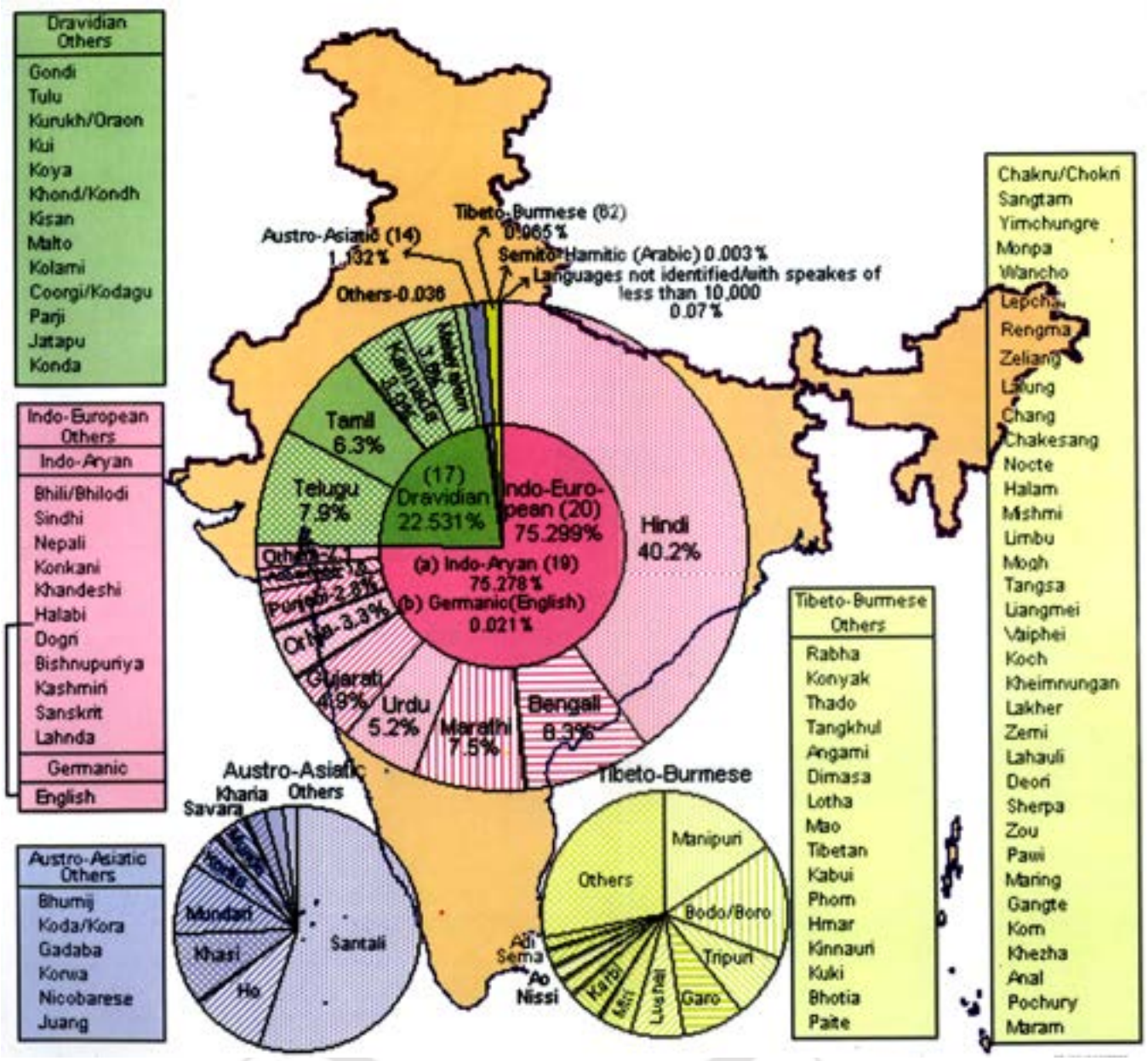


Figure 2.1: Distributions of Indian language families as per the 1991 census. The gures in bracket refer to the number of languages (inclusive of mother tongues grouped under them) in each language family. It can be noted that 40:2% of the population are native Hindi speakers. The image is adapted from http://www.ciil-lisindia.net/. Copyright 2011 c Central Institute of Indian Languages.

languages and English, with Hindi-English being the most dominant one. During British colonial rule, many words from English were borrowed in Hindi. The majority of them were proper nouns, including a few collective/abstract nouns. Over the period, they got internalized in Hindi for the ease of reference or the lack of acceptable equivalent Hindi words. In this study, except for the proper nouns borrowed from English, all remaining embedded English words are assumed to be code-switched.

Table 2.1: Example Hindi-English sentences along with their English translated versions showing the inter-sentential code-switching and the variants of the intra-sentential code-switching. Type-1 and Type-2 variants of intra-sentential code-switching refer to high and low contextual information being carried by the non-native (English) words, respectively.

Inter-sentential code-switching	Example-1	she is the daughter of our ceo, वह यहाँ दो दिन के लिए आई है she is the daughter of our ceo, she has come here for two days	
	Example-2	मुझे अमेरीका में रहते चार साल हो गए, but I still miss my country I have been living in america for four years, but I still miss my country	
Intra-sentential code-switching	Type-1	Example-1	कृपया मुझे मेरा current account balance बताएं please tell me my current account balance
		Example-2	देश की currency every year change होनी चाहिए the country's currency should change every year
	Type-2	Example-1	class और object के बीच relationship क्या है what is the relationship between class and object
		Example-2	meeting का outcome क्या था what was the outcome of meeting

Table 2.1 shows a few example Hindi-English and their translated English sentences with different modes of code-switching while highlighting the differences in the contextual information carried by the non-native words. In Type-1 intra-sentential code-switching, the non-native language words either occur in sequence or form a phrase, thus carry some contextual information. Whereas, in the Type-2 case, the non-native language words are embedded into the native language sentences in such a manner that virtually no contextual information could be derived from those words alone.

2.2 Existing Works on Code-Switching

Current literature on code-switching can be grouped into three broad research areas: (i) linguistics, (ii) natural language processing (NLP), and (iii) speech processing. The researchers in linguistics have studied the impact of code-switching from the point of view of socio-linguistics [65,83] and syntactics [70,84]. In [65], the researchers highlighted the socio-psychological and linguistic factors behind the code-switching phenomenon. The authors in [83] examined the grammatical structure and specific syntactic boundaries of a language that occurs due to code-switching. In [70], based on the locations of the non-native words, code-switching was broadly classified into two modes. When the switching happens within the sentence, it is referred to as the *intra-sentential* code-switching, and the one predominantly happening at the sentence boundary is referred to as the *inter-sentential*

code-switching [84]. Intra-sentential mode of switching is a common phenomenon and has become an identifying characteristic in bilingual communities [85].

In the NLP domain, the researchers explored the code-switching phenomenon for language modeling [72, 77, 86] and machine translation [87] tasks. In the recent past, there have been a few attempts to train an LM for code-switching data [56, 77, 86, 88–90]. In the absence of a sufficient amount of code-switching text data, in [77], the authors applied translation-based and semantics-based mappings to efficiently estimate the probability of low-frequency and unseen mixed-language N-gram events. In [86], the authors explored an interpolated LM where the vocabulary of both the languages was merged while estimating the LM probabilities for each language separately. The word entries of one language are included in the LM of the other language with zero counts, and those are assigned a small back-off probability. In this way, all the words have non-zero probabilities, which enabled them to handle the intra-sentential code-switching. In contrast to interpolating two monolingual LMs, bilingual LM is also proposed. For training bilingual LM, in [56], the authors merged the training texts with the words which have the same spelling in more than one of the considered languages added with respective language tags. Thus, the counts of such similar words are distinguished based on the language tag, and an LM was trained on the merged training text corpus. For efficient language modeling of code-switching data, in [88, 91], the authors proposed a few semantic features such as POS and language identification (LID) features, which can help to predict the code-switching instances in the sentence. For this task, the factored language model (FLM) is applied. A detailed discussion on the FLM is provided in Chapter 5.

In the speech processing area, there have been exciting research challenges in the code-switching domain, which include acoustic modeling [55, 56, 92], language identification and diarization [76, 93], and speech synthesis [94]. For ASR of code-switching data, there are two major distinct frameworks in literature: (i) multi-pass, and (ii) one-pass. In a multi-pass framework, the exact instance of language switching in the utterance is determined, and the language of that instance is identified. Later, the corresponding language-dependent ASR system is used to recognize the speech segment. Whereas, in the one-pass approach, an ASR system is developed by encompassing both the languages in the code-switching data. In [55], an integrated one-pass approach where the speakers switch back and forth between the two languages, was proposed for ASR of code-switching data. This

framework is based on a three-layered recognition scheme consisting of a mixed-language HMM-based acoustic model, a knowledge-based and data-driven probabilistic pronunciation model, and a tree-structured searching net. In the same work, the traditional multi-pass recognizer is also implemented by following a similar procedure presented in [76, 95]. The authors in [56] proposed a one-pass framework for acoustic model adaptation were the models of the involved languages are interpolated and merged to cross adapt and then recognize the code-switching speech data. In another work [92], a one-pass framework with a modified pronunciation dictionary is proposed. The modified pronunciation dictionary enables us to avoid building an acoustic model for the mixed languages and further facilitated to perform ASR on the models trained on one of the involved languages in code-switching.

The recent works have explored the E2E framework in the code-switching ASR domain. In the very first work [58], Seki *et al.* explored an E2E ASR system for code-switching tasks on an artificially created dataset obtained by concatenating the monolingual utterances. In contrast, Shan *et al.* [60] employed a real Mandarin-English code-switching dataset for developing the attention-based E2E ASR system. For improving the ASR performance, the multi-task learning (MTL) framework involving the LID [59, 96] was employed. In another work, Li *et al.* [97] explored a CTC-based E2E ASR system combined with frame-level LID for recognizing Chinese-English code-switching data. In a recent work on Mandarin-English code-switching ASR, Zeng *et al.* [61] experimented with data augmentation, MTL for LID, byte-pair encoding, and expansion of vocabulary in LM for N-best rescoring in the context of attention-based E2E framework.

It is to note that the scope of this thesis is limited only to the acoustic and language modeling aspects, in the context of code-switching ASR task.

2.3 Existing Code-Switching Databases

Though the phenomenon of code-switching is widespread across the world, the effective handling of code-switching in the above-said applications is still quite challenging compared to that of the monolingual case. The primary reason for the same is the availability of domain-specific resources is minimal. A few of the existing resources are listed in the following. The CUMIX [77], a Cantonese-English code-switching speech corpus, was developed at the Chinese University of Hong Kong.

This database contains 17 hours of speech data read by 40 speakers. The purpose of this corpus is to develop Cantonese-English code-switching ASR system. Lyu et al. [98] created a training dataset consisting of monolingual Taiwanese and Mandarin speech data from 100 speakers, with each speaker uttering 700 utterances in both the languages. For evaluation purposes, a small Mandarin-Taiwanese code-switching test set containing 4000 utterances recorded from 16 speakers was also developed. The English-Spanish code-switching speech corpus was compiled by Franco and Solorio at the University of Texas [99]. This corpus contains 40 minutes of transcribed spontaneous conversations of 3 speakers. The SEAME corpus [75, 100], a Mandarin-English code-switching conversational speech corpus, is developed at Nanyang Technological University, Singapore, and Universiti Sains Malaysia. This database contains 63 hours of spontaneous Mandarin-English code-switching interview and conversational speech uttered by 157 Singaporean and Malaysian speakers. The CECOS [101], a Chinese-English code-switching speech corpus containing 12.1 hours of speech data collected from 77 speakers uttering prompted code-switching sentences is developed at the National Cheng Kung University in Taiwan. A corpus of a Sepedi-English code-switching speech corpus was created by the South African CSIR [102]. This database consists of 10 hours of prompted speech, sourced from radio broadcasts, and read by 20 Sepedi speakers. FAME! [74], a Frisian-Dutch code-switching speech corpus of radio broadcast speech, is developed at Radboud University, Nijmegen. The recordings are collected from the archives of Omrop Fryslan, the regional public broadcaster of the province Fryslan. The database covers almost a 50 years time span. The Malay-English corpus consists of 100 hours of Malaysian Malay-English code-switching speech data from 120 Chinese, 72 Malay, and 16 Indian speakers [56]. MediaParl is a Swiss accented bilingual database containing both French and German recordings as they are spoken in Switzerland. The data was recorded at the Valais Parliament. Valais is a bi-lingual Swiss canton with many local accents and dialects [73]. The FACST, a French-Arabic speech corpus, consists of records of code-switching read and conversational utterances by 20 bilingual speakers who tend to code-switch in their daily lives [103]. It contains about 7.30 hours of data. Westhuizen et al. created a South African speech corpus containing English-isiZulu, English-isiXhosa, English-Setswana, and English-Sesotho code-switching speech utterances from South African soap operas. The soap opera speech is typically fast, spontaneous, and may express emotion, with a speech rate higher than prompted speech in the same

languages [78]. Injy Hamed et al. recently developed an Arabic-English code-switching corpus by conducting the interviews with 12 participants [104]. A small Hindi-English code-switching speech corpus was collected at the Hong Kong University of Science and Technology [69]. In this corpus, the speech data correspond to interviews of student volunteers, which is about 30 minutes of data collected from 9 speakers. Recently, a phonetically balanced code-mixed (PBCM) Hindi-English speech corpus was developed by the speech and vision lab, International Institute of Information and Technology, Hyderabad [105]. This corpus consists of 78 speakers, with each speaker having recorded around one minute of speech in read style. Also, the Microsoft Research (MSR) group has developed a Hindi-English code-switching speech corpus consisting of 50 hours of conversational speech spoken by around 500 speakers [106]. The salient details of all the above-discussed code-switching speech corpora are briefly summarized in Table 3.7 of Chapter 3.

Motivated by the efforts done elsewhere, as a part of this thesis, we endeavored to create an open resource for research in ASR of code-switching speech in the Indian context. A monolingual ASR system may be capable of recognizing a few words from a foreign language but cannot handle a significant amount of code-switching in the data. In the recent past, researchers have reported that the native language of the speaker influences the foreign (non-native) language acquisition [107]. In India, English is taught from the elementary level in the schools across the country. Despite that, the English pronunciations of the majority of the pupils carry significant native language influences. On account of the existence of variants of English pronunciations and code-switching effects, the development of an ASR system for Hindi-English code-switching speech data is a challenging task. To the best of our knowledge, there is no large-sized Hindi-English corpus publicly available. Towards addressing that constraint, we have created a Hindi-English code-switching text corpus. Along with that, a Hindi-English speech corpus is also created that covers all typical sources of variations such as accent, session, channel, age, gender, and the influence of the mother tongue. The sentences spoken in the speech corpus are a subset of the text corpus. The details of the created Hindi-English text and speech corpora along with the sources and the protocol used for the creation of the corpus are discussed in Chapter 3.

2.4 Conclusions

In this chapter, a brief discussion on the code-switching phenomenon and the emergence of Hindi-English code-switching in India is provided. The state-of-the-art techniques employed for language modeling and speech recognition of code-switching data are explained in detail. Some of the discussed techniques are employed in the later chapters to provide the contrast to different systems explored in this thesis. Following that, the details of the existing code-switching corpora are also presented.

With the given literature review, we note that the phenomenon of code-switching is widespread across the world. Hence, handling of such phenomenon by the existing speech-based applications is very much necessary for better HMI. However, the research in the code-switching domain is some what limited due to lack of availability of domain specific resources. Towards addressing the data scarcity issue and promoting research in code-switching domain, as a part of this thesis, we have created a Hindi-English code-switching corpus. The details of the same are presented in the following chapter.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

3

Hindi-English Code-Switching (HingCoS) Corpus

Contents

3.1 HingCoS Text Corpus 25
3.2 HingCoS Speech Corpus 27
3.3 Lexical Resources 30
3.4 Statistical Analysis of the HingCoS Corpus 32
3.5 Comparison with Existing Code-switching Corpora 37
3.6 Experimental Setups and Evaluations 39
3.7 Conclusions 47

It is highlighted earlier that the phenomenon of code-switching is widespread across the world. Though the research activity in this domain is somewhat limited due to lack of availability of the domain-specific resources. Towards addressing that data scarcity issue, a Hindi-English code-switching corpus was created as a part of this research at the Indian Institute of Technology Guwahati (IITG). The created corpus contains the code-switching text as well as the speech data and is formally referred to as the *IITG-HingCoS* corpus. But for brevity, we refer to that as the *HingCoS* (हिंगकोष) corpus in the remainder of this thesis. This corpus has been made public [1] to augment the limited available resources as well as to promote more research activity in Hindi-English code-switching domain.

Like any monolingual speech corpus, the code-switching speech data can be collected in any of the three modes: (i) read speech, (ii) conversational/interview speech, and (iii) radio/television broadcast speech. However, the researchers have pointed out that the code-switching takes place both in spoken and written formats during spontaneous interactions [9, 10]. It would have been preferable to collect the data in conversational mode. For the ease of creation, the code-switching speech data has been collected in the read speech mode. But, the speakers were asked to read the text corresponding to spontaneous interactions. For the same, we first created the Hindi-English code-switching text corpus. The collection of code-switching text data is not easy since the traditional sources like newspaper/broadcaster websites are monolingual. Therefore, we had to access the blogging websites and other social media portals to get the relevant code-switching text data. For developing the ASR system, apart from the speech and text data, the lexical resources like pronunciation dictionary are also needed. The novel contribution in this domain lies in the creation of a common phone set covering the Hindi-English code-switching, which is used to create the pronunciation dictionary.

In the following, we describe the procedure followed for the collection of HingCoS text and speech data along with the development of the lexical resources using the created common phone set. We have also provided a brief survey of existing code-switching corpora and contrasted them with the created HingCoS corpus. Following that, the evaluations of the created HingCoS corpus in both language modeling and speech recognition tasks are presented.

[1] `https://www.iitg.ac.in/eee/emstlab/HingCoS_Database/HingCoS.html`

3.1 HingCoS Text Corpus

At first, the text data has been collected by crawling a few Indian blogging websites that cover different contexts and follow conversational style with significant code-switching. The details of those websites and their context is discussed below.

ShoutMeHindi [2]: It contains information about how to start a blogging website and how to earn money from it. It also explains the social media websites (Facebook, Twitter, YouTube, etc.) and gives some tips and tricks to use them.

*Computing Notes in Hinglis*h [3]: This blog explains about object-oriented database management system in detail along with its design methodology and behavioral concepts.

Techyukti [4] and *HindiMe* [5]: These websites discuss about more than 100 varieties of recent advancements in technology. The salient topics include mobiles, cameras, PUBG, WhatsApp, Paytm, online voter ID enrollment process, adhaar card application process, search engine optimization, etc.

LearnCpp [6]: It is a free website having tutorials about *C++* programming language that includes the steps to write, compile, and debug the programs written in *C++* along with plenty of examples.

In all the above blogs, the bloggers have tried to explain the chosen context by writing in Hindi with frequent code-switching to English words/phrases. Also, the bloggers have followed their individualistic and rather casual writing styles. At times, even some Hindi and English words are written in cross scripts. As a result of that, the collected data exhibits a lot of variabilities, and its removal is not only essential but also a bit challenging. Therefore, the crawled data is first normalized before converting it into meaningful sentences, and the steps followed for the text normalization is described below.

[2] https://shoutmehindi.com
[3] https://notesinhinglish.blogspot.in
[4] https://www.techyukti.com
[5] https://hindime.net
[6] http://www.learncpp.com

Table 3.1: Quantifying the impact of text normalization of the raw text data as described in Section 3.1.1. The non-characters include punctuation marks, braces, numerals, mathematical symbols, emoticons, etc.

Attribute	Raw	Normalized
Number of unique non-characters	1,657	Nil
Number of unique Hindi words	10,737	6,029
Number of unique English words	24,498	8,614
Vocabulary size	36,892	14,643
Number of sentences	12,284	25,988

3.1.1 Steps Involved in Text Normalization:

Punctuation marks, special characters, bullet marks, emoticons, etc., are filtered out.

The braced explanations, website links, directory paths, etc., are filtered out while retaining the key information.

All numerals, mathematical, and currency symbols are replaced by their spellings either in Hindi or English based on the context.

All the uppercase English characters in the data are converted to lowercase characters.

Except for the proper nouns, all English and Hindi words written in cross scripts are fixed to have the correct scripts, i.e., the Hindi words are written in the Hindi alphabets (Devanagari script), and the English words are written in English alphabets (Latin script). Also, any error spotted in the spellings of Hindi and English words is fixed.

All shorthand words are converted to their respective full forms, while all standard abbreviations are left as they are.

Finally, the sentence begin, and end markers are inserted to parse the data into meaningful sentences while removing any erroneous repetitions of words and phrases if spotted.

Table 3.1 quantifies the impact of the above-mentioned text normalization process. A few examples illustrating the text normalizations are shown in Figure 3.1. Further, the validation of the text normalization in the language modeling domain is reported in Table 3.2. A detailed analysis of the HingCoS text corpus is presented later in Section 3.4.

Raw text:
नमस्कार, मैं गुरमीत, ShoutMeHindi का Senior Editor हूँ. आप सभी के सहयोग से हमारा यह blog, हिन्दी भाषा में ब्लॉग्गिंग और online पैसे कमाने के सम्बंधित जानकारी उपलब्ध करवाने वाला एक popular blog बन चूका है. इसी तरह तरह अपना सहयोग देते रहिये और हम आपके लिए नईं-नईं information उपलब्ध करवाते रहेंगे. :)
सबसे पहले आपको www.youtube.com वेबसाइट को open करे और Right side उपर दिए गए Circle option पर जाकर Creator Studio click करे.
Facebook ads को use करने के कुछ कारण, मैंने नीचे mention किये हैं:

- हर महीने, 2 billion (200 करोड़) लोग Facebook को use करते हैं. यह audience का एक बहुत ही ज्यादा बड़ा base है.
- America में, यदि लोग 5 minutes अपने फ़ोन पर बिताते हैं, तो उन 5 minutes में से 1 minute Facebook use करते हैं.

Gmail, Google+ की तरह Google का Product है|
Hi Frnds, Kya apko YouTube Video Editor ke bare me janakri hai?

Normalised text:
<s> नमस्कार मैं गुरमीत shoutmehindi का senior editor हूँ </s> <s> आप सभी के सहयोग से हमारा यह blog हिन्दी भाषा में blogging और online पैसे कमाने के सम्बंधित जानकारी उपलब्ध करवाने वाला एक popular blog बन चुका है </s> <s> इसी तरह अपना सहयोग देते रहिये और हम आपके लिए नईं नईं information उपलब्ध करवाते रहेंगे </s> <s> सबसे पहले आपको youtube website को open करे और right side उपर दिए गए circle option पर जाकर creator studio click करे </s> <s> facebook ads को use करने के कुछ कारण मैंने नीचे mention किये हैं </s> <s> हर महीने दो billion लोग facebook को use करते हैं </s> <s> यह audience का एक बहुत ही ज्यादा बड़ा base है </s> <s> america में यदि लोग five minutes अपने फ़ोन पर बिताते हैं तो उन five minutes में से one minute facebook use करते हैं </s> <s> gmail google plus की तरह google का product है </s> <s> hi friends क्या आपको youtube video editor के बारे में जानकरी है </s>

Figure 3.1: Illustration of salient text normalizations described in Section 3.1.1 that have been applied to raw Hindi-English data collected from the web sources. Note that, <s> and </s> represents the sentence begin and end markers, respectively.

Table 3.2: Validation of the text normalization process. In this experiment, two language models (LMs) are created using the raw and the normalized versions of the training set. For performance evaluation, a development and a test datasets which are non-overlapping to the training dataset, are also created. The corresponding perplexities of the LMs trained on raw- and normalized-text are reported. Note that, the out of vocabulary rate of the development and the test datasets with respect to the normalized training dataset turns out to be 1.69% and 2.08%, respectively.

Training dataset	Evaluation dataset	Perplexity
Raw	Development	351.02
	Test	359.61
Normalized	Development	174.93
	Test	176.43

3.2 HingCoS Speech Corpus

For creating the HingCoS speech corpus, about 30% of the sentences available in the HingCoS text corpus were randomly selected for recording the speech data from native Indian speakers. The selection of the sentences has been done in such a way that they cover the majority of the contexts described in Section 3.1. The speech data is collected over a toll-free telephone-based voice-server available in the Electro-Medical and Speech Technology (EMST) Laboratory at IITG. A group of

students and residents of IITG participated in the speech data collection. Each of the volunteers was given 100 unique sentences for recording the speech data by calling to the voice-server. Those 100 sentences were further partitioned into 5 groups of 20 sentences each. The volunteers were asked to record each group of sentences in a separate session. We also recorded the meta-data of volunteers comprising the name, age, gender, mother tongue, and native state. The call flow of the voice-server used for recording the HingCoS speech corpus is shown in Figure 3.2.

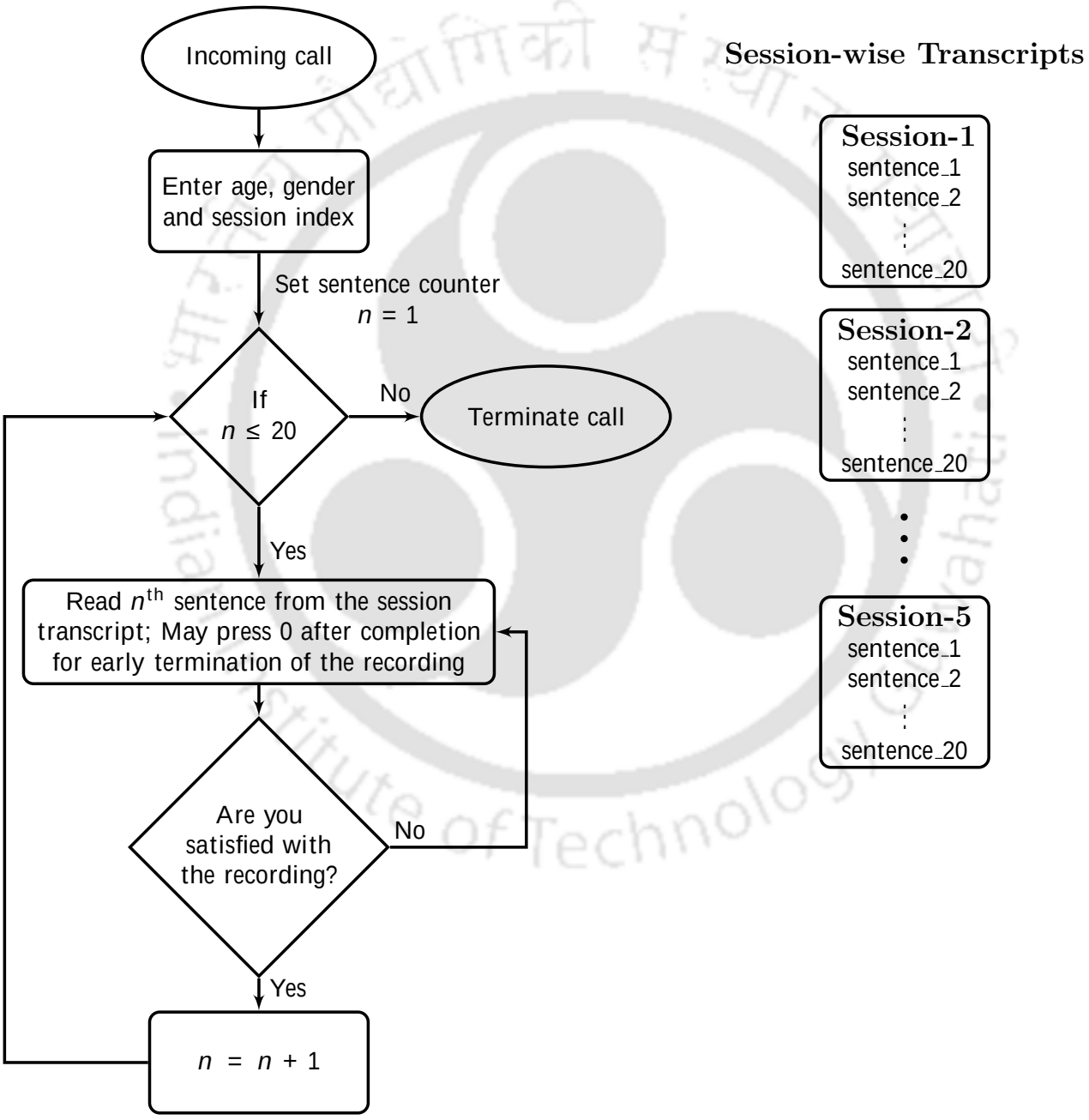


Figure 3.2: The call flow of the voice-server used by the volunteers for the session-wise recording of Hindi-English transcripts. Each of the volunteers was given 100 sentences and was asked to record them in 5 different sessions with each session have 20 sentences.

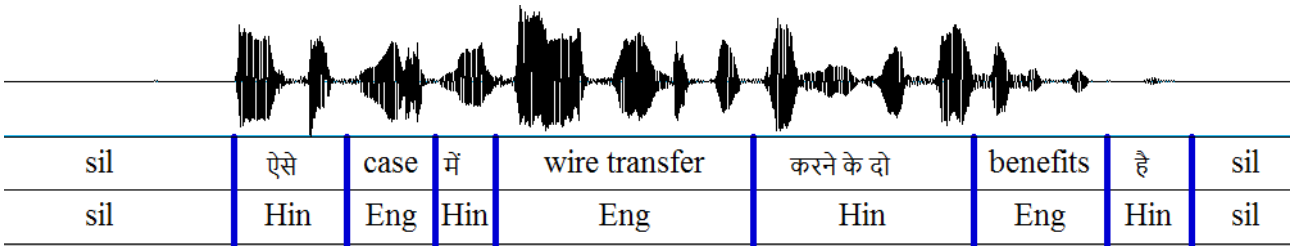


Figure 3.3: An example of the typical recorded Hindi-English speech utterance marked with native and non-native word/phrase boundaries along with their corresponding language identity labels. The short-hand notations `Hin' and `Eng' are used to denote Hindi and English words/phrases, respectively.

All the text data collected from the blogs is written in a style as if an expert is educating/explaining his/her audience about a topic. These sentences are read by the volunteers while recording the speech data. Therefore we referred to it as "read" speech in this work. In order to ensure the quality and integrity of the collected speech data, each of the volunteers was given the following instructions:

Read the given sentences at least two times before recording the speech data in order to sound them in a spontaneous manner.

Prior to the actual recording, familiarize with the call-flow of the voice-server by conducting a few dummy recordings.

Record each group of sentences in a separate session while keeping a gap of at least one day between the sessions and choosing different environments as far as possible.

Enter the correct personal details as well as the session index. Carefully read the prompted sentence from the text transcript provided for that session.

Owing to the use of voice-server, the recorded speech data has the sampling rate of 8 kHz and the precision of 16-bits. Speech files corresponding to the read sentences are stored in *.wav* format and labeled as <speaker ID>_<gender>_<age>_<session index>_<sentence number> *.wav* (For example, if a 29 year old male speaker having unique speaker ID as Spk10, accessed the voice-server and read the 3^{rd} sentence from Session-1 transcript, then that particular speech file is labeled as Spk10_M_29_1_03.*wav*). Figure 3.3 shows a typical recorded Hindi-English speech utterance marked with native and non-native word/phrase boundaries along with their corresponding language identity labels.

Though the database collection protocol ensures a quality recording condition, a few challenges still exist due to technical issues, such as the creation of empty/broken recordings due to call drop or power failure. Therefore, a lot of manual hours are invested in inspecting and pruning out any empty and broken recordings. We employed a voice activity detection (VAD) for removing the long silences and any non-overlapping background noises present in the recorded data. This VAD included background noise suppression and was developed in an earlier work [108]. A detailed analysis of the HingCoS speech corpus is presented in Section 3.4.

3.3 Lexical Resources

It is well known that the lexicon plays an important role in the ASR task. It establishes the link between the acoustic representation of basic sound units and the symbols outputted by the ASR system. The design of a lexicon involves two steps: (i) fixing of the vocabulary covering the task, and (ii) the listing of all possible pronunciation variants of each word in the vocabulary. Unlike the monolingual task, the lexicon in the code-switching task has to cover the words from two or more languages involved.

In this section, we discuss the development of a lexicon for the Hindi-English ASR system. Firstly, for compact acoustic modeling, we intended to define a phone set that covers all basic sounds present in both Hindi and Indian-English. To achieve that, a composite phone set is used, which has been proposed recently in the context of computer processing of major Indian languages [109]. This composite phone set consists of 81 romanized labels that cover sounds in Hindi, Bengali, Marathi, Malayalam, Tamil, and Telugu languages. Common romanized labels are assigned to the sounds across different languages based on their perceptual similarity. We extended that idea to define a common romanized phone set that covers the sounds in both Hindi and Indian-English languages without making any changes to the labels already defined for Hindi in [109]. For defining phone labels for Indian-English, we made use of 39 CMU ARPAbet labels along with their root words [7]. Each ARPAbet has been assigned to an existing Hindi label based on the perceptual similarity of the respective English root word being typically pronounced by Indian speakers. Whereas, we assigned our own romanized labels to a few ARPAbets that do not have perceptual similarity with

[7] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

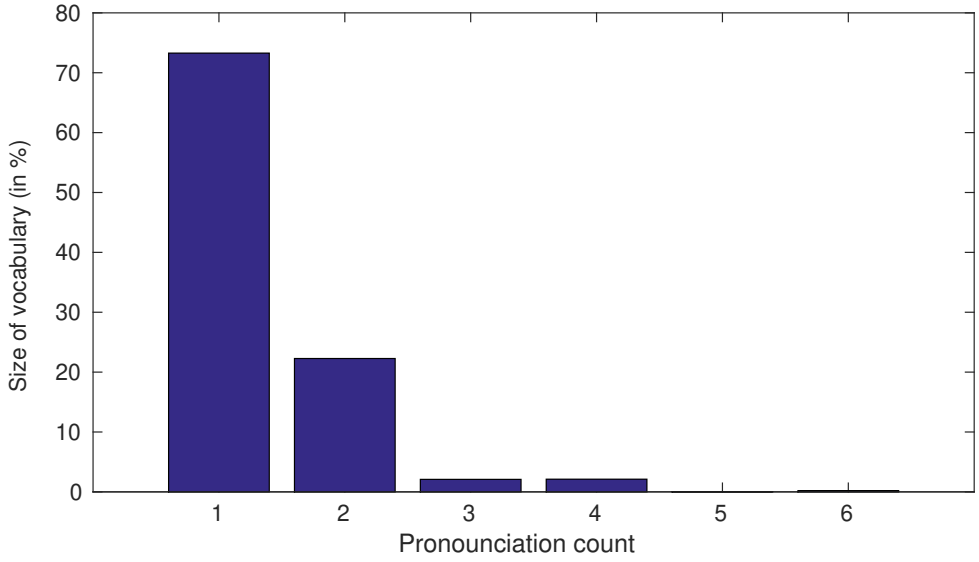
any of the Hindi phones. The complete set consists of 62 romanized labels and is referred to as the Indian real pronunciation alphabets (IRPAbet) in this work. It is worth noting that owing to the Indian accent, many ARPAbet labels get mapped to a single IRPAbet label. The lists of the IRPAbet labels that cover the sound units in Hindi and Indian-English languages are given in Table 3.3.

Table 3.3: The IRPAbet consisting of 62 labels de ned in this work for labeling the sounds in both Indian-English and Hindi languages. For Indian-English, the CMU ARPAbet labels are assigned with Hindi phone set based on perceptual similarity, while a few new labels introduced to cover Indian-English are marked in gray colour. Owing to the Indian accent, many ARPAbet labels get mapped to a single IRPAbet label.

S. No.	Hindi char.	IRPAbet	S. No.	Hindi char.	IRPAbet
1	अ	a	29	द	d
2	आ	aa	30	ध	dh
3	इ	i	31	न,ऩ	n
4	ई	ii	32	प	p
5	उ	u	33	फ	ph
6	ऊ	uu	34	ब	b
7	ऋ,ॠ	rq	35	भ	bh
8	ए	ee	36	म	m
9	ऐ	ei	37	य,य़	y
10	ओ	o	38	र,ऱ	r
11	औ	ou	39	ल	l
12	क	k	40	व	w
13	ख	kh	41	श	sh
14	ग	g	42	ष	sx
15	घ	gh	43	स	s
16	ङ	ng	44	ह	h
17	च	c	45	क़	kq
18	छ	ch	46	ख़	khq
19	ज	j	47	ग़	gq
20	झ	jh	48	ज़	z
21	ञ	nj	49	झ़	jhq
22	ट	tx	50	ड़	dxq
23	ठ	txh	51	ढ़	dxhq
24	ड	dx	52	फ़	f
25	ढ	dxh	53	◌	q
26	ण	nx	54	◌:	hq
27	त	t	55	◌ँ	mq
28	थ	th			

S. No.	ARPAbet (example)	IRPAbet	S. No.	ARPAbet (example)	IRPAbet
1	AA (odd)	ao	29	S (sea)	s
2	AE (at)	ae	30	SH (she)	sh
3	AH (hut)	a	31	T (tea)	tx
4	AO (ought)	ao	32	TH (theta)	th
5	AW (cow)	au	33	UH (hood)	u
6	AY (hide)	ai	34	UW (two)	uu
7	B (be)	b	35	V (vee)	w
8	CH (cheese)	c	36	W (we)	w
9	D (dee)	dx	37	Y (yield)	y
10	DH (thee)	d	38	Z (zee)	z
11	EH (Ed)	e	39	ZH (seizure)	z
12	ER (hurt)	er			
13	EY (ate)	ei			
14	F (fee)	f			
15	G (green)	g			
16	HH (he)	h			
17	IH (it)	i			
18	IY (eat)	ii			
19	JH (gee)	j			
20	K (key)	k			
21	L (lee)	l			
22	M (me)	m			
23	N (knee)	n			
24	NG (ping)	ng			
25	OW (oat)	o			
26	OY (toy)	oy			
27	P (pee)	p			
28	R (read)	r			

Figure 3.4: The distribution of the vocabulary based on the pronunciation count. Note that, the lexicon contains a total of 8,911 entries out of which 6,616 entries are unique.



Secondly, a unique word list is extracted from the developed HingCoS text corpus. The phoneme-level transcriptions for all those words have been done manually using the IRPAbet labels while covering all the pronunciation variations present in the speech data. Both the phonetic labels and the pronunciations are finally cross-checked by a linguist at our end. The distribution of the vocabulary based on the pronunciation count is given in Figure 3.4.

3.4 Statistical Analysis of the HingCoS Corpus

In this section, we present the salient attributes of the HingCoS text and speech corpora. First, we describe the key statistics of the HingCoS text corpus and it is followed by the details of the HingCoS speech corpus. Later, different distributions of the attributes of the text and speech corpora are plotted.

3.4.1 Analysis of the HingCoS Text Corpus

The HingCoS text corpus consists of 26k sentences that are covered by a vocabulary of 14.6k words (6029 Hindi and 8614 English). The lengths of the sentences vary from 3 to 57 words. It is worth highlighting that the collected text corpus contains about 0.1 million code-switching instances, i.e., where the bloggers have switched to English words/phrases while writing Hindi sentences. The

Table 3.4: Key statistics of the HingCoS text corpus developed in this study. Note that, the code-switching instance refers to the location where switching happens either from Hindi to English or vice-versa.

# sentences	# words		# unique words		# code-switching instances
	Hindi	English	Hindi	English	
25,988	381,603	196,556	6,029	8,614	104,912

Table 3.5: Top 10 most frequent code-switching word pairs (Hindi-to-English and vice versa) that occur in HingCoS text corpus along with their corresponding log-likelyhood probabilities.

Hindi to English		English to Hindi	
Word pair	**Log-likelihood**	**Word pair**	**Log-likelihood**
helpful रहा	-0.026	पाचों website	-0.080
proceed पर	-0.045	आप debit	-0.131
bitcoins के	-0.080	मैरा blog	-0.131
complexity को	-0.080	आपको ios	-0.156
infected है	-0.080	की audio	-0.156
modification के	-0.080	लिए unity	-0.156
depend करता	-0.101	निर्माता company	-0.228
cases में	-0.109	आपका memory	-0.249
click करें	-0.131	आपके account	-0.249
sms नही	-0.131	आपभी social	-0.249

key statistics of the created Hindi-English code-switching text corpus are summarized in Table 3.4. The most frequent code-switching word pairs (Hindi-to-English and vice versa) that occur in HingCoS text corpus are given in Table 3.5. The distributions of code-switching instances with respect to the varying length of the sentences in the HingCoS text corpus are given in Figure 3.5. Further, the plot of average code-switching instances for varying length of the sentences in the HingCoS text corpus is also computed and is shown in Figure 3.6.

The existing approach of finding POS tags for the code-switching data employs separate POS taggers corresponding to the involved languages. But it is found to yield incorrect POS tags in particular to the non-native words due to limited context information. To address the same, we recently proposed a more efficient POS tagging scheme [110] for Hindi-English text data. It consists of two steps: (i) the POS tags for the native (Hindi) words are derived conventionally, i.e., by

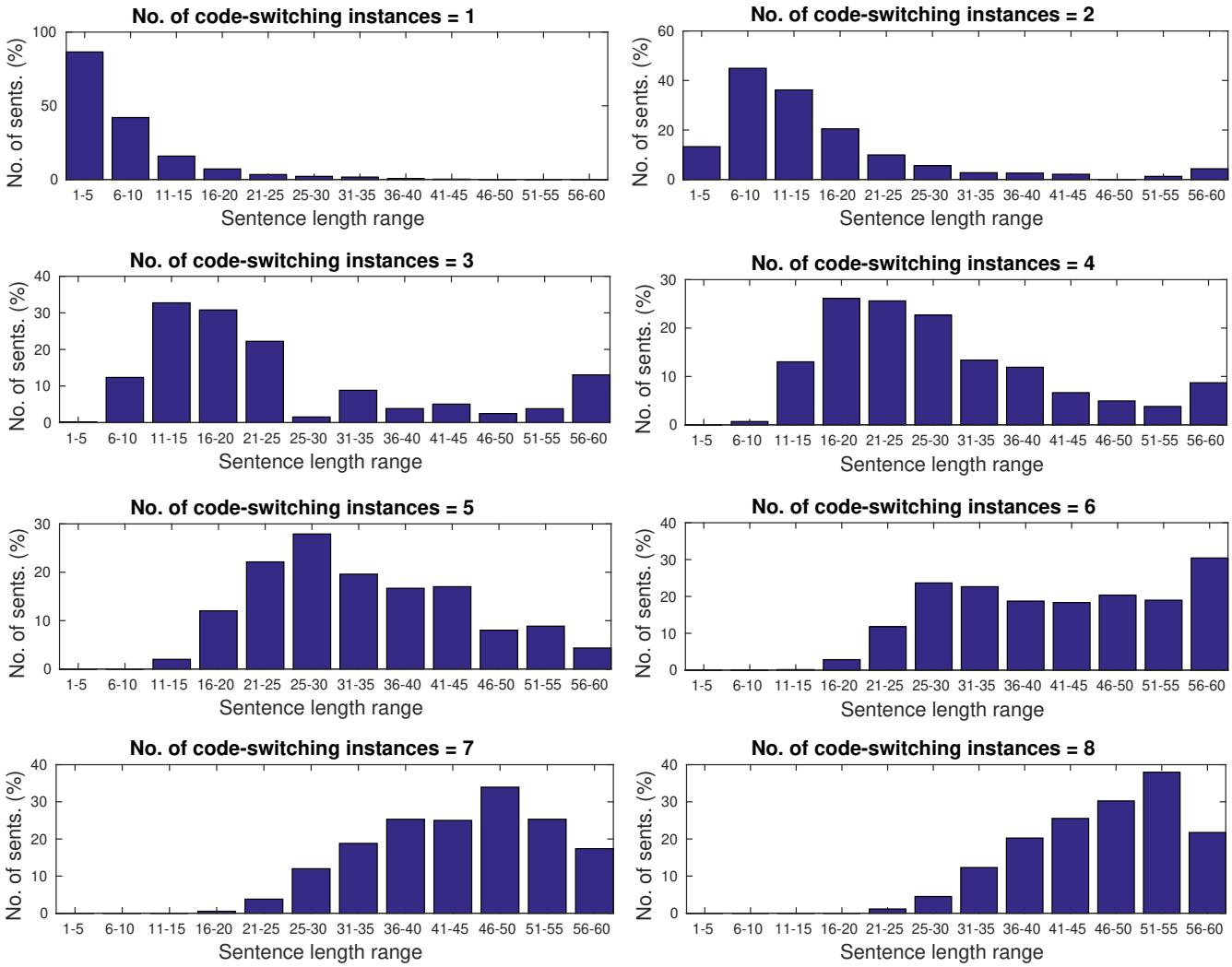


Figure 3.5: Distribution of code-switching instances for varying length of the sentences in the HingCoS text corpus. For the ease in display, the counts in each sentence length group are normalized separately, i.e., the sentence counts in a length group across 8 code-switching instances considered sum up to 100%.

passing the Hindi-English text to a Hindi POS tagger, and (ii) the given Hindi-English text is converted to pure English text through a machine translator and then the POS tags are derived using an English POS tagger. The final POS tags are derived through the distillation of the POS tags for the native and non-native words in the given Hindi-English sentence derived in the above two steps. Following the above-mentioned scheme, the POS tags for the HingCoS text corpus are derived. The distributions of the English and Hindi words present in the HingCoS text corpus with respect to their POS labels are shown in Figure 3.7. For validation purpose, a small set consisting of 100 sentences from the HingCoS text corpus are randomly selected and manually labeled for POS tags. On evaluating, the validation accuracy of the proposed POS scheme turns out to be 89.1%.

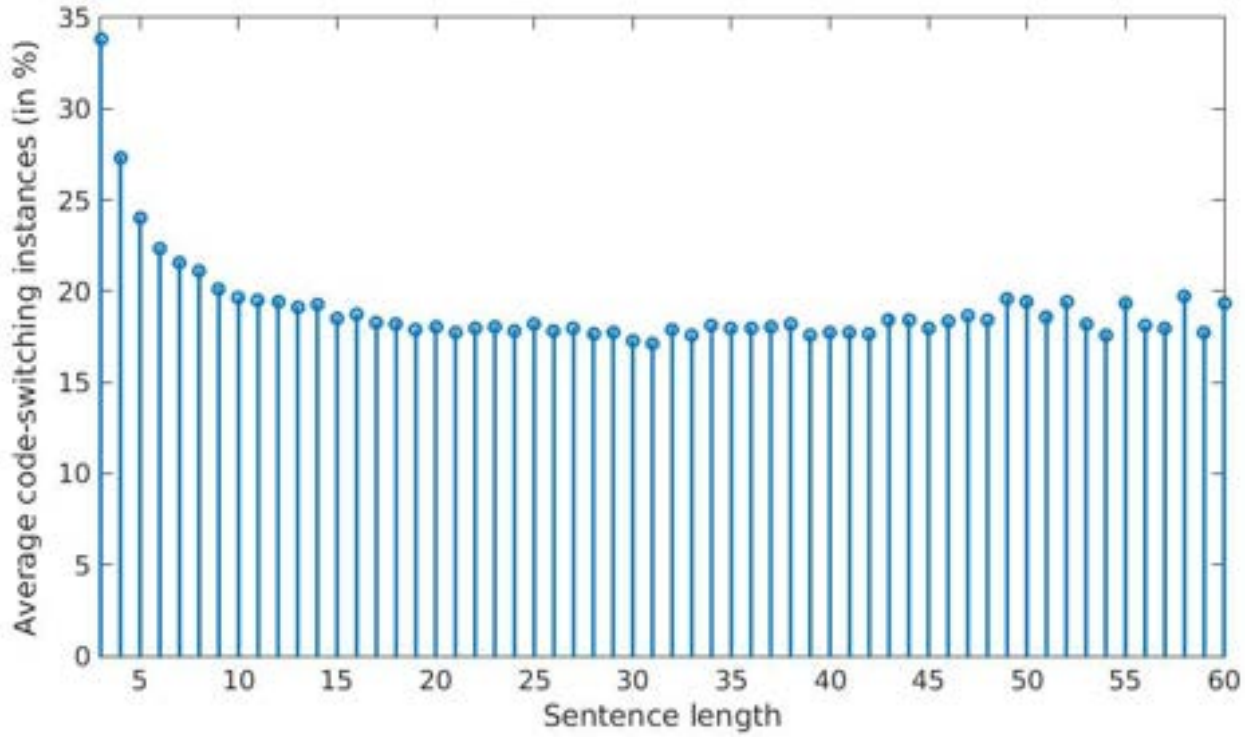


Figure 3.6: Plot of the average code-switching instances with respect to the length of sentences in HingCoS text corpus. It is to highlight that, in each of the sentences in HingCoS corpus about 20% (approx.) of the words are being code-switched.

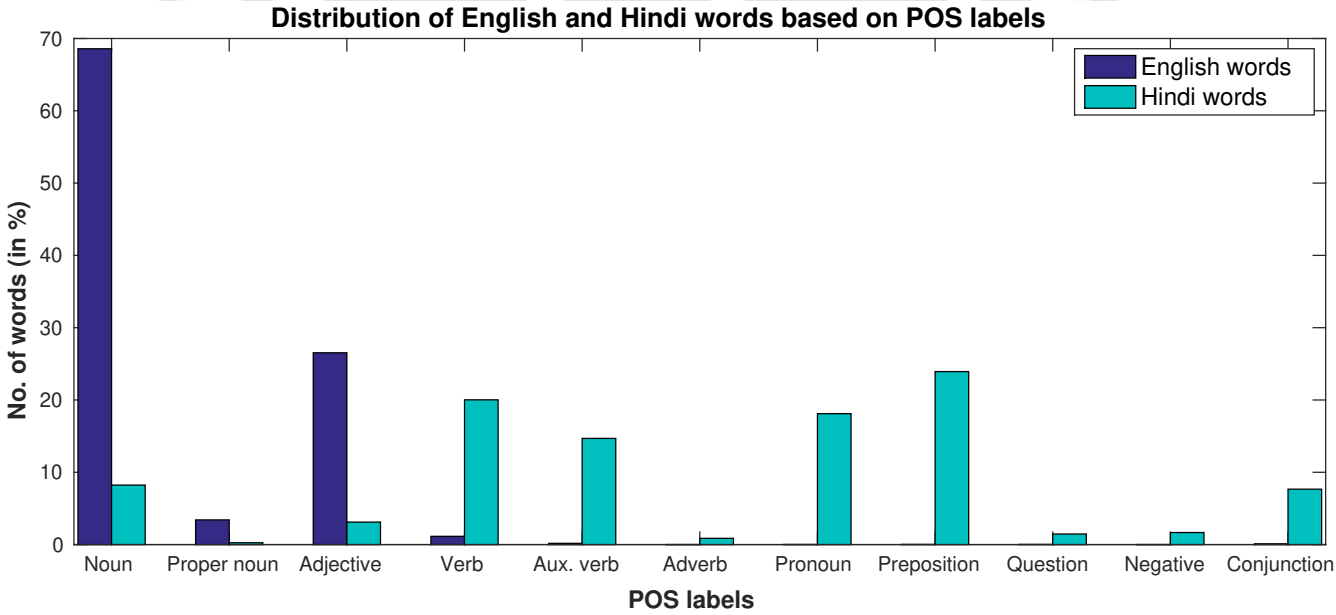
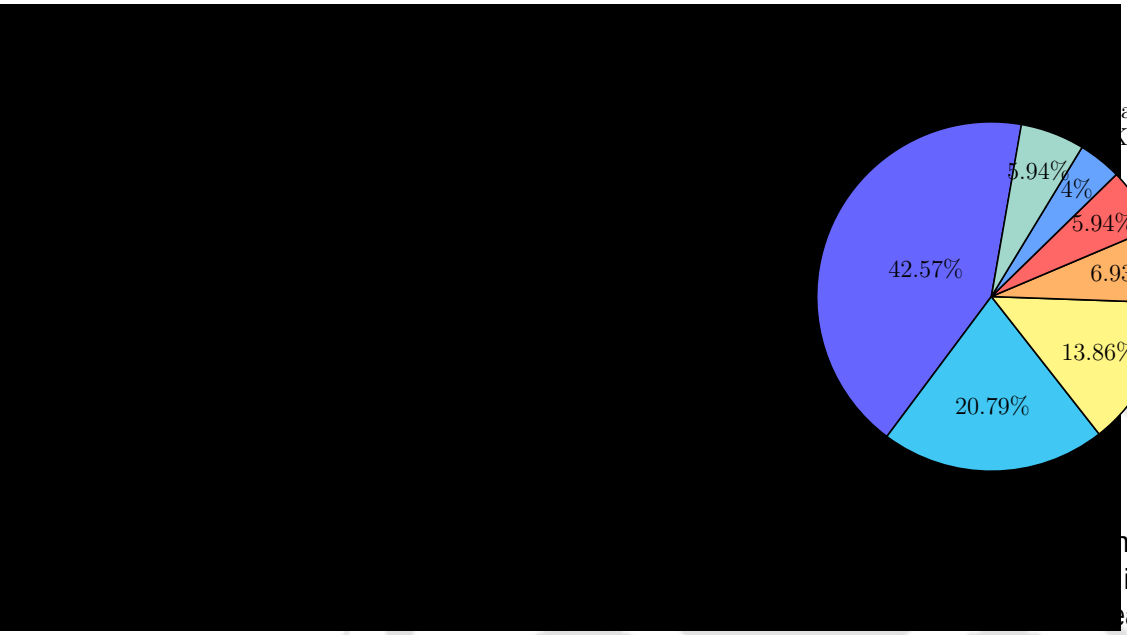


Figure 3.7: Distributions of the English and Hindi Words in the HingCoS text corpus based on their parts of speech (POS) labels. Note that, for the ease in display, the POS tags for Hindi and English words have been normalized separately, i.e., all POS labels corresponding to English/Hindi words sum up to 100%.

Table 3.6: The details of the HingCoS speech corpus developed in this study. Note that, the locations where switching happens either from Hindi to English or vice-versa are referred to as the code-switching instances.

# utterances	# words		# unique words		# code-switching instances
	Hindi	English	Hindi	English	
9,251	125,653	50,719	2,644	3,901	30,035



h as bar-plot. Whereas, indi-English speech data akers and is followed by Assamese and Bangla native speakers.

3.4.2 Analysis of the HingCoS Speech Corpus

The HingCoS speech corpus contains about 25 hours of Hindi-English speech data contributed by 101 speakers (64 male and 37 female). Table 3.6 summarizes the details about the total number of sentences, the total and the unique number of words spoken in Hindi and English portion of the data and the total number of code-switching instances in the HingCoS speech corpus. As per the meta-data collected, the speakers belong to 13 native language backgrounds in India. The distribution of the speakers' age along with the gender, and the distribution of the speakers based on their mother tongue (native language) are shown as a bar-plot and pie-chart, respectively in Figure 3.8. These speakers are native to 22 different states of India. Most of the states of India are associated with at least one distinct native language or regional dialect. Thus, the collected Hindi-English speech data happen to carry wide variations in the accent of the speakers. The native state-wise distribution of speakers is shown on the political map of India in Figure 3.9.

Figure 3.9: Distribution of the native states of the speakers in the collected speech data. The majority of speakers are from Assam and is followed by Uttar Pradesh. A total of 22 states are covered in the HingCoS speech corpus. The states of India represented in the HingCoS speech corpus are marked in the associated

State ID	State Name	# Speakers (M/F)
1	Assam	27 (M:10/F:17)
2	Uttar Pradesh	15 (M:13/F:2)
3	West Bengal	8 (M:6/F:2)
4	Bihar	8 (M:7/F:1)
5	Odisha	7 (M:3/F:4)
6	Rajasthan	6 (M:3/F:3)
7	Telangana	4 (M:3/F:1)
8	Chhattisgarh	3 (M:3/F:0)
9	Maharashtra	3 (M:3/F:0)
10	Uttrakhand	3 (M:2/F:1)
11	Delhi	2 (M:2/F:0)
12	Kerala	2 (M:1/F:1)
13	Madhya Pradesh	2 (M:1/F:1)
14	Andhra Pradesh	2 (M:2/F:0)
15	Arunachal Pradesh	2 (M:1/F:1)
16	Jharkhand	1 (M:1/F:0)
17	Tripura	1 (M:1/F:0)
18	Mizoram	1 (M:0/F:1)
19	Manipur	1 (M:1/F:0)
20	Karnataka	1 (M:1/F:0)
21	Sikkim	1 (M:0/F:1)
22	Jammu & Kashmir	1 (M:0/F:1)
	Total	**101 (M:64/F:37)**

3.5 Comparison with Existing Code-switching Corpora

In literature, a few code-switching speech and text corpora are already reported, and they happen to cover different native and non-native language combinations. In this section, we briefly summarize their salient attributes in Table 3.7, and Table 3.8 and compare them with those of the developed HingCoS corpus.

From the literature review presented in Chapter 2, it can be noted that a very small sized code-switching acoustic and linguistic resources are publicly available so far covering the Indian context. This motivated us to create moderate sized Hindi-English resources so that current technological advances in acoustic and language modeling can be explored. Among the publicly accessible code-switching corpora created in different contexts, the SEAME corpus happens to be the largest one and is followed by the HingCoS corpus reported in this work. In all the code-switching speech corpora reported so far, the speech data is recorded in clean conditions using good quality microphones.

Table 3.7: Contrastive comparison of existing code-switching speech databases reported in the literature including the HingCoS speech corpus described in this work. The developed HingCoS corpus consists of 25 hours of read speech data from 101 speakers and is de nitely one among the biggest code-switching corpora reported so far in the literature.

Reference	Name	Language pair(s)	Speech style	Dur. (hrs)	# spkr. (M/F)	Age grp.	# uttr.	Vocab.	Data recording	Access
Cao, et al. [77]	CUMIX	Cantonese-English	read	17	40 (20/20)	19-26	8,000	–	microphone; 48 kHz	Public
Lyu, et al. [76,98]	–	Mandarin-Taiwanese	read	4.8	24	–	4,600	–	–	–
Franco, et al. [99]	–	English-Spanish	conversational	0.7	3	–	–	1516	–	Public
Lyu, et al. [75]	SEAME	Mandarin-English	conversational	51.7	157	18-34	42,759	15,338	microphone; 16 kHz	Public
Shen, e al. [101]	CECOS	Chinese-English	read	12.1	77 (62/15)	20-35	6,700	–	microphone; 16 kHz	–
Modipa, et al. [102]	SPCS	Sepedi-English	read	10	20 (12/8)	17-27	450	–	–	–
Ylmaz, et al. [74]	FAME!	Frisian-Dutch	conversational	18.5	309	–	–	–	microphone; 48 kHz	Public
Ahmed, et al. [56]	–	Malay-English	read	100	208	–	–	–	microphone; 16 kHz	–
Imseng, et al. [73]	MediaParl	French-German	conversational (parliamentary debates)	6	7	–	2,617	–	microphone; 44.1 kHz	Public
Amazouz, et al. [103]	FACST	French-Arabic	read and stimulated spontaneous	7.3	20 (10/10)	23-39	–	–	–	–
Westhuizen, et al. [78]	–	English-isiZulu, English-isiXhosa, English-Setswana, and English-Sesotho	conversational (soap opera episodes)	14.3	–	–	10,343	–	microphone; 32 kHz	Public
Hamed, et al. [104]	–	Arabic-English	conversational	5.3	12 (6/6)	–	1,234	–	microphone; 16 kHz	Public
Sunit S., et al. [106]	MSR	Hindi-English	conversational	50	500	–	51,158	18,900	–	–
A. Pandey, et al. [105]	PBCM	Hindi-English	read	–	78 (40/38)	–	6,126	–	microphone; 48 kHz	–
A. Dey, et al. [69]	–	Hindi-English	conversational	–	9	–	108	–	–	–
This work [111]	HingCoS	Hindi-English	read	25	101 (61/40)	19-40	9,251	6,542	telephone; 8 kHz	Public

Table 3.8: Contrastive comparison of existing code-switching text databases reported in the literature including the HingCoS text corpus described in this work.

Reference	Language pair	Corpus name	# utterances	# words	Vocubulary	# code-switching instances
Hamed, et al. [72]	Arabic-English	–	240,874	2,590,954 81.3% Arabic and 16.5% English	–	–
Lyu, et al. [75]	Mandarin-English	SEAME	42,759	–	15,338	–
This work [111]	Hindi-English	HingCoS	25,988	578,159 66% Hindi and 34% English	14,643	104,912

Unlike those works, HingCoS speech corpus contains speech recorded in a realistic environment using a landline and mobile phones. This choice facilitates the development of telephone-based speech applications. Another unique feature of HingCoS corpus is the diversity in the linguistic background of the speakers who contributed the speech data. The native language of only 42.57% of the speakers is Hindi while the remaining ones come from other Indian language backgrounds. The speech data in HingCoS corpus is collected in read-style, while the transcripts correspond to web blogs are written in a conversational style with atleast one code-switching instance in each utterance. Note that, the data recording protocol allowed the volunteers to re-record in case of any hesitation or disfluency. Hence, the speaking style of the collected data has been referred to as the *read* speech instead of spontaneous (conversational) speech. The HingCoS corpus consists of 25 hours of read speech data from 101 speakers. This is definitely among the biggest Hindi-English code-switching corpora, carefully designed and annotated. But, the HingCoS corpus is read speech, while the conversational mode of recording is preferred to cover the natural variations in speech. However, the conversation speech brings in additional challenges while developing the ASR systems such as, huge manual effort for annotation, increase in vocabulary size, etc. These challenges forced us to record the speech in read mode.

3.6 Experimental Setups and Evaluations

The main motivation behind the creation of the HingCoS corpus is to facilitate more research in code-switching ASR task in the Indian context. In this section, the created corpus has been evaluated for the Hindi-English speech recognition task to benchmark its quality. For this purpose, both acoustic and language models are created using the appropriate data from the HingCoS corpus. The details of the acoustic and linguistic datasets, the front-end features, different acoustic and language modeling approaches employed, and the tuning of model parameters are discussed in the following subsections.

3.6.1 Acoustic and Linguistic Datasets

For language modeling, there are 25988 sentences available in the HingCoS text corpus. We have divided them into three non-overlapping groups containing 22737, 2136, and 1115 sentences

for training, testing, and development purposes, respectively. For acoustic modeling, a total of 9251 sentences out of the HingCoS text corpus spoken by native speakers are recorded. The created HingCoS speech corpus includes 2136 utterances corresponding to the above-defined linguistic test set and about 26% of the remaining text corpus (totaling 7115 utterances) for acoustic modeling purpose. The 2136 sentences are further partitioned into testing and development sets which consists of 1976 and 160 sentences, respectively.

3.6.2 Front-End Features

For a thorough experimental evaluation, the acoustic models created employing a number of front-end features are explored. The front-end signal processing has been primarily done using the standard MFCC features and the more contemporary i-vector [112] based features. The parametric details of all these features are described in the next paragraphs.

The computation of MFCC features has been done considering 25 ms of hamming windowed speech frames along with a 10 ms frameshift. Feature vector corresponding to each frame consists of a log energy coefficient (C0) and 12-dimensional MFCC features (C1-C12). For incorporating the dynamic characteristics of the vocal tract system, the 13-dimensional features obtained above have been appended with corresponding velocity and acceleration components. Hence, we finally have a 39-dimensional feature vector, which is used for acoustic modeling.

Motivated by a recent work [113], the i-vector based acoustic features are also employed for training the ASR systems. For deriving the i-vector representations of the speech data, the use of the Gaussian mixture model-based universal background model (GMM-UBM) has been done. First, the 13-dimensional static MFCC features are time-spliced to capture the dynamic information across the frames. In time-splicing, four frames on either side of the central frame are concatenated to form a 117-dimensional (13 9) feature vector, which is then projected to a 40-dimensional space using the linear discriminant analysis (LDA) [114]. On these low-dimensional features, a 1024 component gender-independent GMM-UBM is learned. The *total variability* matrix (T-matrix) for estimating the i-vectors is randomly initialized and trained using the expectation-maximization algorithm [112]. The 150-dimensional i-vectors are used in acoustic modeling.

3.6.3 Acoustic Model Training

In this study, different kinds of HMM-based hybrid modeling paradigms [1] have been employed to develop the ASR systems. We have explored the time delay deep neural network (TDNN), feed-forward neural network (FDNN), and subspace Gaussian mixture model (SGMM) based acoustic modeling approaches in addition to the traditional GMM based approach. All the experimental evaluations are performed using the Kaldi toolkit [31].

GMM-HMM system:

This system is initialized with a context-independent monophone model using 39-dimensional MFCC features. Each of the phonemes is modeled by a three-state left-to-right HMM model. Later, the cross-word triphone models are trained with a decision tree-based state tying approach to capture the contextual information. The 40-dimensional LDA-based feature vectors derived earlier are further decorrelated by employing the maximum likelihood linear transform (MLLT) [115]. The resultant features are further normalized by using feature-space maximum likelihood linear regression (fMLLR) [116], and the speaker adaptive training (SAT) [117] is performed. Later, the cross-word triphone models are trained on these normalized features.

SGMM-HMM system:

In the conventional GMM-HMM systems, a large number of model parameters are required to be estimated. The SGMM based acoustic modeling framework addresses this issue by representing the complex distribution of parameters in a compact way. Here, the HMM states share a common structure globally, and only the state-dependent model parameters are required to be estimated. Instead of estimating GMM parameters directly from the training data, the model parameters are derived from the low-dimensional model and speaker subspaces that can capture phonetic and speaker variations. As a result of that, the total number of parameter estimation is greatly reduced, which makes the learning of the model parameters possible on a limited amount of training data. In SGMM [118] based modeling techniques, the unit distributions are derived from a GMM-UBM learned on a part of training data.

FDNN-HMM system:

The FDNN-HMM [43] based acoustic modeling approach is also explored in this study. A

multi-layered FDNN is trained using time-spliced features normalized with LDA+MLLT+fMLLR as the input and computes the posterior probabilities over HMM states as the output. The parameters used in training the DNN-HMM system are given in Subsection 3.6.5.

TDNN-HMM system:

In the typical DNN architectures, the initial layers try to learn an affine transformation for the entire temporal context while training the models. But, it requires the availability of a large amount of training data to learn good transformations. This issue can be addressed by using the TDNN [119] architecture, where the initial transformations learn narrower context, and deeper layers try to learn longer temporal relationships. The specifications of the parameters used in training the TDNN-HMM system are given in the Subsection 3.6.5.

3.6.4 Language Model Training

In ASR, the LM reduces the search space while decoding the sequence of words in a sentence. Also, it helps in computing the joint probability of the word sequence $P(W)$. In this section, we discuss different LM paradigms employed for evaluation. The n-gram LMs trained by employing the IRSTLM language modeling toolkit [120] and the RNN-based LM trained using the RNNLM language modeling toolkit [121] are explored in this study.

***N*-gram language model**:

The n-gram language model predicts the next word in a sentence by using the previous $(n \quad 1)$ words [45]. In this technique, the probability of observing the word sequence $w_1, \ldots, w_N$ is approximated as

$$P(W) = \prod_{i=1}^{N} P(w_i j w_1, \ldots, w_{i-1}) \prod_{i=1}^{N} P(w_i j w_{i-(n-1)}, \ldots, w_{i-1}) \quad (3.1)$$

For this joint probability distribution, there is not enough data for modeling all the sequence lengths in a given language. So, the conditional probability is approximated to the history of previous L words, where the value of L is very much less than n. Hence, the joint probability

$P(W)$ is approximated as

$$P(W) = \prod_{i=1}^{N} P(w_i | w_{i-(L-1)}, \ldots, w_{i-1}) \tag{3.2}$$

The frequency count in the n-gram LM is given by the following equation,

$$P(w_i | w_{i-(L-1)}, \ldots, w_{i-1}) = \frac{\text{count}(w_{i-(L-1)}, \ldots, w_{i-1}, w_i)}{\text{count}(w_{i-(L-1)}, \ldots, w_{i-1})} \tag{3.3}$$

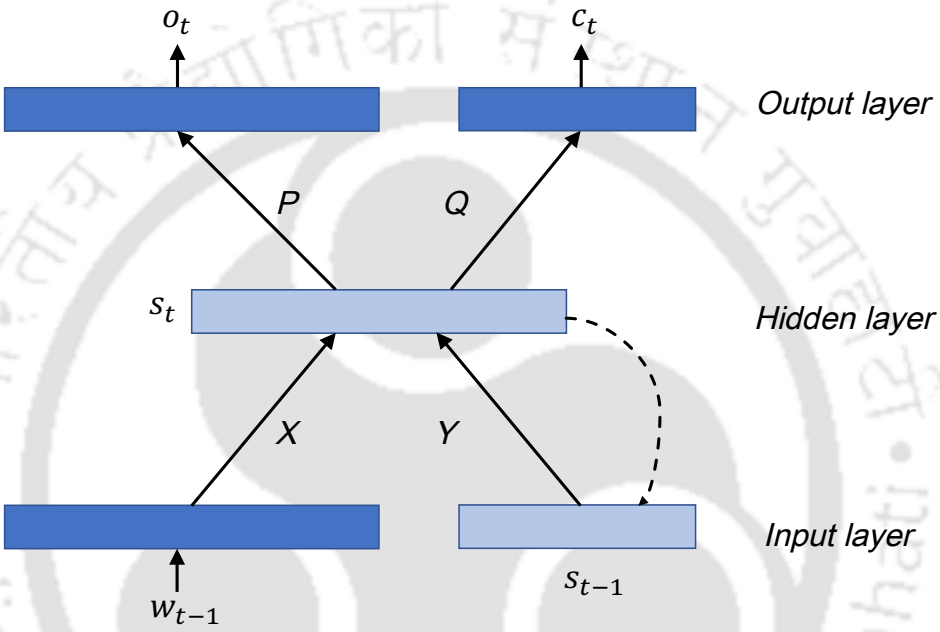


Figure 3.10: Network architecture for the class based RNNLM. The previous context s_{t-1} and the word w_{t-1} are fed as inputs for modeling the present context s_t at the hidden layer. The output layer is factorized to two parts: c_t for the word classes, and o_t for words conditioned on the classes. The $X; Y$ are weight matrices between input and hidden layer, while the $P; Q$ are weights matrices between hidden and output layers.

Recurrent neural network language model:

The RNNs possesses the ability to model the long-term dependencies with the presence of the feed-back connections. This ability of RNN has been exploited for effective language modeling in contrast to the traditional n-gram LM. The architecture for the single layered RNN employed for language modeling task is shown in Figure 3.10. Theoretically, the RNNLM compute the probability of the next word w_t, by utilizing the full history of word sequence $(w_{t-1}, \ldots, w_1)$ by the recurrent connections. From Figure 3.10, we note that, the RNNLM architecture has an input layer, a hidden layer, and an output layer. At time t, the input to the RNN is denoted by w_{t-1}, the state of the hidden layer is denoted by s_t. Whereas, the

output layer is factorized to two parts: c_t for the word classes, and o_t for words conditioned on the classes [122]. The training of RNNLM having all the words in the vocabulary in the output layer is computationally complex. To overcome this issue, the words are clustered into classes c_t based on the word counts, and then the RNNLM is trained using this class information [123]. The previous context information s_{t-1} and the word w_{t-1} are fed as inputs for modeling the present context information s_t. The output layer uses this information s_t to compute the probability of the next word w_t in the sequence. Given the context information s_t, the probability of a word w_t is approximated as a product of the probability of the class to which w_t belongs and the class conditional probability of w_t. The computations of the RNNLM are defined by the following equations.

$$s_t = f(w_{t-1}.X + s_{t-1}.Y) \tag{3.4}$$

$$c_t = g(s_t.Q) \tag{3.5}$$

$$o_t = g(s_t.P) \tag{3.6}$$

$$P(w_t|s_{t-1}) = P(c_t|s_{t-1}) \quad P(w_t|s_{t-1}, c_t) \tag{3.7}$$

where, X, Y, P, Q are the weights computed for the corresponding layers and f, g are sigmoid and softmax functions[8], respectively. The RNNLM is trained by using back-propagation through time (BPTT) [124] algorithm which helps the network to store the contextual information for several time steps in the hidden/context layer.

3.6.5 Parameter Tuning

This section describes the specifications of the parameters that are tuned to train the acoustic and language models. The tuning of acoustic and language model parameters have been done using the respective development sets defined in Section 3.6.1.

Language model:

The tuning of context length has been done on the traditional n-gram LM and the results are shown in Figure 3.11. It can be seen that the context length of 5 yields the best perplexity score on the development set.

[8]Sigmoid function is de ned as $f(x) = \frac{1}{1+e^{-x}}$ and Softmax function is de ned as $g(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}}$

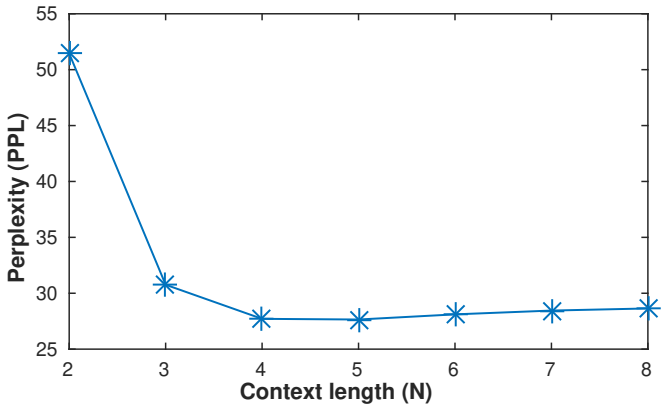


Figure 3.11: Tuning of the context length (N) on the development data. The optimal perplexity score is obtained for N = 5.

The RNNLMs used in the experiments are trained with a single hidden layer having 150 nodes and *sigmoid* as the non-linearity function. By conducting tuning experiments on the Hindi-English development data, the number of classes is set to be 100, and the variable corresponding to BPTT is set as 5.

Table 3.9: The parameters used for training the DNN and TDNN based hybrid models that are employed in this study. Note that the default parameters set by the employed toolkit are used for training the DNN model, while the parameters for TDNN model are set such that the computational complexity does not exceed the compute resources available at our end.

Parameter	**Specification**	
	DNN	**TDNN**
No. of hidden layers	5	5
No. of hidden nodes	1,024	300
No. of epochs	20	5
Size of mini batch	128	512
Initial learning rate	0.015	0.015
Final learning rate	0.002	0.002

Acoustic model:

The context-dependent GMM acoustic models are trained by tuning the number of senones. After tuning, the number of senones is set to be 2500, and the number of Gaussian mixtures per senone is set to be 8 in all the cases. In this work, for training the SGMM, 400 Gaussians are selected for training the UBM. For training the DNN-based AMs, the default parameters set by the toolkit are employed, while for TDNN case, the parameters are set based on the

Table 3.10: Recognition performances in terms of perplexity for n-gram LM and RNNLM trained on the Hindi-English training dataset and evaluated on Hindi-English development and test sets.

Dataset	Perplexity		%OOV
	5-gram LM	RNNLM	
Development	27.65	18.77	0.04
Test	62.29	40.13	0.52

available compute resources. The parameters used for training the DNN and TDNN based AMs are given in Table 3.9. Note that, the performance fo the DNN and TDNN based systems can be further improved by proper tuning of parameters.

3.6.6 Evaluation Results

The evaluation of HingCoS text and speech corpora has been done in the language modeling and speech recognition domains separately, and the results are reported in the following.

Language modeling:

Both n-gram and RNN based LMs are developed using $22,737$ training sentences in the text data having a wordlist of 14k. Table 3.10 shows the LM performances in terms of the perplexity as well as the percentage out-of-vocabulary (OOV) words for both development and test datasets. In case of n-gram LM, the value of n is fixed to be 5 after tuning done on the development data as shown in Figure 3.11. From Table 3.10, we can note that the RNNLM has resulted in consistently better recognition performances in contrast to the n-gram LM on both the development and test sets.

Speech recognition:

The experimental studies have been conducted for 4 different modeling paradigms: GMM-HMM, SGMM-HMM, DNN-HMM, and TDNN-HMM. For contrast purposes, the acoustic models are also trained on the combined phone set of size 94 (Hindi 55 and English 39) along with the proposed IRPAbet phone set of size 62. In decoding, the 5-gram and the RNNLMs discussed in Table 3.10 are employed. The evaluation results of HingCoS speech corpus in terms of word error rate (WER) are given in Table 3.11. On considering the 5-gram LM, among all the systems developed, the TDNN-HMM based system trained on the IRPAbet

Table 3.11: Evaluation of Hindi-English code-switching speech corpus in context of ASR task. The performance results in terms of percentage word error rate (%WER) along with the 95% con dence interval in brackets are reported for the test set comprising of 1976 sentences as de ned in Section 3.6.1. Along with the proposed IRPAbet phone set-based systems, the evaluation has also been done for the systems trained by combining both the Hindi and English phone sets de ned in Table 3.3.

Phone model	AM	Front-end features	LM	% WER (95% confidence interval)	
				IRPAbet	Combined
Monophone	GMM	MFCC	5-gram	52.9 (0.50)	56.1 (0.50)
Triphone		MFCC		32.9 (0.47)	35.3 (0.48)
		MFCC + LDA		31.9 (0.47)	34.8 (0.48)
		MFCC + LDA + SAT		27.6 (0.45)	31.2 (0.47)
	SGMM			24.3 (0.43)	27.2 (0.45)
	DNN			22.8 (0.42)	26.4 (0.44)
	TDNN	MFCC + i-vector		20.7 (0.41)	23.9 (0.43)
		MFCC + i-vector	RNN	**19.5 (0.40)**	21.3 (0.41)

phone set using MFCC plus i-vector front-end features yields the best WER. On the use of RNNLM for rescoring, further improvement in the performance of the TDNN-HMM system is noted. These trends are consistent with those reported in the literature.

3.7 Conclusions

To explore the current technological advances in acoustic and language modeling of code-switching data, we have developed a moderate sized Hindi-English code-switching corpus referred to as HingCoS corpus. This corpus consists of code-switching text data having 26k sentences with a total of 0.58 million words. In addition to that, the corpus also contains 25 hours of matching speech data corresponding to 9251 code-switching sentences covering a vocabulary of 6542 words. Also, the corpus includes a lexicon covering the majority of the pronunciation variations for each of the words present in the collected speech data. For the creation of the lexicon, a common phone set that covers all basic sounds present in both Hindi and Indian-English is defined. It consists of 62 romanized labels and is referred to as the Indian real pronunciation alphabets (IRPAbet) in this thesis. Among the publicly accessible code-switching corpora created in different contexts, the SEAME (Mandarin-English) corpus happens to be the largest one and is followed by the created HingCoS corpus. Unlike the existing code-switching speech corpora reported so far, the HingCoS corpus contains speech recorded in a realistic environment using landline and mobile phones. The

speech data is collected in read-style, while the transcripts correspond to web blogs are written in a conversational style. The HingCoS corpus is made public, and the details for accessing the same are available on the corpus webpage [9]. Also, for the sake of sanity check, several baseline ASR systems are developed on HingCoS corpus by following the hybrid framework. Among the developed systems, the TDNN-HMM based system resulted in the best WER of 19.47%. In the following chapters, we have explored the challenges in acoustic and language modeling of code-switching data by employing the HingCoS corpus.

[9] https://www.iitg.ac.in/eee/emstlab/HingCoS_Database/HingCoS.html

4

Exploration of End-to-End Framework for Code-Switching Speech Recognition

Contents

4.1	Variants of E2E ASR Frameworks	50
4.2	E2E Hindi-English Code-Switching ASR Systems	54
4.3	Context-Dependent T2W Transduction	60
4.4	Experimental Setup	61
4.5	Experimental Results	65
4.6	Conclusions	67

In Chapter 3, the developed Hindi-English code-switching corpus referred to as the HingCoS corpus has been discussed in detail. Following that, we also have developed the baseline ASR systems on the said corpus by employing the HMM-based ASR framework. However, as discussed in Chapter 1, the HMM-based ASR framework is limited by several factors such as multi-module training, conditional independence hypothesis, and data forced segmentation alignment. Towards addressing those issues in the HMM-based framework, in the recent past, the E2E framework was proposed and successfully explored in the monolingual ASR task [33, 49–51, 125, 126]. The E2E framework has a unified DNN architecture that is trained with a global objective function. The network is trained with characters as the output targets, given the acoustic features as input. Thus, the E2E ASR framework is a simplified model with joint optimization criteria and does not require any phonetically labeled training data.

4.1 Variants of E2E ASR Frameworks

In this thesis, we have explored two variants of the E2E framework for developing an ASR system for Hindi-English code-switching data, namely, (i) the CTC based framework [53, 127], and (ii) the sequence-to-sequence modeling with an attention mechanism based framework [33, 49]. In the following, first, the details of both those E2E frameworks are presented. We then discuss the development of the E2E ASR systems for Hindi-English code-switching data by employing the HingCoS corpus.

4.1.1 CTC-based E2E Framework

CTC based E2E framework consist of a deep bidirectional long short term memory (DBLSTM) encoder which is trained to minimize the CTC cost function as shown in Figure4.1. These components are described below in detail.

4.1.1.1 DBLTSM Network

The DBLSTM is a prominent sequence modeling architecture. It combines the advantage of multiple levels of representation that come from the use of a deep network along with long range context enabled by the use of LSTM networks. Conventional LSTMs process sequence data from left to right, thus making use of only the previous context. A detailed summary of the LSTM

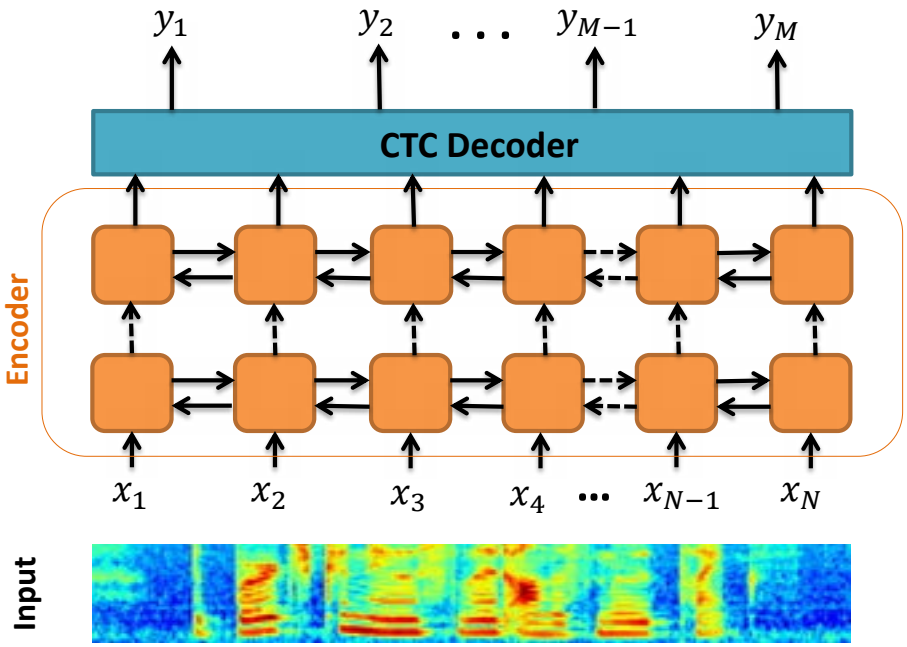


Figure 4.1: Architecture of the CTC-based E2E network. The encoder is a deep network consisting of BLSTM cells. Given a target transcription $\boldsymbol{y}$ and the input feature vector $\boldsymbol{x}$, the network is trained to minimize the CTC cost function as CTC($\boldsymbol{x}$) = log P($\boldsymbol{y}|\boldsymbol{x}$).

network is given in Appendix A. In speech recognition tasks, making use of future context can be useful. BLSTMs process input data in both directions with separate hidden layers which are fed forward to the same output layer. The following equations illustrate the calculation of forward and backward activations:

$$\overrightarrow{h}_t = H(W_{x,\overrightarrow{h}} x_t + W_{\overrightarrow{h},\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \tag{4.1}$$

$$\overleftarrow{h}_t = H(W_{x,\overleftarrow{h}} x_t + W_{\overleftarrow{h},\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \tag{4.2}$$

where $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ represent the forward and backward activations respectively, W terms denote the weight matrices, i.e., $W_{x,\overrightarrow{h}}$ is the forward weight matrix between input and hidden layers, and b terms denote the bias vectors, i.e., $b_{\overrightarrow{h}}$ is the forward bias vector for hidden layer. H is the hidden layer function which is usually a sigmoid function applied element wise. The other terms have their conventional meanings as defined in [128]. The output y_t is given by

$$y_t = W_{\overrightarrow{h},y} \overrightarrow{h}_t + W_{\overleftarrow{h},y} \overleftarrow{h}_t + b_y \tag{4.3}$$

The network is trained to minimize the CTC loss function as explained in the following section.

4.1.1.2 CTC Cost Function

CTC allows training of the network without requiring a prior alignment between input and output sequences. In CTC, the output softmax layer of the network has one unit each for the targets in addition to a blank symbol ϕ denoting a null emission. For a given training speech example, there are as many possible alignments as there are ways of separating the labels with blanks. At every time-step, the network decides whether to emit a symbol or not. As a result, a distribution over all possible alignments between the input and target sequences is obtained.

Finally, CTC employs a dynamic programming based forward-backward algorithm to obtain the sum over all possible alignments and produces the probability of output sequence given a speech input. Given a target transcription $\boldsymbol{y}$ and the input feature vector $\boldsymbol{x}$, the network is trained to minimize the CTC cost function as

$$\text{CTC}(\boldsymbol{x}) = \quad \log \text{P}(\boldsymbol{y}|\boldsymbol{x}) \tag{4.4}$$

where $\log \text{P}(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{a} \in \boldsymbol{\beta}(\boldsymbol{y},\ \boldsymbol{x})} \text{P}(\boldsymbol{a}|\boldsymbol{x})$, a is an alignment, and $\beta(\boldsymbol{y},\ \boldsymbol{x})$ is the set of all possible sequences between $\boldsymbol{y}$ and $\boldsymbol{x}$.

4.1.2 Attention-based E2E Framework

In this section we describe one of the popular attention-based E2E architecture referred to as listen, attend, and spell (LAS). The LAS architecture comprises of three modules: listener, attender, and speller. The listener is a pyramidal architecture consisting of BiLSTM cells. It acts as an encoder and transforms an input feature vector $\boldsymbol{x}$ into a higher order vector representation $\boldsymbol{h}$. The encoded output vector $\boldsymbol{h}$ along with the decoder state s_i is passed to the attender. At every time instance, the attender takes $\boldsymbol{h}$ and decoder state s_i as the inputs and outputs the context c_i. It acts like an alignment generator determining which encoded features in $\boldsymbol{h}$ should be attended for accurate prediction of the current output symbol y_i. The output of this attention module c_i is then passed to the speller, which is an LSTM decoder. It takes the context information c_i as well as the previous prediction y_{i-1} in order to predict the current symbol y_i. The listener, attender and the speller are trained together to minimize the cross-entropy loss and thus making it a complete end-to-end system. The typical architecture of the LAS network is shown in Figure 4.2.

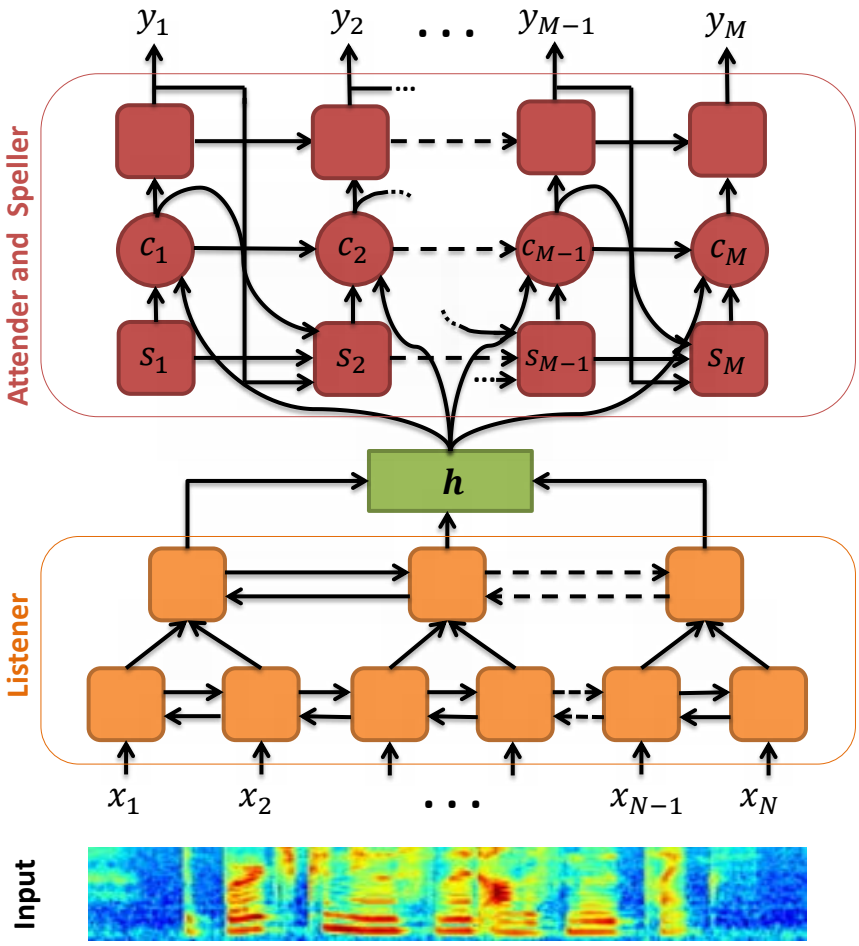


Figure 4.2: Architecture of LAS network. It consists of three modules namely: listener (a BiLSTM encoder), attender (an alignment generator), and speller (an LSTM decoder).

The mathematical representations of each step in the LAS architecture are given as

$$\boldsymbol{h} = \text{Listener}(\boldsymbol{x}) \tag{4.5}$$

$$c_i = \text{Attender}(\boldsymbol{h}, s_i) \tag{4.6}$$

$$s_i = \text{LSTM}(y_{i-1}, s_{i-1}, c_{i-1}) \tag{4.7}$$

$$p(y_i|\boldsymbol{x}) = \text{Speller}(c_i, y_{i-1}). \tag{4.8}$$

The entire network is trained to optimize the log probability given below.

$$\max_{\lambda} \sum_{i} \log \text{P}(y_i|\boldsymbol{x}, y^*_{<i}; \lambda) \tag{4.9}$$

where λ represents the LAS model parameters and $\mathbf{y}^*_{<i}$ refers to the ground truth of the previously decoded targets. For more detailed explanation on LAS network, the readers are referred to [50].

As already discussed, the E2E network is trained with characters as the output targets and hence does not require the phonetically labeled training data. For multiple languages being involved, these attributes become more attractive in the case of code-switching ASR. Motivated by that, the recent works have explored the E2E framework in the code-switching ASR domain [58–61]. In the existing code-switching, E2E ASR works, the target set is derived by simply combining the character sets of the languages involved. It is argued that such systems would suffer from high confusability among the cross-language targets unless a sufficiently large amount of data is available for training. The possible cause of the confusability lies in the broad acoustic similarity among sound units involved in most of the code-switching language pairs. Also, for the enhanced target set, such systems would exhibit high computational complexity. In the context of low resourced modeling, one can avoid such a confusability if a common phone set covering the underlying languages in code-switching data is used as the output target.

Motivated by the above-cited reasons, we first explore the earlier defined common phone set (i.e., the IRPAbet labels defined in Chapter3) as a reduced target set for developing an E2E Hindi-English code-switching ASR system using the *HingCoS* corpus. For contrast purposes, the E2E ASR systems were also developed using the combined character set as targets. In the following, we present a detailed discussion on the development of those E2E ASR systems.

4.2 E2E Hindi-English Code-Switching ASR Systems

In this section, we report a maiden exploration of the E2E framework for the Hindi-English code-switching ASR task. The primary experimentation has been done in the context of attention-based E2E framework. Later, the proposed techniques are revalidated in the context of the CTC-based E2E framework for completeness.

4.2.1 Combined Target Set Modeling

Typically, the E2E ASR systems are trained for the character set of the spoken language as the output target, given the acoustic features. In the languages which involve both upper and lower case characters, the transcription is normalized to either of the cases. Thus, in the context of code-switching, such systems have to model the combined character set of the underlying languages. In

English word	Hindi word (English Translation)
Bus	बस (enough)
Fool	फूल (flower)
May	में (in)
Say	से (from)
Lay	ले (take)

Table 4.1: The character sets of Hindi (68) and English (26) languages that are used as targets in building the conventional E2E Hindi-English code-switching ASR system.

Hindi characters	अ आ इ ई उ ऊ ऋ ए ऐ ओ औ क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न प फ ब भ म य र ल व श ष स ह क़ ख़ ग़ ज़ झ़ ड़ ढ़ फ़ ो ॉ ॆ ् ौ ृ ू ु ी े ़ ा ि ँ ः ं
English characters	a b c d e f g h i j k l m n o p q r s t u v w x y z

Table 4.2: Evaluation of the E2E ASR system trained on the combined target set for the Hindi-English code-switching task. The WER is computed after the transduction of the hypothesized targets. The percentage of invalid words generated indicate the naiveness of the transduction scheme employed.

%TER	%WER	%$\langle unk \rangle$
21.85	33.92	8.2

Word	**Combined target set**	**Reduced target set**
bharat	b h a r a t	bh aa r a t
भारत	भ ा र त	bh aa r a t
english	e n g l i s h	i ng l ii sh
अंग्रेज़ी	अ ं ग ् र े ज ़ ी	a ng r e j ii

this work, this approach is referred to as the combined target set modeling. The character sets of the Hindi and English languages are shown in Table 4.1. Using those, we built an E2E Hindi-English code-switching ASR system with 95 targets comprising Hindi (68), English (26), and a special character '_' used for separating the words. The TER of the developed system using the combined target set is reported in Table 4.2. For reference purposes, a few decoded character sequences are shown in Table 4.3. For converting the hypothesized target sequences to their corresponding word sequences, a scheme usually followed in the literature is employed. The same has been referred to as the *naive transduction* scheme in this work. In that scheme, all-white spaces between the targets are removed first, and then each of the '_' labels is replaced by a single space to derive the hypothesized word sequence.

From Table 4.3, it can be noted that the output character sequences often get corrupted with cross-lingual character substitutions. Thus, even for a single character in the output sequence being misclassified, the naive transduction scheme would not find any valid word match in the task wordlist. In those cases, it outputs a *unknown* ($\langle unk \rangle$) label, which in turn degrades the WER. Table 4.2 also shows the WER for the developed E2E system along with the percentage of $\langle unk \rangle$ labels in the output. The high percentage of $\langle unk \rangle$ labels shows the naiveness of the transduction scheme. The issue of cross-lingual target confusability can be addressed either by considering a sufficiently large amount of speech data for acoustic modeling or by deriving the target labels by exploiting the acoustic similarity. In the next subsection, we attempt to reduce the confusability among the cross-lingual targets.

Word	Combined target set	Reduced target set
hindi	h i n d i	h i n dx ii
हिंदी	ह ि ं द ी	h i n dx ii
english	e ng l i sh	i ng l ii sh
अंगरेजी	अ ं ग र े ज ी	a ng r ei j i

Table 4.3: Two sample decoded output sequences of the attention-based E2E ASR system developed using a combined target set for the Hindi-English code-switching task. The English translations of the sentences are given in the braces. The errors obtained in the hypothesized character sequences have been highlighted. Note that the symbol '_' is used to mark the word boundaries. The invalid words produced by the transduction process are labeled as *hunk i*.

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us page*) **Ref. target sequence:** c o m p a n y _क े _ a b o u t _ u s _ p a g e _ म े ं _ ज ा न क ा र ी _ ह ै **Hyp. target sequence:** क o m p a n y _क े _ए b o u t _ u s _ p a g e _ म े ं _ ज ा न क ा र ी _ ह ै **Hyp. sentence:** *<unk>* के *<unk>* us page में जानकारी है
Example 2	**Ref. sentence:** आपको hindi में blogging शुरू करनी चाहिए (*you should start blogging in hindi*) **Ref. target sequence:** आ प क ो _ h i n d i _ म े ं _ b l o g g i n g _ श ु र ू _ क र न ी _ च ा ह ि ए **Hyp. target sequence:** आ प क ो _ h i n d i _ म े ं _ b l o g g i n g _ श ु र ू _ क र न ी _ च ा ह ि ए **Hyp. sentence:** आपको hindi में blogging शुरू करनी चाहिए

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us page*) **Ref. target sequence:** k a m p a n ii _ k ee _ a b au tx _ a z _ p ei j _ m ee _ j aa n k aa r ii _ h ei **Hyp. target sequence:** k a m p a n ii _ k ee _ a b au tx _ a **s** _ p ei j _ m ee q _ j aa n k aa r ii _ h ei **Hyp. sentence:** company के about *<unk>* page में जानकारी है
Example 2	**Ref. sentence:** आपको hindi में blogging शुरू करनी चाहिए (*you should start blogging in hindi*) **Ref. target sequence:** aa p k o _ h i n dx ii _ m ee _ b l aa g i ng _ sh u r uu _ k a r n ii _ c aa h i ee **Hyp. target sequence:** aa p k o _ h i n dx ii _ m ee _ b l aa g i ng _ sh u r uu _ k a r n ii _ c aa h i ee **Hyp. sentence:** आपको **हिंदी** में blogging शुरू करनी चाहिए

4.2.2 Reduced Target Set Modeling

The reduced target set modeling refers to employing fewer target labels than those involved in the combined target set modeling based E2E code-switching ASR system. Recently, in the context of the multilingual ASR task [129], the authors successfully used the union of phone sets of the underlying languages as targets to the E2E ASR system instead of the combined character set. Motivated by that, we have employed a common phone set, i.e., the IRPAbet labels defined in Chapter3, having 62 labels that cover both Hindi and English languages. For the ease of reference, the said phone set creation is briefly outlined next. We borrowed the phone set for the Hindi language from a composite phone set covering the majority of the Indian languages already defined for computer processing [109]. As the Hindi phone set is bigger, the English phones were heuristically mapped to corresponding Hindi phones having a broad acoustic similarity. And those which could not be mapped to Hindi phones were given unique labels. In this work, we employ that common phone set along with the special character '_' as the reduced target set in training the E2E ASR system for the Hindi-English code-switching task. The reduced target set based E2E ASR system was trained following the identical setup as used for the combined target set based system discussed in the previous subsection. In Table 4.4, we show the decoded outputs produced by the reduced target set based E2E ASR system for the same set of example sentences as considered in Table 4.3. On comparing those tables, it can be noted that the reduced target set based E2E system exhibits a reduction in the cross-lingual target confusability and thus resulting in improved TER performance

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us page*) **Ref. target sequence:** c o m p a n y _ क े _ a b o u t _ u s _ p a g e _ म े ं _ ज ा न क ा र ी _ ह ै **Hyp. target sequence:** क o m p a n y _ क े _ क् b o u t _ u s _ p a g e _ म े ं _ ज ा न क ा र ी _ ह ै **Hyp. sentence:** *<unk>* के *<unk>* us page में जानकारी है
Example 2	**Ref. sentence:** आपको hindi में blogging शुरू करनी चाहिए (*you should start blogging in hindi*) **Ref. target sequence:** आ प क ो _ h i n d i _ म े ं _ b l o g g i n g _ श ु र ू _ क र न ी _ च ा ह ि ए **Hyp. target sequence:** आ प क ो _ h i n d i _ म े ं _ b l o g g i n g _ श ु र ू _ क र न ी _ च ा ह ि ए **Hyp. sentence:** आपको hindi में blogging शुरू करनी चाहिए

Table 4.4: Two sample decoded output sequences of the attention-based E2E ASR system trained on the reduced target set for the Hindi-English code-switching task. For contrast purposes, the sentences are kept the same as considered in Table 4.3. Note that the symbol '_' is used to mark the word boundaries. The invalid words produced by the transduction process are labeled as *hunki*.

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us page*) **Ref. target sequence:** k a m p a n ii _ k ee _ a b au tx _ a z _ p ei j _ m ee _ j aa n k aa r ii _ h ei **Hyp. target sequence:** k a m p a n ii _ k ee _ a b au tx _ a **s** _ p ei j _ m ee q _ j aa n k aa r ii _ h ei **Hyp. sentence:** company के about *<unk>* page में जानकारी है
Example 2	**Ref. sentence:** आपको hindi में blogging शुरू करनी चाहिए (*you should start blogging in hindi*) **Ref. target sequence:** aa p k o _ h i n dx ii _ m ee _ b l ao g i ng _ sh u r uu _ k a r n ii _ c aa h i ee **Hyp. target sequence:** aa p k o _ h i n dx ii _ m ee _ b l ao g i ng _ sh u r uu _ k a r n ii _ c aa h i ee **Hyp. sentence:** आपको **हिंदी** में blogging शुरू करनी चाहिए

Table 4.5: Evaluation of the E2E ASR system trained on reduced target set for Hindi-English code-switching task.

%TER	%WER	%*hunki*
18.1	40.19	6.3

measure as given in Table 4.5.

For converting the reduced target set sequence to corresponding word sequence, a pronunciation dictionary for all Hindi and English words in the HingCoS corpus is created. During T2W transduction, each target segment separated by '_' labels is searched in the created pronunciation dictionary and is replaced with the word corresponding to it. In the case of homophone words, the one having the highest unigram count is chosen. If there is no match, then that target segment is replaced with the *hunki* label. Following that, each of the '_' labels is replaced by a single space to produce the hypothesized word sequence. The WER, along with the percentage of *hunki* labels, are also reported in Table 4.5. On comparing Tables 4.2 and 4.5, the proposed reduced target set modeling scheme is noted to provide a substantial reduction in the TER as well as the *hunki* labels in the output. On the flip side, the WER gets significantly degraded. The possible causes of WER degradation are (i) the naivety in the employed T2W transduction scheme, and (ii) the enhanced confusability among the homophones (the words having identical pronunciation but different spellings) within or across the languages involved in code-switching. In the following, the mentioned issues and the proposed approach to overcome those issues are discussed in detail.

what is the relationship between class and object	

Hindi characters	अ आ इ ई उ ऊ ऋ ए ऐ ओ औ क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण प फ ब भ म य र ल व श ष स ह क़ ख़ ग़ ज़ झ़ ड़ ढ़ फ़ ो ॉ ॊ ो ौ ॎ ॏ ॐ ॑
English characters	a b c d e f g h i j k l m n o p q r s t u v w x y z

Table 4.6: Examples of di erent types of homophones present in the Hindi-English code-switching data. In the HingCoS corpus, there are a total of 96 homophones.

Proper nouns	Collective/ abstract nouns	Abbreviations	Intra-language	Inter-language
amit - अमित hindi - हिंदी japan - जापान	internet - इंटरनेट ticket - टिकट station - स्टेशन	ATM - एटीएम USA - यूएसए CEO - सीईओ	sea-see due - dew dye - die	fool - फूल bus - बस say - से

4.2.2.1 Naivety in T2W transduction

We first highlight the weakness of the T2W transduction approach typically employed to produce WERs for E2E ASR systems. Let $\{T_{h_i}\}_{i=1}^{n}$ be the segmented hypothesized target sequence, $\{T_{c_j}\}_{j=1}^{m}$ be the segmented corresponding target sequence, and $\{W_i\}_{i=1}^{m}$ be the desired word sequence associated to $\{T_{c_j}\}_{j=1}^{m}$. The objective of the decoder is to determine the desired word, given the corresponding segment of the hypothesized target sequence. In this transduction approach, the desired word W_i is produced by the decoder only when an exact match for a segment T_{h_i} is found in the pronunciation dictionary D. For a single entry of T_{h_i} being in error, no valid word corresponding to it would be found in D. In such cases, the *hunkit* label is emitted. More formally, the same can be expressed by a decoder function F as

$$F(T_{h_i}) = \begin{cases} W_i & \text{if } T_{h_i} \in D \\ hunkit & \text{otherwise.} \end{cases} \tag{4.10}$$

From Eqn. 4.10, it can be noted that, unless the TER of the E2E ASR system is very low, the derived word-level hypothesis will lead to a degraded WER due to the presence of *hunkit* labels.

Ref. sentence	क्या आपने google web light से अपने stats में traffic को notice किया
Naive approach	क्या आपने google web <unk> से अपने <unk> में traffic co notice किया
Sample *k*-best list	क्या आपने google web light से अपने stats में traffic co notice किया क्या आपने google web light से अपने stats में traffic को notice किया क्या आपने google web light से अपने status में traffic co notice किया क्या आपने google web light से अपने status में traffic को notice किया क्या आपने google web lite से अपने stats में traffic co notice किया क्या आपने google web lite से अपने stats में traffic को notice किया क्या आपने google web lite से अपने status में traffic co notice किया क्या आपने google web lite से अपने status में traffic को notice किया

Word	Combined target set	Reduced target set
hindi	h i n d i	h i n dx ii
हिंदी	हि ं दी	h i n dx ii
english	e n g l i s h	i ng l ii sh
अंगरेज़ी	अ ं ग र े ज़ ी	a ng r ei j i

4.2.2.2 Homophone confusability

Almost every language has some homophones with the English language having a large number of them. In the context of code-switching, the homophone issue gets further enhanced as they may occur across the languages too. For the ease of illustration, we have grouped them into five broad categories, namely proper nouns, internalized collective/abstract nouns, abbreviations, intra-language, and inter-language homophones. This categorization has been done based on parts-of-speech and language information. Table 4.6 lists a few examples of those homophone categories for the Hindi-English code-switching case. There are 96 homophones in the HingCoS corpus, and their

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us p* **Ref. target sequence:** c o m p a n y _ क े _ a b o u t _ u s _ p a g e _ म े ं _ ज ा न क ा र ी **Hyp. target sequence:** क ं प न ी _ क े _ ब ा उ ट _ अ स _ प ा ग े _ म े ं _ ज ा न क ा र ी **Hyp. sentence:** <unk> के <unk> us page में जानकारी है

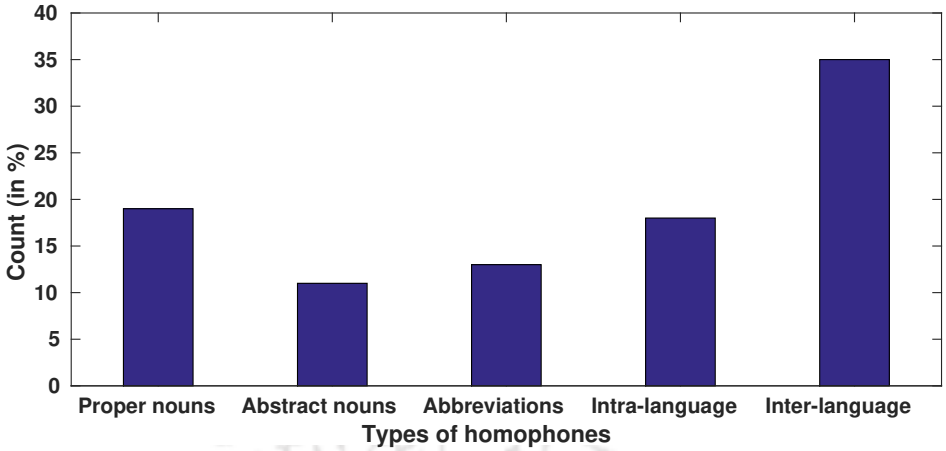


Figure 4.3: Histogram of di erent types of homophones present in the HingCoS corpus. It is worth noting that a large number of homophones in the HingCoS corpus belong to the inter-language category.

distribution in terms of earlier defined broad categories is shown in Figure 4.3. It is worth noting that a large number of homophones belong to the inter-language category. Despite the reduction of target set confusability with the proposed target set, it was observed that the confusability during T2W transduction gets enhanced for homophones. Referring to Example 2 in Table 4.4, it can be observed that, for a proper noun having the hypothesized target sequence "h i n dx ii", the T2W transduction process can yield either "hindi" or "हिंदी" as the word output. A high frequency of such errors ends up degrading the WER.

In the context of hybrid ASR task involving Hindi-English code-switching, the authors in [130] faced a similar challenge and explored merging of some identical sounding words based on the unigram counts in their database. It is argued that, by following such an approach, we can handle only a sub-set of homophones (primarily proper nouns, abbreviations) but not the set comprising intra- and inter-language homophones. For effective handling of the latter set of homophones, we would require more in-depth context information rather than unigram counts. Therefore, we have proposed a context-dependent T2W transduction process developed by exploiting modularized decoding for addressing the earlier highlighted issues. The proposed scheme employs an explicit *error model* (EM) along with an LM to provide context information. The details of the same are presented in the following

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us pag* **Ref. target sequence:** c o m p a n y _ क े _ a b o u t _ u s _ p a g e _ म े _ ज ा न क ा र ी _ ह **Hyp. target sequence:** क ं o m p a n y _ क े _ ए b o u t _ u s _ p a g e _ म े _ ज ा न क ा र ी _ **Hyp. sentence:** <*unk*> के <*unk*> us page में जानकारी है
Example 2	**Ref. sentence:** आपको hindi में blogging शुरू करनी चाहिए (*you should start blogging in hindi*) **Ref. target sequence:** आ प क ो _ h i n d i _ म े _ b l o g g i n g _ श ु र ू _ क र न ी च ा **Hyp. target sequence:** आ प क ो _ h i n d i _ म े _ b l o g g i n g _ श ु र ू _ क र न ी _ च ा **Hyp. sentence:** आपको hindi में blogging शुरू करनी चाहिए

Example 1	**Ref. sentence:** company के about us page में जानकारी है (*information is in the company's about us pag* **Ref. target sequence:** k a m p a n ii _ k ee _ a b au tx _ a z _ p ei j _ m ee _ j aa n k aa r ii _ h ei **Hyp. target sequence:** k a m p a n ii _ k ee _ a b au tx _ a **s** _ p ei j _ m ee q _ j aa n k aa r ii _ h ei **Hyp. sentence:** company के about <*unk*> page में जानकारी है
Example 2	**Ref. sentence:** आपको hindi में blogging शुरू करनी चाहिए (*you should start blogging in hindi*) **Ref. target sequence:** aa p k o _ h i n dx ii _ m ee _ b l ao g i n g _ sh u r uu _ k a r n ii _ c aa h i ee **Hyp. target sequence:** aa p k o _ h i n dx ii _ m ee _ b l ao g i n g _ sh u r uu _ k a r n ii _ c aa h i ee **Hyp. sentence:** आपको हिंदी में blogging शुरू करनी चाहिए

4.3 Context-Dependent T2W Transduction

In hybrid ASR literature, a few works have already explored the modularized decoding for T2W transduction. Demuynck *et al.* [131,132] proposed a two-step decoding process that employed morpho-syntactic and morpho-phonologic constraints for T2W transduction. Following that work, Zweig, and Nedel [133] presented an empirical study on the error-robustness of T2W transduction across a variety of languages. For that study, the decoding objective for the transduction of i^{th} hypothesized segment is formulated as shown below.

$$\begin{aligned}
\arg\max_{W_i} \mathrm{P}(W_i|T_{h_i}) &= \arg\max_{W_i} \mathrm{P}(W_i)\,\mathrm{P}(T_{h_i}|W_i) && (4.11)\\
&= \arg\max_{W_i} \mathrm{P}(W_i) \sum_{T_{c_i}} \mathrm{P}(T_{c_i}, T_{h_i}|W_i) && (4.12)\\
&= \arg\max_{W_i} \mathrm{P}(W_i) \sum_{T_{c_i}} \mathrm{P}(T_{h_i}|T_{c_i}, W_i)\,\mathrm{P}(T_{c_i}|W_i) && (4.13)\\
&\quad \arg\max_{W_i, T_{c_i}} \mathrm{P}(W_i)\ \mathrm{P}(T_{c_i}|W_i)\ \mathrm{P}(T_{h_i}|T_{c_i}) && (4.14)
\end{aligned}$$

In the above formulation, the first and second factors respectively denote the LM and the PM, while the third factor accounts for the EM. With the maximization performed over all possible correct target sequences T_{c_i} and their corresponding words W_i, the earlier discussed *hunki* and homophone issues can be resolved. Exploiting these observations, a novel T2W transduction scheme for E2E ASR systems has been evolved and is explained below.

The hypothesized target sequences produced by an E2E ASR system may contain one or more errors. For the error modeling purpose, we have employed the Levenshtein (edit) distance-based search. Let $\{T_{h_i}\}_{i=1}^{n}$ denote a hypothesized target sequence having n segments. For each segment T_{h_i}, we have determined all possible (say p) pronunciation sequences $\{T_{c_j}\}_{j=1}^{p}$ in PM having edit distances up to a predetermined threshold [1]. For the reduced target set case, those sequences may further map to homophones within or across the languages. Let a set $\{W_{i_k}\}_{k=1}^{m}$ denote all possible (say m) words returned by that search. On appending each of those words to the current partial sentence S, a corresponding new partial candidate sentence is constructed. All those constructed sentences are now pruned based on the context information derived from an appropriate LM and

[1] In this work, the threshold is set as zero when the minimum edit distance value of the matches is zero; otherwise, it is set as one more than the minimum edit distance value.

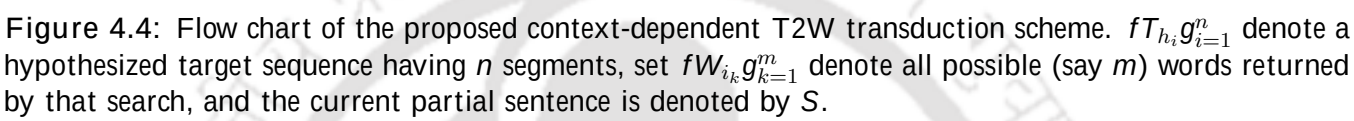


Figure 4.4: Flow chart of the proposed context-dependent T2W transduction scheme. $fT_{h_i}g_{i=1}^{n}$ denote a hypothesized target sequence having n segments, set $fW_{i_k}g_{k=1}^{m}$ denote all possible (say m) words returned by that search, and the current partial sentence is denoted by S.

the 1-best sentence is generated. When all segments in T_h get processed, the 1-best output yields the final transducted output. The overall flow diagram of the proposed T2W transduction scheme is shown in Figure 4.4.

The innovation in the proposed scheme is demonstrated with the help of an example shown in Table 4.7. The top-two rows of that table show the word- and target-level reference transcriptions for an example utterance. The output generated by the reduced target set-based E2E ASR system for that utterance is given in the third row. Whereas, the last-two rows correspond to the outputs produced by the naive and the proposed transduction schemes. On comparison, it can be noted that the proposed scheme not only avoids *hunki* labels but also can handle intra- and inter-language homophone pairs such as "light–lite" and "co–को", respectively. The first attribute refers to effective error modeling, while the second one is the result of context modeling.

Hindi-English	meeting	का	outcome	क्य
POS tag	Noun	Prep	Noun	Que

The LM scores of the candidate sentences clearly show how effectively the homophone issue gets resolved. Thus, LM plays a vital role in the proposed T2W transduction scheme.

Hindi-English sentence: meeting का outcome क्या था
English translated version: what is the outcome of the meeting

CSI tagged output:
W-meeting:S-मुलाकात:C-Yes W-का:S-का:C-No W-outcome:S-परि
W-क्या:S-क्या:C-No W-था:S-था:C-No

4.4 Experimental Setup

This section describes the creation of a database used for the evaluation of the proposed approaches. Later, the parameter tuning experiments of different types of E2E ASR systems developed

Hindi-English code-switching sentences	meeting का outcome क्या था rajadhani express के बारे में information कैसे प्रा class और object के बीच relationship क्या है
English translations	what is the outcome of the meeting how to get information about rajadhani express

	प फ ब भ म य र ल व श ष स ह क़ ख़ ग़ ज़ झ ड़ ढ़ फ़ ॉ ॉ ॊ ॊ ॊ ो ौ ् ॎ ॏ ॐ ॑ ॒ ॓ ॔ ॕ
English characters	a b c d e f g h i j k l m n o p q r s t u v w x y z

Proper nouns	**Collective/ abstract nouns**	**Abbreviations**	**Intra-language**	**Inter-language**
amit - अमित hindi - हिंदी japan - जापान	internet - इंटरनेट tieket - टिकट station - स्टेशन	ATM - एटीएम USA - यूएसए CEO - सीईओ	sea-see due - dew dye - die	fool - फूल bus - बस say - से

Table 4.7: Demonstration of T2W transduction achieved by the naive and the proposed schemes for the reduced target set case. The hypothesized target segments and words in error are shown in the `red' colour. For the proposed scheme, the highlighted LM score corresponds to the 1-best output.

Ref. sentence	क्या आपने google web light से अपने stats में traffic को notice किया
Ref. target sequence	k y aa _ aa p n ee _ g uu g a l _ w ae b _ l ii tx _ s ee _ a p n ee _ s tx ae tx s _ m ee _ tx r ae f i k _ k o _ n o tx i s _ k i y aa
Hyp. target sequence	k y aa _ aa p n ee _ g uu g a l _ w ae b _ l i tx _ s ee _ a p n ee _ s tx ei tx s _ m ee _ tx r ae f i k _ k o _ n o tx i s _ k i y aa
Naive T2W transduction	क्या आपने google web <*unk*> से अपने <*unk*> में traffic co notice किया
Proposed T2W transduction (Candidate sentences with LM scores)	-32.91 क्या आपने google web light से अपने stats में traffic co notice किया **-27.91** क्या आपने google web light से अपने stats में traffic को notice किया -32.33 क्या आपने google web light से अपने status में traffic co notice किया -28.58 क्या आपने google web light से अपने status में traffic को notice किया -35.79 क्या आपने google web lite से अपने stats में traffic co notice किया -32.51 क्या आपने google web lite से अपने stats में traffic को notice किया -37.10 क्या आपने google web lite से अपने status में traffic co notice किया -33.22 क्या आपने google web lite से अपने status में traffic को notice किया

is also described. Following that, the details of the LM employed in the proposed approach are provided.

4.4.1 Database Preparation

In this work, the *HingCoS Corpus* discussed in Chapter 3 has been used for experimentation purposes. The text data in the HingCoS corpus consists of 25988 Hindi-English code-switching sentences and has a vocabulary of 14643 words (6029 Hindi and 8614 English). The lengths of sentences vary from 3–57 words, and on an average, there are 3–4 code-switching instances per sentence. For a total of 9251 Hindi-English text sentences in the HingCoS corpus, the corresponding speech data spoken by 101 speakers (61 males and 40 females) is also available. The speech data, being collected over telephones, is sampled at 8 kHz with a resolution of 16 bits/sample. The total size of the speech data is about 25 hours. The salient statistics of the HingCoS corpus are summarized in Table 4.8.

Though we are not intended to contrast the baseline hybrid ASR systems with the E2E ASR systems, the training, development, and testing sets for both the setups are kept same as defined in Section 3.6.1, just to have a fair comparison among the developed systems. The training, develop-

Table 4.8: Salient statistics of text and speech components of the HingCoS corpus. The CS count refers to the number of code-switching instances in the data.

Type	# sent.	# words		# unique words		CS count
		Hindi	English	Hindi	English	
Text	25,988	381,603	196,556	6,029	8,614	104,912
Speech	9,251	125,653	50,719	2,644	3,901	30,035

ment, and testing sets are non-overlapping sets having 7115, 160, and 1976 utterances, respectively. These sets are also non-overlapping in terms of the speakers involved. For language modeling, excluding the earlier defined acoustic test set, the remaining text data is partitioned into training and development sets having 22700 and 1312 sentences, respectively. In this way, both acoustic and language models are evaluated on the same test set.

4.4.2 System Description and Parameter Tuning

The Nabu toolkit [35] is used for developing both the attention- and CTC-based E2E ASR systems. The parameter settings used for analyzing the speech data include window length of 25 ms, window shift of 10 ms, and pre-emphasis factor of 0.97. The 40-dimensional log Mel-filterbank energies per speech frame are used as features for acoustic modeling.

4.4.2.1 Attention-based E2E ASR system

The details of the LAS architecture used for developing an attention-based E2E ASR system are as follows. The listener has 3 pyramidal BLSTM layers, with 256 units in each layer. The pyramidal step size is kept as 2, and the dropout rate in training is set to 0.5. The speller has 2 LSTM layers, with 256 units in the input layer. The dropout rate for the speller is also set to 0.5. The average cross-entropy loss is used as a loss function. The model is trained for 300 epochs with a batch size of 32 and a learning rate set to 0.1 with a decay 0.01. For decoding, a beam-search decoder with beam width set to 10 is employed.

For training the LAS network, the number of epochs and the number of hidden units in the input layer of the encoder are selected by performing the tuning experiments on the acoustic development set. The tuning of the LAS network is done for the combined target set case, and the same parameters are fixed even for the reduced target set case. The plots showing the trends of those

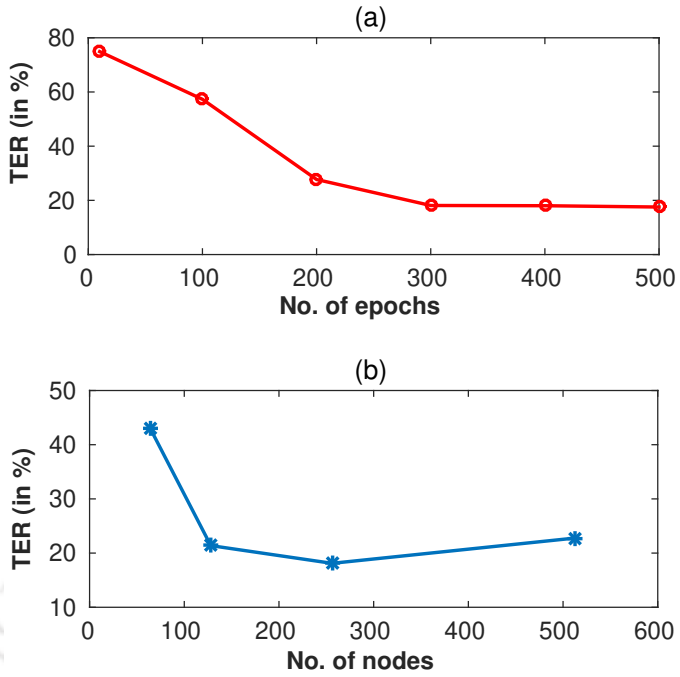


Figure 4.5: The tuning of parameters for attention-based E2E ASR system. (a) selection of number of epochs, and (b) selection of number of nodes in the encoder. The TER saturates after 300 epochs, while it degrades beyond 256 nodes.

experiments are given in Figure 4.5. Note that tuning for the number of epochs is done by keeping the number of nodes as fixed and vice-versa. The remaining parameters are set to their default values as defined in the toolkit. The TER is found to saturate after 300 epochs, while it degrades beyond 256 nodes.

4.4.2.2 CTC-based E2E ASR system

In our implementation of the CTC-based system, the DBLSTM encoder network consists of 4 layers and 256 units in each layer. The remaining network training parameters are kept the same as that of the attention-based system. The CTC-based system is trained and evaluated on identical data partitions, as already described in Section 4.4.1.

4.4.2.3 RNN-based LMs

For the experimentation purpose, both simple and factor modeling-based RNNLMs are developed using the RNNLM toolkit [121]. Both kinds of RNNLMs are developed employing identical network architecture with *sigmoid* as the non-linearity function. After performing the tuning of

Table 4.9: Quality assessment of T2W transduction in terms of %WERs along with 95% con dence interval, obtained for attention- and CTC-based E2E ASR systems developed using reduced and combined target sets. In Naive method, neither an error model (EM) nor a language model (LM) is involved.

E2E System	**Transduction method**	**EM / LM**	**%WER (95% confidence interval)**	
			Reduced	**Combined**
Attention	Naive	No / No	40.2 (0.49)	33.9 (0.48)
	Proposed	Yes / RNN	**31.1 (0.47)**	32.3 (0.47)
	Contrast	Yes /Unigram	33.0 (0.47)	33.2 (0.48)
		Yes / No	36.8 (0.49)	33.6 (0.48)
CTC	Naive	No / No	42.2 (0.50)	41.1 (0.50)
	Proposed	Yes / RNN	**35.0 (0.48)**	38.0 (0.49)
	Contrast	Yes /Unigram	37.9 (0.49)	39.2 (0.49)
		Yes / No	39.4 (0.49)	40.7 (0.50)

simple RNNLM on the linguistic development set, the salient parameters of the architecture are a hidden layer with 200 nodes, the value of back-propagation through time variable set to 5, and the number of classes set to 100.

4.5 Experimental Results

In this section, we present the evaluation of the proposed context-dependent T2W transduction scheme in the context of the Hindi-English code-switching ASR task. The evaluation has been done on both attention- and CTC-based E2E architectures.

4.5.1 Evaluation of the T2W Transduction

For the primary proposal in this work, i.e., the reduced target set for the E2E ASR system, the results are already discussed in Section 4.2. From the experiments done on the HingCoS corpus, it can be deduced that the proposed reduced target set modeling yields about 17% relative improvement in TER in contrast to the combined target set case. Towards addressing the challenges in T2W transduction with the reduced target set modeling, we have also proposed a context-dependent T2W transduction scheme as the secondary contribution. Table 4.9 presents the detailed evaluation of the same while studying the impact of both the error model and the inclusion of context information. For attention-based E2E ASR framework, the proposed transduction scheme yields a WER of 31.09%, which happens to be 22.6% relative improvement over that of the naive transduction

scheme. Further, to study the impact of context information on the transduction performance, we also evaluated the proposed scheme with unigram LM and with no LM. The latter case refers to randomly choosing one among the word possibilities available during the error modeling. From those results, we can conclude that most of the improvement in the T2W transduction performance has been achieved on account of better context modeling. For comparison purposes, the results for the combined target set modeling case are also given in Table 4.9. Unlike the reduced target set modeling, the homophone issue does not crop up in the combined target set case, despite that the reduced target set modeling results in the best WER.

For a thorough evaluation, the proposed approaches are also evaluated in the CTC-based E2E ASR framework. On comparing with the corresponding performances of the attention-based system, it can be noted that the CTC-based E2E framework has exhibited similar performance trends.

As argued earlier and also obvious from Table 4.9, for the reduced target set case, the T2W transduction performance is highly dependent on the quality of the context information provided by the LM. The effective language modeling of code-switching text data is a research problem in itself. Towards that end, we propose a novel textual feature that helps the LM to identify the possible code-switching locations. The proposed textual feature aids LM for efficient handling of the code-switching phenomenon. In Chapter 5, we present a detailed discussion on the proposed textual feature.

4.5.2 Computational Complexity

All systems are developed on a HP-Z440 workstation. The memory requirement and the computational complexity for different systems along with the key specifications of the said workstation, are given in Table 4.10. From that table, it can be noted that, the reduced target set based E2E ASR system training takes much lesser memory and computational time when compared to the combined target set case. The reduction in the memory and computational time for reduced target set when compared to the combined target set is attributed to 34% relative reduction in the target labels that need to be modeled in the former case.

Table 4.10: The details of the memory usage and the computational time for training the E2E ASR systems using both the reduced and combined target sets. Note that, the reduced target set takes less memory and computational time when compared to the combined target set. This behaviour is attributed to the 34% relative reduction in the target labels that need to be modeled in the former case.

E2E system	Target set	GPU usage (in MB)	RAM usage (in GB)	Avg. minibatch time (in sec.)
Attention	Reduced	1412	4.66	2.36
	Combined	1832	12.60	3.41
CTC	Reduced	1223	4.63	1.57
	Combined	1791	12.50	2.92

*y*CPU: Intel® Xeon®, 64-bit, @3.60GHz 12; RAM: 128 GB, DDR4;
GPU: GeForce GTX 1060, 6 GB.

4.6 Conclusions

In this Chapter we have explored the development of code-switching E2E ASR system on limited resources. For efficient modeling of the code-switching E2E ASR system, the acoustic similarity-based target reduction scheme has been proposed. Interestingly, the reduced target set based E2E ASR system outperformed the combined target set one in terms of TER. But, a reverse trend was noted when those target sequences were converted to word sequences, i.e., for computing the WER. This degradation in WER is because of the enhanced confusability among the homophones within or across the languages involved. For addressing the same, a context-dependent T2W transduction scheme that employs an explicit EM along with an LM is proposed. The proposed approach is noted to consistently outperform the combined target set based E2E ASR modeling in terms of target/word error rate. For thorough evaluation, the proposed approaches are evaluated on both attention- and CTC-based Hindi-English code-switching E2E ASR systems. To the best of authors' knowledge, for the Hindi-English code-switching task, the E2E system is yet to be reported. Also, it is to highlight that the reduced target set based E2E ASR systems training takes much lesser memory and computational time when compared to the combined target set case.

We also note that the performance of the proposed context-dependent T2W transduction scheme is highly dependent on the quality of the context information provided by the LM. Towards that end, we propose a novel textual feature that aids LM to efficiently handle the code-switching phenomenon. The details of the proposed textual feature are presented in Chapter 5.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

5

Language Modeling of Code-Switching Text Data

Contents

5.1 Review of Factored Language Model 71
5.2 Improved POS Textual Features 74
5.3 Proposed Code-Switching Location Textual Features 79
5.4 Experimental Setup 85
5.5 Results and Discussion 88
5.6 Revalidation on Mandarin-English Code-Switching Data 92
5.7 Assessment of the Proposed Textual Features in T2W Transduction 96
5.8 Conclusion 98

In the earlier chapter, we have discussed the proposed context-dependent T2W transduction scheme. Also, we note that, the proposed T2W transduction is highly dependent on the quality of the context information provided by the LM. Towards that end, in this chapter, we mainly focus on improving the context modeling in LM for better T2W transduction. In general, the training of an LM for code-switching data can be done by any one of the following approaches: (i) tediously collect a large amount of code-switching text corpus as attempted in [72, 134], and (ii) augmenting an existing monolingual LM to handle the code-switching data as attempted in [135, 136]. In this thesis, we explore the language modeling of Hindi-English code-switching data by following both the said approaches.

Hindi	रानी	हमारे	सी ई ओ	की	इकलौती	बेटी	है
Hinglish	Rani	हमारे	CEO	की	only	daughter	है
POS	Noun	Pronoun	Noun	Prep	Pronoun	Noun	Verb
LID	Eng	Hnd	Eng	Hnd	Eng	Eng	Hnd
CSL	No	No	No	No	Yes	Yes	No

Table 5.1: An example Hindi sentence along with a few of its Hindi-English code-switched variants for highlighting the increased word sequence variability due to code-switching.

Hindi	वर्ग और वस्तु के बीच रिश्ता क्या है?
Acceptable code-switching	Class और object का बीच relationship क्या है? वर्ग और वस्तु के बीच relationship क्या है?
Odd code-switching	Class और object के between रिश्ता what है? वर्ग और वस्तु के बीच relationship what है?

क्या आप मुझे गरीबरथ द्रुतगामी का आगमन समय बता सकते है?
क्या आप मुझे Garibrath express का arrival time बता सकते है?
क्या आप मुझे Garibrath express का आने का वक्त बता सकते है?
क्या आप मुझे गरीबरथ द्रुतगामी का arrival time बता सकते है?

The traditional LMs are employed to predict the probability of occurrence of a given sequence of words from the training data. These LMs capture only the syntax information and ignores the semantic information present in the data while training. For the code-switching case, those LMs suffer from a lack of generalizability between training and testing contexts. To explain that, consider a root Hindi sentence and a few of its Hindi-English code-switched variants, as shown in Table 5.1. Note that these sentences are similar in orthography and semantics but involve different word sequences. Thus, despite having seen any one of the sentences during training, the LM fails to predict the other three variants effectively during testing. Also, the code-switching phenomenon cannot be characterized as a random mixing of words or phrases from two or more languages [62]. In fact, the bilingual code-switching phenomena have been noted to follow some broad syntactic rules [63, 64]. For demonstrating that fact, let us consider a few acceptable as well as odd Hindi-English code-switching versions of an example Hindi sentence as listed in Table 5.2. Thus, it is hypothesized that the language modeling of code-switching data gets more effective with the inclusion of syntactical information embedded in the given sentences. Towards that end, in this

Hinglish	Garibrath	express	का	arrival	time	क्या	है
POS	Noun	Adj	Prep	Adj	Noun	Ques	
LID	Eng	Eng	Hnd	Eng	Eng	Hnd	Hnd
CSL	No	Yes	No	Yes	Yes	No	No

Intra-sentential	Type-1	भारत में *popular free virtual credit card services* कितनी हैं how many popular free virtual credit card services are in india
		अपने *budget* के अनुसार *investments* कर सकते हैं you can invest as per your budget
	Type-2	*class* और *object* के बीच *relationship* क्या है what is the relationship between class and object

Table 5.2: Example sentences highlighting the broad syntactic rules being followed in Hindi-English code-switching. For the given example Hindi sentence, the English translation is presented below it.

Hindi sentence	वर्ग और वस्तु के बीच रिश्ता क्या है
Translated English	*what is the relationship between class and object*
Acceptable code-switching	class और object के बीच relationship क्या है वर्ग और वस्तु के बीच relationship क्या है
Odd code-switching	class और object के between रिश्ता what है वर्ग और वस्तु के बीच relationship what है

thesis, we have proposed a few textual features for better context modeling of the code-switching data. For incorporating the such context information into language modeling, the factored language model (FLM) is employed. In the following, we present a brief review on FLM along with a detailed description of the proposed textual features.

5.1 Review of Factored Language Model

FLM happens to incorporate morphological and linguistic information while training an LM [137]. In this method, each word w_t in the vocabulary V is denoted by a group of K features as

$$w_t \quad [f_t^1, f_t^2, \ldots, f_t^K] = f_t^{1:K} = \mathrm{F}_t. \tag{5.1}$$

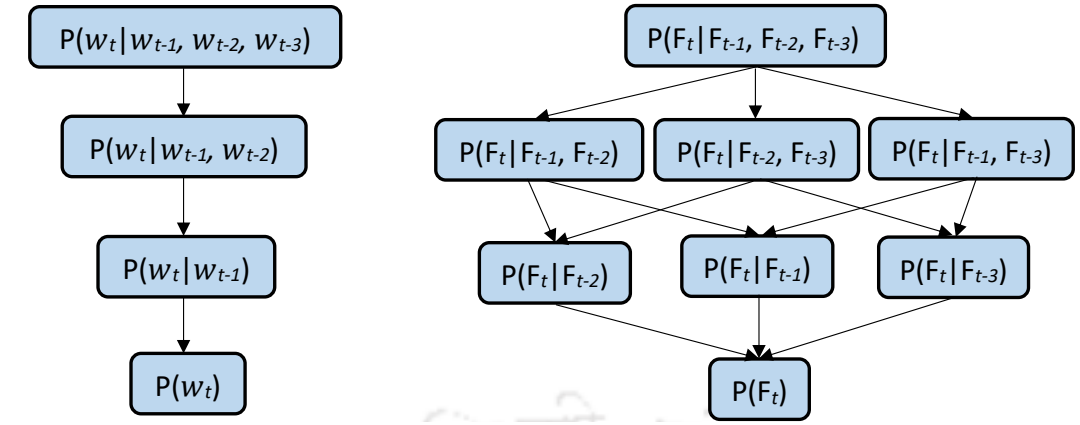
These features include morphological features (like roots, stems, etc.) or any other linguistic features of the word w_t. The probabilistic representation of an FLM, over a sentence of T words and each word having K features, is given as

$$\begin{aligned} \mathrm{P}(w_1, w_2, \ldots, w_T) &= \mathrm{P}(\mathrm{F}_1, \mathrm{F}_2, \ldots, \mathrm{F}_T) \\ &= \mathrm{P}(\mathrm{F}_{1:T}) \end{aligned} \tag{5.2}$$

In this work, we have employed both factored n-gram and factored RNNLM to capture the proposed textual features in modeling.

5.1.1 Factored n-gram LM

Similar to the traditional n-gram LM, the factored n-gram LM also uses the Markov independence assumption. Given the history of (n 1) words in the sentence, the probability of the next



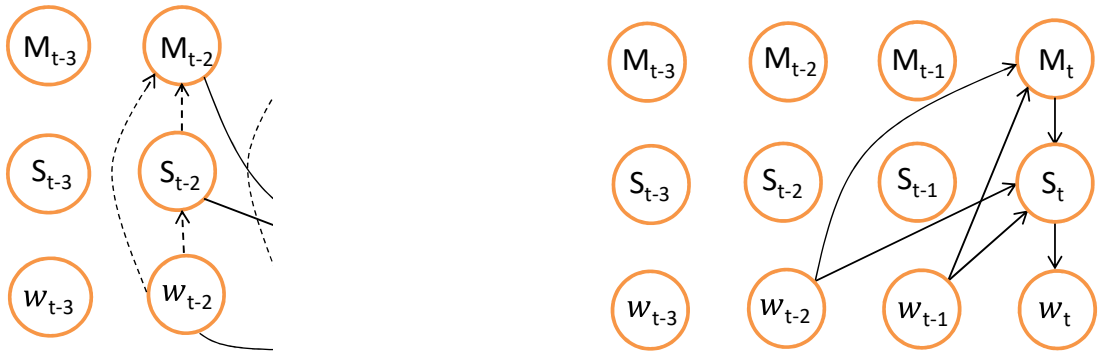


Figure 5.2: Two example topologies among many possible back-o options that can be employed in training the FLMs. Here each word w_t has two features represented by M_t and S_t, respectively.

word w_t can similarly be written as

$$\begin{aligned} \mathrm{P}(\mathrm{F}_{1:T}) &= \prod_{t=1}^{T} \mathrm{P}\left(\mathrm{F}_t | \mathrm{F}_{t-(n-1):t-1}\right) \\ &= \prod_{t=1}^{T} \prod_{k=1}^{K} \mathrm{P}\left(f_t^k | f_t^{1:k-1}, \mathrm{F}_{t-(n-1):t-1}\right) \end{aligned} \tag{5.3}$$

In n-gram LM, back-off techniques are employed to handle the unseen word sequences during testing [45, 138]. In Figure 5.1, the possible back-off paths for the traditional 4-gram LM and the factored 4-gram LM are shown for comparison purposes. We note that the factored n-gram LM has a wide range of back-off modeling options in addition to those permitted by the traditional n-gram models. The modeling of the optimal FLM involves two steps: (i) defining an appropriate set of

features, and (ii) finding the FLM with the best back-off model over these features. To find the optimal FLM, a generalized parallel back-off technique has been used in its training. When the FLM encounters an unseen word sequence, it estimates the probability based on the result obtained from several different back-off paths. The FLM can combine all the estimates derived from these back-off choices or pick the most reliable back-off estimate. In either case, the back-off paths are dynamically chosen based on the current values of the variables. This procedure is called generalized parallel back-off. For a more detailed explanation, the reader is referred to [139]. Two example topologies among many possible back-off options that can happen with the FLM are shown in Figure 5.2. In this approach, a series of LMs are consulted in sequence, and the model that can reliably estimate the probability of an unseen event in development data is chosen for evaluation [140].

5.1.2 Factored RNNLM

The RNNLMs are already reported to be more efficient in modeling long-term dependencies and semantic information than the n-gram LMs [141–144]. In recent works, the RNNs are also used in training FLMs and have resulted in significantly improved recognition performances [145, 146]. Motivated by that, we have also employed the factor modeling in RNNLM and the same is referred to as fRNNLM in this thesis. The network architecture of fRNNLM while highlighting the component variables is shown in Figure 5.3. The fRNNLM predicts the posterior probability of the current word w_t as

$$\begin{aligned} \mathrm{P}(w_t | \mathrm{F}_{t-1}, s_{t-1}) &= \sum_{c_t} \mathrm{P}(w_t | F_{t-1}, s_{t-1}, c_t)\, \mathrm{P}(c_t | F_{t-1}, s_{t-1}) \\ &\arg\max_{c_t} \mathrm{P}(w_t | F_{t-1}, s_{t-1}, c_t)\, \mathrm{P}(c_t | F_{t-1}, s_{t-1}) \end{aligned} \tag{5.4}$$

where F_{t-1} denotes the features corresponding to w_{t-1}, i.e., the previous word, s_{t-1} refers to the previous context [147] or RNN state [148], and c_{w_t} represents the class to which the word w_t belongs to [123]. These classes are derived by partitioning the vocabulary of training data into groups based on the word counts and this helps in reducing the search complexity. In fRNNLM training, we learn the different 1-hot embeddings for each factor of the word separately and then represent that respective word by concatenating the learned embeddings of those factors [149]. This concatenated

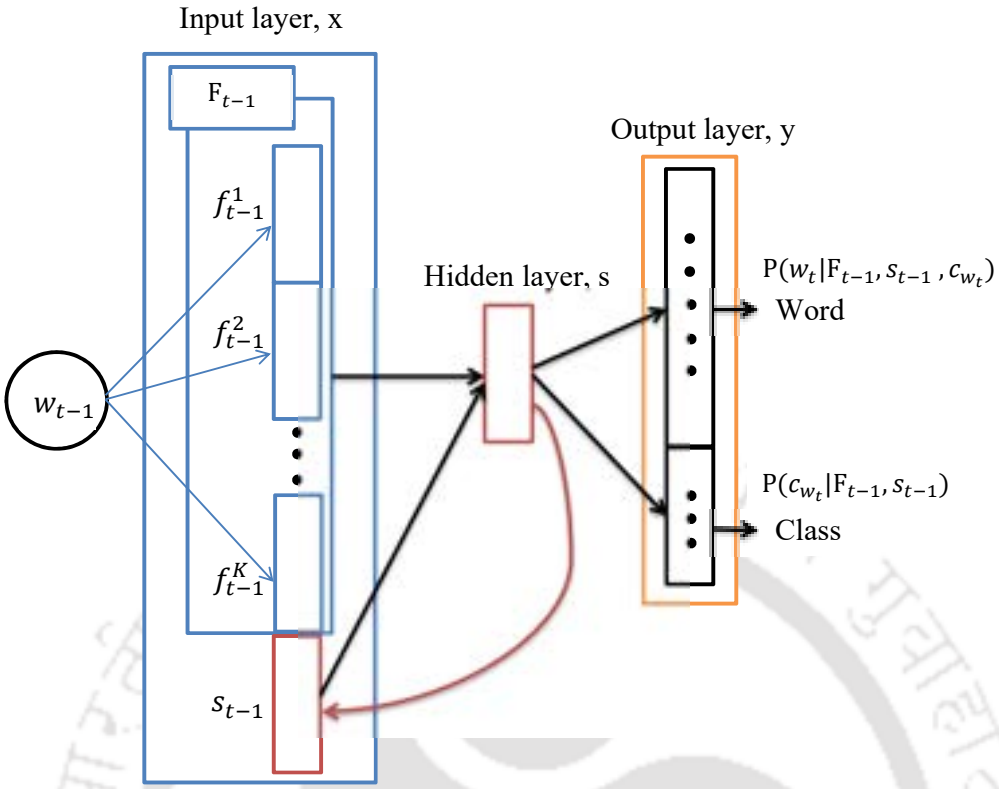


Figure 5.3: Architecture of the factored RNNLM. Where *x*, *s*, and *y* denote the input, the hidden and the output layers, respectively.

feature vector is used for training the fRNNLM to predict the next word in the sequence.

For training the FLMs, the appropriate set of features are derived either using linguistic knowledge or using data-driven techniques. Some of these features are discussed in the following sections. The created FLMs are then evaluated on code-switching test data under two conditions based on the availability of code-switching training data. Thus, the code-switching test data has been evaluated on the FLMs trained on (i) similar code-switching data (when sufficient amount of such data is available for training purpose), and (ii) monolingual native language data (when limited amount of code-switching data is available for adaptation/augmentation purpose).

5.2 Improved POS Textual Features

In natural language processing, the parts-of-speech of a word in a given sentence is derived based on the semantic information present in that sentence. The choice of the POS features of a word is decided by its neighbouring words. Therefore, the same word can have different POS features in semantically different sentences. On the combined modeling of the words with their POS features,

the trained LM is expected to model the semantic information more effectively. In the following, we describe how the POS features can be introduced to handle the Hindi-English code-switching data in two different conditions: (i) incorporating POS features directly into code-switching LM, and (ii) incorporating POS features into monolingual LM to handle the code-switching data.

5.2.1 Incorporating POS Features in Code-Switching LM

Barring a few dominant Indian languages, the POS taggers for most of them are yet to be developed due to lack of linguistic resources such as text corpora, morphological analyzers, lexicons, etc. In [71], the authors proposed an approach for extracting the POS features of the Spanish-English code-switching sentences through separate monolingual Spanish and English POS taggers. Later, based on the language of the words in the code-switching sentences, the POS features are derived from the output of that respective monolingual POS tagger. But, in the case of intra-sentential code-switching, the majority of the non-native language words have almost nil or insufficient context information associated with them. As a result of that, improper POS tagging of the non-native words is produced by the corresponding monolingual POS tagger. For addressing this issue, we describe a novel strategy for achieving more effective POS tagging of sparsely embedded non-native words in the code-switching data. Though presented in the context of Hindi-English code-switching, the proposed approach can also be applied to other code-switching contexts. The key innovation of the proposed POS tagging strategy lies in translating the Hindi-English code-switching training data into pure English sentences with the help of any functional Hindi-to-English machine translation (MT) tool such as *Google Translate*, *Bing Translator*, *Yandex Translate*, etc. A recent study reports that some of these translators can handle code-switching data apart from pure languages [87].

The role of MT is limited to capture the broad contextual information for English words embedded in the Hindi-English sentence. For the efficacy of the proposed strategy, a highly accurate translation is not mandatory. The obtained English sentences are then passed through the monolingual English POS tagger [150] to extract the POS features of the English words in the Hindi-English code-switching data. In this way, the context information of the English words can be preserved, which helps in extracting the POS features accurately. Whereas, the POS features of native Hindi words in the Hindi-English code-switching data are extracted by following the existing approach [71],

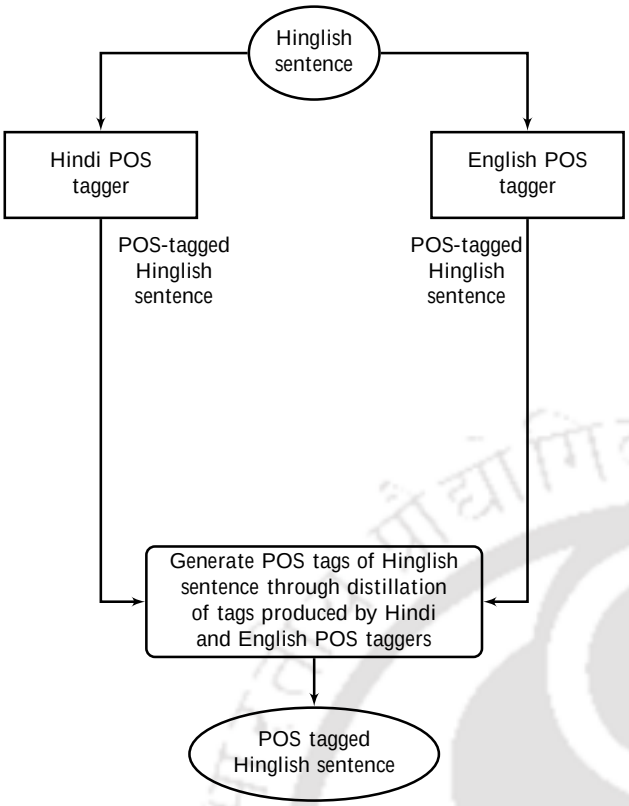


Figure 5.4: (a) Flow chart of the existing Hindi-English POS tagger. (b) Flow chart of the proposed approach. The key innovation in the proposed approach has been highlighted, which enables us to extract more accurate POS features of English words embedded in Hindi-English data. Note, for the English words not present in the translated version, the POS tags are derived using the existing approach shown in (a).

i.e., by employing an open-access monolingual Hindi POS tagger[1]. The Hindi POS tagger uses the AnnCorra POS tagset as defined in [151]. Whereas, the English POS tagger uses the Penn Treebank POS tagset as defined in [152]. These two POS taggers have different tags for similar POS information. Hence, they are mapped to a unified POS tagset containing 27 labels, and the same has been considered to extract the final POS tags for Hindi-English code-switching data.

In cases where the MT fails to produce the English words that appear in the original Hindi-English sentence, the POS tags are derived using the existing approach. i.e., the POS tags for English words are extracted by passing the Hindi-English sentences to the English POS tagger. However, in our experimentation, it is observed that 96 % of English words in Hindi-English sentences are retained in their English translated versions by Google Translate. The algorithm for the proposed POS tagging strategy is given as a flow chart in Figure 5.4(b). Table 5.3 shows the POS features

[1]Hindi POS tagger. [Online] http://sivareddy.in/downloads. Accessed: 2017-09-30.

common men आधिक पीड़ित है
mango मेरा favourite fruit है

आम आदमी आधिक पीड़ित है
आम मेरा पसंदीदा फल है

English words	result	tension	meeting	outcome	only	daughter	common	men	mango	favourite
Hindi words	परिणाम	चिंता	बैठक	परिणाम	इकलौती	बेटी	आम	आदमी	आम	पसंदीदा
POS tag	Noun	Noun	Noun	Noun	Adj.	Noun	Adj.	Noun	Noun	Adj.

Table 5.2: Improved POS Textual Features

Hinglish sentence	रानी	हमारी	वर्तमान	है
Hindi sentence	रानी	हमारी	वर्तमान	है
LID tag	Hnd	Hnd	Eng	Hnd
Hinglish sentence	rani	hamaari	present	hai
CSL tag	No	No	No	No
LID tag	Eng	Hnd	Eng	Hnd
CSL tag	No	No	Yes	No

Table 5.3: Incorporation of POS information in the modeling of Hindi-English sentences. (a) The existing approach, i.e., employing English POS tagger to Hindi-English words in Hindi-English sentences while using the Hindi POS tagger for the rest. On account of the lack of contextual information, the English POS tagger outputs the Noun tag for most of the words in Hindi-English sentences. (b) The proposed approach, i.e., employing English POS tagger, on converted Hindi-English training sentences into English using a machine translation system. Note that the LID tag "Hnd" refers to the list of markers and includes surrounding punctuation (as mentioned in the Treebank POS Tags).

Hinglish sentence	देश	की	currency	every	year	change	होनी	चाहिए
Hinglish sentence	देश	की	currency	every	year	change	होनी	चाहिए
English POS tagger o/p	List	Adj	Noun	Det	Noun	Noun	Verb	Adverb
Hindi POS tagger o/p	Noun	Prep	Adj	Noun	Adj	Noun	Verb	Aux verb
Distilled o/p	Noun	Prep	Noun	Det	Noun	Noun	Verb	Aux verb
English POS tagger o/p	List	Adj	Noun	Det	Noun	Noun	Verb	Adverb
Distilled o/p	Noun	Prep	Noun	Det	Noun	Noun	Verb	Aux verb

(a)

Hinglish sentence	देश	की	currency	every	year	change	होनी	चाहिए
Hinglish sentence	देश	की	currency	every	year	change	होनी	चाहिए
English sentence	currency	of	the	country	should	change	every	year
Hindi POS tagger o/p	Noun	Prep	Adj	Noun	Adj	Noun	Verb	Aux verb
English sentence	currency	of	the	country	should	change	every	year
Distilled o/p	Noun	Prep	Noun	Det	Noun	Verb	Verb	Aux verb
English POS tagger o/p	Noun	Prep	Det	Noun	Modal	Verb	Det	Noun
Distilled o/p	Noun	Prep	Noun	Det	Noun	Verb	Verb	Aux verb

(b)

obtained by the existing approach and the proposed novel strategy, for example, Hindi-English sentence. From Table 5.3, we note that the proposed strategy helps to estimate contextually correct POS features for the code-switching (English) words than the existing approach.

5.2.2 Incorporating POS Features in Monolingual LM

Recent research has shown that if the low resourced language is topologically related to a rich resourced language, it is possible to develop a POS tagger for the former using the linguistic resources of the latter [153]. On studying the topological similarity between Hindi-English and their Hindi translated versions, we notice a good match. The Hindi-English sentences were translated to Hindi using *Google Translate* [2], which was provided with the correct supervision about the source and the target languages as English and Hindi, respectively. As a result of that, it has been observed that the foreign language (English) words in the code-switching sentences get translated to the corresponding native language (Hindi) words while mostly retaining the syntax. In our database, about 70 % of

[2] https://translate.google.co.in. Accessed: from August 2018 to July 2019

	Hinglish sentences	क्या आप मुझे deccan queen का departure time बता सकते है functions के नाम भी lowercase letter से ही शुरू होते हैं कृपया मुझे मेरा current account balance बताएं
Case 1	**Hindi translated versions**	क्या आप मुझे डेक्कन रानी का प्रस्थान समय बता सकते हैं कार्यों के नाम भी छोटे अक्षर से ही शुरू होते हैं कृपया मुझे मेरा चालू खाता शेष बताएं
	English translated versions	can you tell me the departure time of deccan queen the names of the functions also start with a lowercase letter please tell me my current account balance
	Hinglish sentences	advice दे रहा है कि आप इसे remove कर दीजिये यदि आपकी organization grow करती हैं तो आप भी grow करेंगे केवल कुछ crazy moments capture करने के लिए तैयार रहिये
Case 2	**Hindi translated versions**	सलाह दे रहा है कि आप इसे हटा दीजिये यदि आपकी संगठन बढ़ती हैं तो आप भी बढ़ेगे केवल कुछ पागल पल पकड़ने के लिए तैयार रहिये
	English translated versions	i advice you to remove it if your organization is growing then you will also grow just be ready to capture some crazy moments

Table 5.4: Typical Hindi-English code-switching sentences in the database and their respective Hindi and English translated versions. Case 1 and Case 2 refer to the presence and absence of one-to-one topological matching between Hindi-English sentence and the corresponding translated Hindi version, respectively.

Example Hinglish sentences	Code-switching list		Non-code-switching list		
result के बारे मे tension मत करो	result	classical	के	पीड़ित	है
meeting का outcome क्या था	tension	dancer	बारे	मेरा	
common man बहुत पीड़ित है	meeting	present	में	बहुत	
mango मेरा favourite fruit है	outcome	common	मेरा	पीड़ित	
rani एक classical dancer है	common	man	बहुत	है	
आपका present ceo कौन है	man	mango	पीड़ित		
	mango	favourite	है		
	fruit	result			

Table 5.5: An example sentence pair highlighting the fact that the POS features extracted for Hindi text remain mostly valid for Hindi-English code-switching text too. The English translation for the given Hindi sentence is mentioned in bracket.

Hindi sentence	आम	आदमी	बहुत	पीड़ित	है
Hinglish sentence	common	man	बहुत	पीड़ित	है
POS tag	Adj	Noun	Adverb	Verb	Aux. verb

Hindi sentence	परिणाम	के	बारे	में	चिंता	मत	करो
Hinglish sentence	result	के	बारे	में	tension	मत	करो
CSL tag	Yes	No	No	No	Yes	No	No
POS tag	Noun	Prep	Prep	Noun	Noun	Aux.	Verb

Hinglish sentences | **Hindi translated versions**

Hindi sentence	बैठक	का	परिणाम	क्या	था
Hinglish sentence	meeting	का	outcome	क्या	था
CSL tag	Yes	No	Yes	No	No
POS tag	Noun	Prep	Noun	Ques.	Aux. verb

Hindi sentence	आम	आदमी	बहुत	पीड़ित	है
Hinglish sentence	common	man	बहुत	पीड़ित	है
CSL tag	Yes	Yes	No	No	No
POS tag	Adj	Noun	Adverb	Verb	Aux. verb

Hindi sentence	आम	मेरा	पसंदीदा	फल	है

the Hindi-English sentences retain a one-to-one topological match with their corresponding Hindi translated versions. Whereas the remaining 30 % of the Hindi-English sentences are found to differ topologically on account of the changes in the respective grammar, mostly in verbs with respective to their Hindi translated versions. A few example Hindi-English sentences and their corresponding Hindi and English translated versions for both these cases are listed in Table 5.4

Motivated by those facts, in [135] we explored an approach where the existing monolingual LM is augmented with the code-switching information in the form of POS features. On analyzing the topological similarity between the Hindi-English sentences and their corresponding Hindi translated versions, it is observed that the POS tags of the switched English words mostly remain the same as

those of the corresponding Hindi words. This observation can also be visualized from the example given in Table 5.5. Now, before training the Hindi LM, the training data is augmented with the POS features. On testing the Hindi-English code-switching data, the monolingual Hindi LM trained with POS information is found to result in significant improvement in perplexity (PPL) over the monolingual Hindi LM trained without POS information. This improvement is because whenever the monolingual Hindi LM encounters an English word in the code-switching test utterance, it backs off to the POS information corresponding to that English word.

5.3 Proposed Code-Switching Location Textual Features

In the inter-sentential code-switching case, the non-native words are inserted into the native language sentences mostly without affecting their structure as well as semantics. To exploit this fact, we have proposed a novel textual feature that identifies the locations within a sentence where the code-switching can potentially occur. This feature is referred to as the code-switching location (*CSL*) feature in this work. Similar to the POS case, the proposed CSL features can also be introduced into the FLM training in two different conditions: (i) introducing the CSL features directly into the code-switching data while training the FLM, and (ii) introducing the CSL features into monolingual LM and adopt that LM to handle the code-switching.

5.3.1 Incorporating CSL Features in Code-Switching LM

For Hindi-English code-switching text data, the proposed CSL textual feature not only marks the location of code-switching but also provides information about the equivalent native (Hindi) word. The procedure for extracting the CSL feature for Hindi-English code-switching data is described next. In the HingCoS corpus, the Hindi and English words appear in Devanagari and Latin scripts, respectively. First, we pass every English-scripted word (w_E) through a machine translator to get its equivalent Hindi-scripted word (w_H). Later, a string *f*W-w_E:S-w_H:C-Yes*g* is emitted, which comprises the word w_E along with the CSL feature. Where the definition for identifiers W, S, and C remains the same as that of the earlier case. Similarly, for Hindi-scripted word (w_H), the string including the CSL feature, is emitted as *f*W-w_H:S-w_H:C-No*g*. Note that, as there is no code-switching, W and S are tagged with the same word w_H. The structure of the above strings follows

the syntax of the FLM toolkits [121,154]. The algorithm for the proposed CSL feature extraction is given as a flowchart in Figure 5.5. Also, an example Hindi-English code-switching sentence tagged with the proposed CSL feature is shown in Table 5.6.

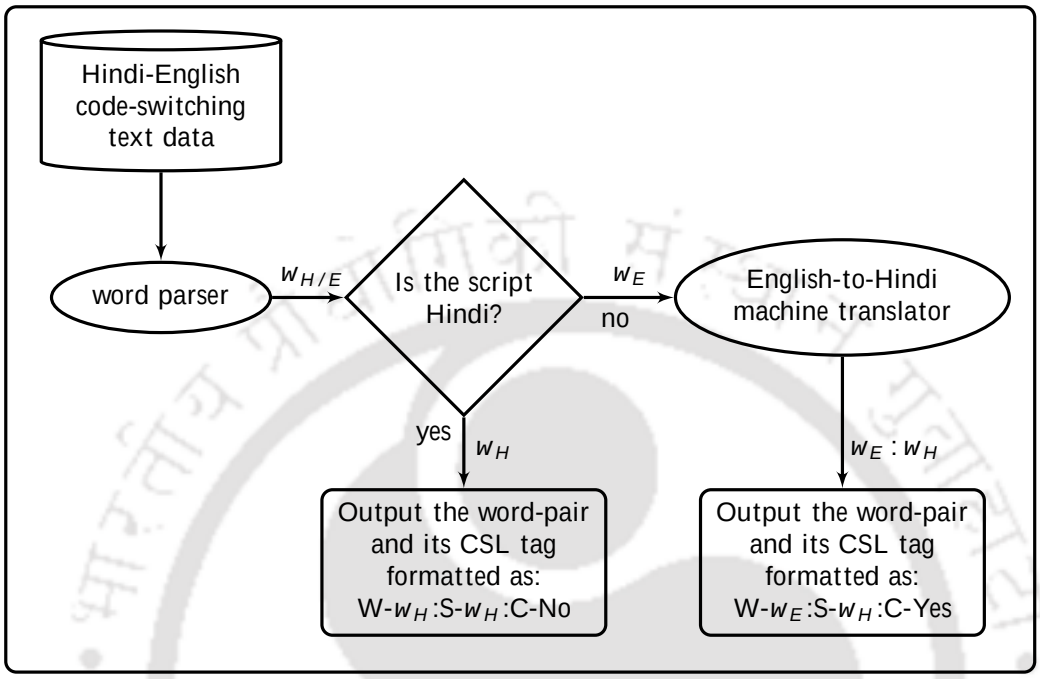


Figure 5.5: Flow chart of the scheme employed to tag the code-switching text data with the proposed CSI feature. In this work, English-to-Hindi translation is done by employing the *Google Translate*, an online machine translation tool.

Hindi-English	meeting	का	outcome	क्या	था
POS tag	Noun	Prep.	Noun	Ques	Aux. verb

Table 5.6: The proposed CSL tagged output for an example Hindi-English code-switching sentence. The English translation of example sentence is given in bracket for reference.

CSL tagged output: W-meeting:S-मुलाकात:C-Yes W-का:S-का:C-No W-outcome:S-परिणाम:C-Yes W-क्या:S-क्या:C-No W-था:S-था:C-No

Hindi-English sentence: meeting का outcome क्या था
(*what is the outcome of the meeting*)

Hindi-English code-switching sentences	meeting का outcome क्या था rajadhani express के बारे में information कैसे प्राप्त करें class और object के बीच relationship क्या है
English translations	what is the outcome of the meeting how to get information about rajadhani express what is the relationship between class and object

5.3.2 Incorporating CSL Features in Monolingual LM

For introducing the CSL features in the training of monolingual Hindi LM, a novel tagger has been developed and it is referred to as the *CSL tagger*. In the following, the steps involved in creating the CSL tagger have been described in detail.

Hindi characters	अ आ इ ई उ ऊ ऋ ए ऐ ओ औ क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ
English characters	a b c d e f g h i j k l m n o p q r s t u v w x y z

Step-1: We begin by collecting a small set of Hindi-English sentences. Each sentence of this set is then translated to Hindi using any suitable machine translator[3]. A few example Hindi-English

[3]In this work, *Google Translate* (https://translate.google.com), an online tool has been employed for translation.

Proper nouns	**Collective/ abstract nouns**	**Abbreviations**	**Intra-language**	**Inter-language**
amit - अमित hindi - हिंदी japan - जापान	internet - इंटरनेट ticket - टिकट station - स्टेशन	ATM - एटीएम USA - यूएसए CEO - सीईओ	sea-see due - dew dye - die	fool - फूल bus - बस say - से

Hindi translated versions	क्या आप मुझे डेक्कन रानी का प्रस्थान का समय बता सकते हैं कृपया मुझे मेरा चालू खाता शेष राशि बताएं कार्यों के नाम भी छोटे अक्षर से ही शुरू होते हैं मेरा एटीएम पत्रक खो गया है तो मैं अपना भुगतान को कैसे रोक सकती हूं
English translated versions	can you tell me the departure time of deccan queen please tell me my current account balance the names of functions also start with a lowercase letter my atm card is lost so how can I stop my payment

sentences along with their Hindi translations are shown below. Note that the English translated versions of the Hindi-English sentences are provided only for the reader's ease.

Hinglish sentences	Hindi translated versions	English translated versions
result के बारे मे tension मत करो	परिणाम के बारे मे चिंता मत करो	do not take tension about the result
meeting का outcome क्या था	बैठक का परिणाम क्या था	what was the outcome of the meeting
common man बहुत पीड़ित है	आम आदमी बहुत पीड़ित है	common man suffers more
mango मेरा favourite fruit है	आम मेरा पसंदीदा फल है	mango is my favourite fruit
rani एक classical dancer है	रानी एक शास्त्रीय नर्तकी है	rani is a classical dancer
आपका present ceo कौन है	आपका वर्तमान सीईओ कौन है	who is your present ceo

Step-2: As argued earlier, the sentences in intra-sentential code-switching follow the syntactic structures of the corresponding native language sentences. Therefore, we find word to word correspondence for most of the words on aligning Hindi-English sentences with their corresponding Hindi translations. In those alignments, the word-pairs that undergo code switching are labeled as 'Yes' while the remaining ones are labeled as 'No'. Although the proper nouns also undergo code-switching, they are labeled as 'No' rather than 'Yes' as this has been done since the romanized and spoken forms of those words happen to be the same across languages. For previously mentioned sentences the obtained CSL features along with the POS features derived using the earlier proposed strategy are shown below.

Hindi sentence	परिणाम	के	बारे	मे	चिंता	मत	करो
Hinglish sentence	result	के	बारे	मे	tension	मत	करो
CSL tag	Yes	No	No	No	Yes	No	No
POS tag	Noun	Prep	Prep	Prep	Noun	Neg	Verb

Hindi sentence	बैठक	का	परिणाम	क्या	था
Hinglish sentence	meeting	का	outcome	क्या	था
CSL tag	Yes	No	Yes	No	No
POS tag	Noun	Prep	Noun	Ques	Aux. verb

Hindi sentence	आम	आदमी	बहुत	पीड़ित	है
Hinglish sentence	common	man	बहुत	पीड़ित	है
CSL tag	Yes	No	No	Yes	No
POS tag	Adj	Noun	Adverb	Verb	Aux. verb

Hindi sentence	आम	मेरा	पसंदीदा	फल	है
Hinglish sentence	mango	मेरा	favourite	fruit	है
CSL tag	Yes	No	Yes	Yes	No
POS tag	Noun	Prep	Adj.	Noun	Aux. verb

Hindi sentence	रानी	एक	शास्त्रीय	नर्तकी	है
Hinglish sentence	rani	एक	classical	dancer	है
CSL tag	Yes	No	Yes	No	No
POS tag	Noun	Adj	Adj.	Noun	Aux. verb

Hindi sentence	आपका	वर्तमान	सीईओ	कौन	है
Hinglish sentence	आपका	present	ceo	कौन	है
CSL tag	No	No	Yes	No	No
POS tag	Prep	Adj	Noun	Ques	Aux. verb

5.3 Proposed Code-Switching Location Textual Features

Step-3: Based on the "Yes" CSL tag, a list is produced which records the concerned word pairs along with their corresponding POS tags. In some cases, a Hindi word can be associated with multiple non-native (English) words based on the context. For instance, the words आम (mango, common) and परिणाम (result, outcome) in the above example sentences take two different non-native (English) words. Since the POS features of words are decided based on their context, it helps in choosing the right code-switching word pairs from multiple choices. This behaviour justifies the attachment of the POS tag with the word pairs. The created list is referred to as the code-switching list (*CS-list*). For the above example sentences, the CS-list is shown below.

Hinglish sentences	Hindi translated versions
result के बारे में tension मत करो	परिणाम के बारे में चिंता मत करो
meeting का outcome क्या था	बैठक का परिणाम क्या था
common man बहुत पीड़ित है	आम आदमी बहुत पीड़ित है
mango मेरा favourite fruit है	आम मेरा पसंदीदा फल है
rani एक classical dancer है	रानी एक शास्त्रीय नर्तकी है
आपका present ceo कौन है	आपका वर्तमान सीईओ कौन है

English words	result	के	बारे	में	tension	मत	करो	meeting	का	outcome	क्या	था
Hindi words	परिणाम	चिंता	वैठक	परिणाम	आम	आदमी	आम	पसंदीदा	फल	शास्त्रीय	नर्तकी	वर्तमान
POS tag	Noun	Noun	Noun	Noun	Adj	Noun	Noun	Adj	Noun	Adj	Noun	Adj

Hinglish sentence	rani	हमारी	present	ceo	है
LID tag	Eng	Hnd	Eng	Eng	Hnd
CSL tag	POS	No	Yes	No	No

Step-4: It is assumed that a large text corpus is available for training a monolingual Hindi LM, and the word list of the same covers the obtained Hindi translations of Hindi-English development set. The training data is first POS tagged using a Hindi POS tagger. Each Hindi word is paired with its POS tag to form a string. All those strings are then searched in the CS-list. If the search succeeds, it could result in unique or multiple outcomes. For a unique outcome, the resulting string from the CS-list is modified to replace the POS tag with the "Yes" CSL tag as "*<Hindi, English, Yes>*". Alternately, we can retain the POS information by merely appending the "Yes" CSL tag to existing string as "*<Hindi, English, POS, Yes>*". When the search returns multiple outcomes, there is ambiguity about the non-native word being code-switched to, and we are only sure about the occurrence of code-switching. Thus, we tag the occurrence of code-switching by replacing the POS tag with the "Yes" CSL tag but with "Unk" (i.e. unknown) in place of the non-native word as "*<Hindi, Unk, Yes>*". In case we do wish to retain the POS information, the following string is produced as output: "*<Hindi, Unk, POS, Yes>*". If there is no matching entry in the CS-list against the search, the "No" CSL tag is attached to the searched Hindi word, and the following string is produced as output: "*<Hindi, Hindi, No>*" or "*<Hindi, Hindi, POS, No>*" to match with the above-defined order of the output strings.

Hinglish sentence	देश	की	currency	every	year	change	होनी	चाहिए
Hindi POS tagger o/p	Noun	Prep	Adj	Noun	Adj	Noun	Verb	Aux verb
English POS tagger o/p	Noun	Adp	Noun	Det	Noun	Noun	Verb	Adverb
Distilled o/p	Noun	Prep	Noun	Det	Noun	Noun	Verb	Aux verb

English sentence	currency	of	the	country	should	change	every	year
English POS tagger o/p	Noun	Prep	Det	Noun	Modal	Verb	Det	Noun
Distilled o/p	Noun	Prep	Noun	Det	Noun	Verb	Verb	Aux verb

[4] It is worth highlighting that out of 5,651 words in the Hindi translated development set only 1,640 words appear in the CS-list. This observation supports our hypothesis that code-switching does not occur randomly, and not all words in the native language are being code-switched.

For subsequent modeling, the proposed CSL tagger produces an output that suits the LM toolkits employed in this work, where W, S, P, and C refers to the input-word, semantic-word, POS tag, and code-switching status, respectively. Typical output produced for the considered Hindi example sentences is shown below. Summarizing the steps mentioned above, a complete flow chart of the CSL tagger has been developed for the ease of understanding of the reader and is shown in Figure 5.6.

Example Hindi sentences:

1. वह नायक नहीं सिर्फ एक आम आदमी था
2. इस आम को काटो और खाओ

English translated versions :

1. he was not a leader only a common man
2. cut this mango and eat

Output of the CSL tagger:

1. W-वह:S-वह:P-PRP:C-No W-नायक:S-नायक:P-NN:C-No W-नहीं:S-नहीं:P-NEG:C-No W-सिर्फ:S-सिर्फ:P-RB:C-No W-एक:S-एक:P-QC:C-No W-आम:S-common :P-JJ:C-Yes W-आदमी:S-man:P-NN:C-Yes W-था:S-था:P-VM:C-No

2. W-इस:S-इस:P-DEM:C-No W-आम:S-mango:P-NN:C-Yes W-को:S-को:P-PSP:C-No W-काटो:S-काटो:P-VM:C-No W- और:S-और:P-CC:C-No W-खाओ:S-खाओ:P-VM:C-No

Hinglish sentences	क्या आप मुझे deccan queen का departure time बता सकते है कृपया मुझे मेरा current account balance बताएं functions के नाम भी lowercase letter से ही शुरू होते हैं मेरा atm card खो गया है तो मैं अपने payment को कैसे रोक सकता हूँ
Hindi translated versions	क्या आप मुझे डेक्कन रानी का प्रस्थान का समय बता सकते हैं कृपया मुझे मेरा चालू खाता शेष राशि बताएं कार्यों के नाम भी छोटे अक्षर से ही शुरू होते हैं मेरा एटीएम पत्रक खो गया है तो मैं अपने भुगतान को कैसे रोक सकता हूँ
English translated versions	can you tell me the departure time of deccan queen please tell me my current account balance the names of the functions also start with a lowercase letter my atm card is lost so how can I stop my payment

Hinglish sentences	Hindi translated versions	English translated versions
result के बारे मे tension मत करो	परिणाम के बारे मे चिंता मत करो	do not take tension about the result
meeting का outcome क्या था	बैठक का परिणाम क्या था	what was the outcome of the meeting
common man बहुत पीड़ित है	आम आदमी बहुत पीड़ित है	common man suffers more
mango मेरा favourite fruit है	आम मेरा पसंदीदा फल है	mango is my favourite fruit
rani एक classical dancer है	रानी एक शास्त्रीय नर्तकी है	rani is a classical dancer
आपका present ceo कौन है	आपका वर्तमान सीईओ कौन है	who is your present ceo

Figure 5.6: Flow chart of a novel tagger developed for the CSL tagging of Hindi text data. This ow chart comprises two modules: code-switching (CS) word pair generation and CSL tagging. Note that, this tagger allows us to develop a monolingual Hindi LM that can handle the Hindi-English code-switching data.

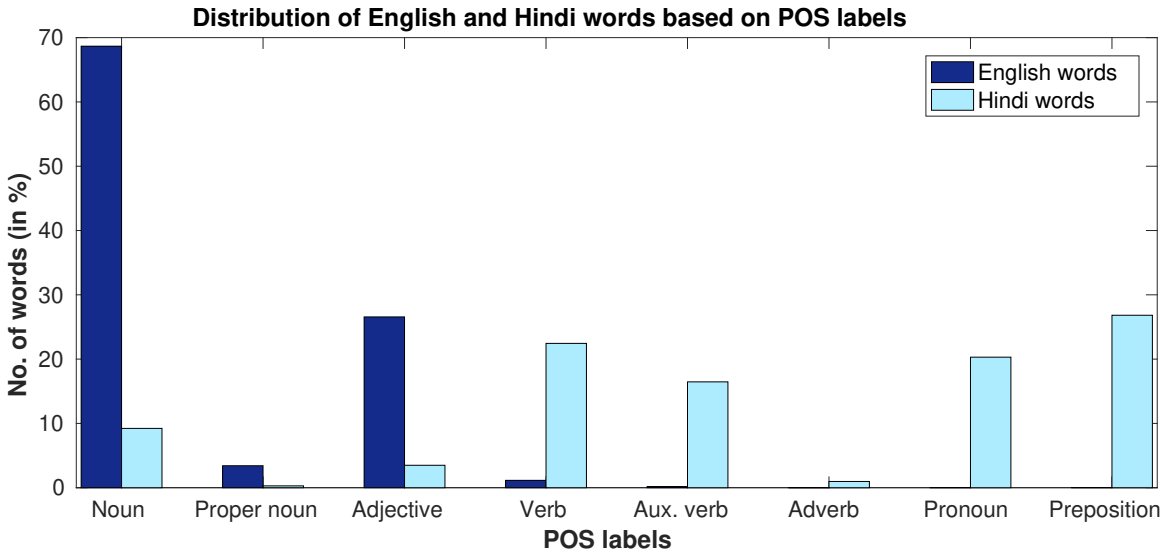


Figure 5.7: Distributions of the English and Hindi Words in the Hindi-English text corpus based on their parts of speech (POS) labels. Note that, for the ease in the display, the POS tags for Hindi and English words are normalized separately, i.e., all POS tags corresponding to English/Hindi words sum up to 100%.

5.4 Experimental Setup

This section describes the creation of datasets used for two different experimental conditions: (i) direct training of Hindi-English LM referred to as "Exp-1", and (ii) augmentation of monolingual Hindi LM to deal with Hindi-English code-switching data referred to as "Exp-2". Note that these two experiments serve entirely different purposes and therefore are independent. The parameter tuning of different types of language models developed is also described in this section.

5.4.1 Database Preparation for Exp-1

For conducting Exp-1, a subset of HingCoS text corpus, which is discussed in Chapter 3 is considered. The distributions of English and Hindi words present in the said corpus with respect to their POS labels are shown in Figure 5.7. From Figure 5.7, we note that the majority of the English words that are being code-switched are nouns, proper nouns, and adjectives. This observation supports our hypothesis that the majority of the sentences involve only the replacement of English words with their Hindi counterparts without requiring any change in the syntax of the sentences. This behaviour is because the syntax of the native sentence is mainly captured by the verbs, pronouns, prepositions, and auxiliary verbs, which are usually not being code-switched. The obtained dataset is then partitioned into training (Train-1), development (Dev-1), and evaluation

Table 5.7: The details of the vocabulary size and the word count of the training, development and evaluation datasets created for evaluating the proposed features under two experimental conditions. The Exp-1 explores training LMs on code-switching data while the Exp-2 explores augmenting monolingual LM to deal with code-switching data. Note that the Dev-1 and Eval-1 datasets are used for parameter tuning and evaluation of both Exp-1 and Exp-2 conditions

Experiment	Dataset	Data type	# sent.	# words		# unique words	
				Hindi	English	Hindi	English
Exp-1	Train-1	Hindi-English	13,071	179,798	71,143	4,980	3,649
	Dev-1		670	8,847	3,578	871	1,227
	Eval-1		1,266	16,701	6,738	1,104	1,655
Exp-2	Train-2	Hindi	100,000	1,798,255	4593	85,827	1433
	Dev-2	Hindi	1,500	15,604	1,032	1,937	377
		Hindi-English	1,500	10,483	5,937	916	1,418

(Eval-1) sets with no sentence-overlap among them. Table 5.7 summarises the salient details of the created datasets.

5.4.2 Database Preparation for Exp-2

For conducting Exp-2, the monolingual Hindi LMs are trained using a large-sized Hindi dataset extracted from an existing Hindi corpus created by IIT Bombay for machine translation purposes [155]. This set consists of 100 thousand Hindi sentences and is referred to as the "Train-2" set. In addition to that, a Dev-2 dataset has been extracted from the earlier created Hindi-English dataset through random selection. For developing the CSL tagger, first, the Dev-2 dataset comprising of 1500 Hindi-English sentences are translated to corresponding Hindi versions primarily with the help of *Google translate*, an online translation tool. In some cases, Google translate produced incorrect Hindi translations, possibly due to its inadequate Hindi vocabulary. In such cases, the Hindi translations are corrected manually with the help of a few online Hindi vocabulary resources [5,6]. It is to be noted that the proper-nouns and the abbreviations present in the Hindi-English sentences are kept unchanged while translating them into Hindi text. The parallel Hindi-English-Hindi version of the Dev-2 dataset is then used for creating the CSL tagger, as discussed in Section 5.3.2. The Dev-1 and Eval-1 datasets generated earlier for conducting Exp-1 are used for tuning and evaluation purposes, respectively. The salient details of these datasets are summarized in Table 5.7.

[5]http://www.rajbhasha.nic.in/hi/hindi-vocabulary
[6]https://hi.wiktionary.org/wiki

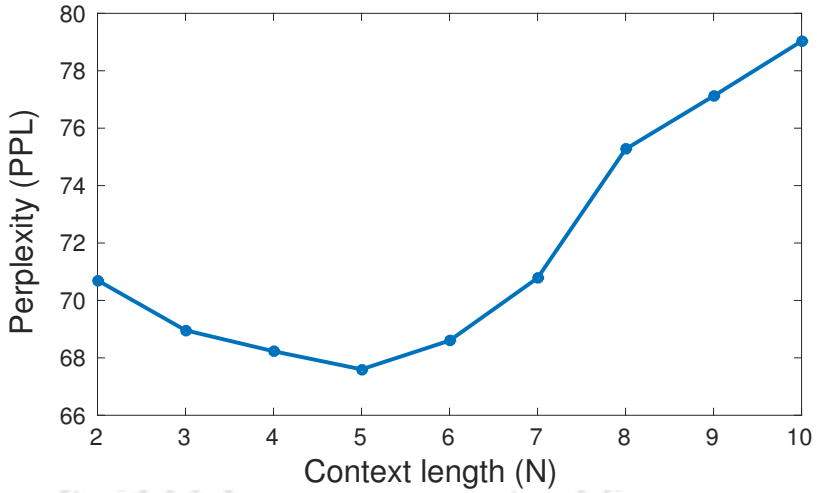


Figure 5.8: Tuning of the context length (N) on the development data. The optimal PPL score is obtained for N = 5.

5.4.3 Parameter Tuning

The normal and the factored n-gram language models used in the experimentation are trained using the SRILM toolkit [156]. For optimizing the structure of factored n-gram LMs, a genetic algorithm (GA) based factored n-gram LM has been trained using the GA-FLM toolkit [7]. It performs a standard genetic algorithm-based search for tuning the parameters. The tuning of context length has been done on the traditional n-gram LM trained on the Hindi-English Train-1 dataset, and the results are shown in Figure 5.8. It can be seen that the context length of 5 yields the best PPL score on the Hindi-English Dev-1 dataset, and the same has also been used for the factored n-gram LMs. Even for the Exp-2 case, the context length of 5 yielded the best PPL score for LMs trained on the Hindi Train-2 dataset and tested on the Hindi-English Dev-1 dataset.

The normal and the factored RNN-based language models used in the experiments are trained using the RNNLM toolkit [121]. In Exp-1 setup, for the Hindi-English Train-1 dataset having a vocabulary of 8629 words, based on the word count, the RNNLMs are trained for the top 5000 words as the output with a single hidden layer having 200 nodes and the *sigmoid* as the non-linearity function. By conducting tuning experiments on the Hindi-English Dev-1 dataset, the number of classes is set to be 100, and the variable corresponding to BPTT is set as 5. In Exp-2 setup, despite the vocabulary of the Train-2 dataset being $87,260$ words, about $10,000$ words are found to have

[7] GA-FLM Toolkit. [Online] https://github.com/zahlenteufel/GA-FLM. Accessed: 2018-02-22

a significant probability. Therefore, the RNNLMs are trained for top 10,000 words as the output with a single hidden layer having 250 nodes and the *sigmoid* as the non-linearity function. The number of classes is set to be 50, and the variable corresponding to BPTT is set as 5 by conducting tuning experiments on the Hindi-English Dev-1 dataset.

5.5 Results and Discussion

In this section, the evaluation of the modified POS tagging scheme and the proposed CSL features for Hindi-English code-switching data has been done.

5.5.1 Evaluation by Incorporating Proposed Features in Code-Switching Hindi-English Data

In this subsection, the evaluations are done by incorporating the proposed features to Hindi-English data before developing both 5-gram and RNN-based FLMs. The POS features obtained using the proposed approach are referred to as POS_P, and those obtained using the existing approach are referred to as POS_E. Table 5.8 shows the PPL scores for the Hindi-English Eval-1 dataset tested on the various FLMs trained on the Hindi-English Train-1 dataset, including both POS_E and POS_P features. The significant improvement in PPL scores for the POS_P over the POS_E highlights the efficacy of proposed innovation in the POS tagging scheme for Hindi-English code-switching data.

From Table 5.8, we note that the proposed CSL features resulted in significant improvement in PPL scores over both the kinds of POS features. By combining the CSL and POS features, further reduction in PPL scores could be achieved, and the best performances are noted for the POS_P+CSL case. For direct modeling of the code-switching data, in [88], the authors explored LID features in the context of code-switching data. Unlike the CSL feature, the LID feature tags each word in the code-switching sentence only with its corresponding language information. For better contrast, we have evaluated the LID features, and their combinations with both kinds of POS features, and the results are also listed in Table 5.8. In the case of LID features, all non-native (English) words are tagged as code-switching instances, which may not be correct for proper nouns and abbreviations. It is worth highlighting that the size of the resulting code-switching list after running the CSL tagger on the Train-1 dataset is 24,808 words, while the size of the resulting LID list is 28,757 words.

Table 5.8: Evaluation of Exp-1 setup, i.e., the incorporation of proposed textual features in the modeling of code-switching Hindi-English data. Recognition performances in terms of PPL for various FLMs trained on Hindi-English Train-1 dataset incorporating di erent combinations of features when evaluated on Hindi-English Eval-1 dataset. Note that POS_E, and POS_P refers to the POS features of Hindi-English code-switching data obtained by employing the existing approach and the proposed approach, respectively. The best performances obtained are highlighted.

Features	**PPL**	
	5-gram LM	**RNNLM**
Word	73.20	65.51
Word + POS_E	32.16	29.47
Word + POS_P	30.15	24.80
Word + CSL	29.91	24.73
Word + POS_E + CSL	28.67	20.18
Word + POS_P + CSL	**28.11**	**18.92**
Word + LID	30.21	25.08
Word + POS_E + LID	29.54	21.50
Word + POS_P + LID	28.99	19.38

Therefore, 3949 CSL tags differ from the LID tags. Also, the CSL features not only marks the location of code-switching but also provides information about the equivalent native word. These explain why the proposed CSL features have outperformed the LID features with/without the POS features. Note that the OOV rate of the Hindi-English Eval-1 and Dev-1 datasets with respect to the Hindi-English Train-1 dataset turns out to be 4.2% and 2.8%, respectively.

5.5.2 Evaluation by Incorporating Proposed Features in Monolingual Hindi Data

Before developing the n-gram and the traditional RNN-based Hindi LMs, the 5000 most frequent words extracted from Wall Street Journal (WSJ) American English text corpus [157] are added to the wordlist obtained for Train-2 Hindi dataset. This allows us to lower the OOV words while evaluating the performance for the Hindi-English test (Eval-1) and development (Dev-1) sets. When evaluated on these LMs, the Hindi-English Eval-1 and Dev-1 datasets resulted in highly degraded PPL scores. This trend is attributed to the fact that the non-native (English) words occurring in Hindi-English Eval-1 and Dev-1 datasets are devoid of the contextual information. Whereas, when the same Hindi-English Eval-1 and Dev-1 datasets are evaluated over the factored Hindi RNNLM trained,

Table 5.9: Evaluation of Exp-2 setup i.e., the incorporation of proposed textual features in the modeling of monolingual Hindi data. The performances (in terms of PPL) of the POS and the proposed CSL features in training the Hindi RNNLM in the context of Hindi-English code-switching task. Since the Hindi RNNLMs are being tuned on Hindi-English Dev-1 dataset, those performances are for reference purpose only. Also, the performances on 5-gram LM are given for contrast purposes. The best performances obtained are highlighted.

LM Type	**Features**	**PPL**	
		Eval-1	**Dev-1**
RNNLM	Word	195.29	189.70
	Word + POS	96.52	83.46
	Word + CSL	78.18	71.89
	Word + POS + CSL	**52.49**	**49.41**
5-gram	Word	224.13	213.34

including the POS features, significant reductions in the PPL scores have been achieved as shown in Table 5.9. This reduction is because the POS features also happen to capture the contextual information in the sentences. Even when the native words are replaced by the non-native words, the POS features will mostly remain unchanged. Thus, by employing the POS features along with the words while training the RNNLM, the context information helps to predict the Hindi-English word sequences efficiently.

Further, from Table 5.9, we note that the inclusion of the proposed CSL features turns out to be slightly more effective than that of the POS features. The CSL features not only carry the contextual information like the POS features but also provide direct information about the associated non-native (English) words. This explains why the CSL features have outperformed the POS features when included in the RNNLM training. Later, when the RNNLM is trained by combining CSL features along with the POS features, further reduction in PPL is achieved. This result shows that the information captured by the CSL features is additive to that of the POS features. The parameters of different kinds of LMs are tuned on the Hindi-English Dev-1 dataset. Thus, for reference purposes, the PPL scores for the Hindi-English Dev-1 dataset are also provided in Table 5.9. The OOV rate of the Hindi-English Dev-1 and Eval-1 datasets with respect to the Hindi-English Train-2 dataset turns out to be 24.97% and 25.19%, respectively. Also, when the RNNLM with only word features is learned by combining the training and the development sets (Train-2 + Dev-2), the PPL scores for Eval-1 and Dev-1 datasets turn out to be 169.27 and

158.45, respectively. Note that, the OOV rate of the Hindi-English Dev-1 and Eval-1 datasets with respect to the combined Hindi-English Train-2 + Dev-2 datasets turns out to be 6.40% and 6.42%, respectively.

Though it is well known that the RNNLMs outperform the n-gram LMs, it would be interesting to see how the n-gram LM performs on these datasets. Hence, the Hindi-English Eval-1 and Dev-1 datasets are evaluated on 5-gram LM created on the Hindi training data, and those PPL scores are also listed in Table 5.9.

5.5.3 Studying the Robustness of the Proposed Approach

The proposed POS tagging approach is found to yield significantly improved POS tags for the code-switching data. It would be interesting to study how robust is the proposed approach vis-a-vis the errors in the machine translation of the code-switching data. For studying the same, the experiments are conducted with LMs trained on data having a varying degree of machine translation errors. For simulating errors in the translation, the English translation of a Hindi-English sentence produced by the Google Translate tool is retranslated to Hindi language and then once again translated back to English. In this way, we manage to produce two logically similar English translations of that Hindi-English sentence with varying amounts of translation errors in a relative sense. For this study, we have created different versions of the LM training (Train-1) dataset by processing about 5, 10, 20, 30, and 50 percent of the available sentences through the earlier mentioned approach. Following that, a separate fRNNLM is trained on each of those versions. A few examples of Hindi-English sentences and the corresponding translated and retranslated English versions are given in Table 5.10. The PPL scores of the Eval-1 dataset with respect to those fRNNLMs are reported in Table 5.11. From Table 5.11, it can be noted that the proposed POS tagging strategy, even with 36 % of the error in the translation, performs better than that of the existing approach (PPL = 29.47).

Table 5.10: Typical Hindi-English code-switching sentences in the database and their respective English translated and retranslated versions. For simulating the translation errors, the English translation of a Hindi-English sentence produced by the Google Translate tool is retranslated to the Hindi language and then again translated back to English. Thus, two logically similar English translations of the given Hindi-English sentence are produced.

Hinglish sentences	एक बार जब आप सभी fields को complete कर दे एक blog को successful बनाने में बहुत ज्यादा time और effort लगता है आपको daily three घंटे blogging के लिए निकालना चाहिए आपने notice किया होगा की traffic समय के साथ साथ कम होता गया है
English translated versions	once you complete all the fields it takes a lot of time and effort to make a blog successful you must remove daily three hours for blogging you may have noticed that the traffic has decreased along time
English retranslated versions	once you have completed all the areas it takes a lot of time to try and make the blog a success you should take three hours daily for blogging you may have noticed that traffic has decreased over time

Table 5.11: Assessment of errors in translation for the proposed POS tagging approach. For simulating the errors during machine translation, varying percent of direct-English translated versions of Hindi-English sentences in the Train-1 are retranslated to Hindi and then translated back to English. The sentence error rates of such retranslations are computed with respect to the direct-English translations of Hindi-English sentences and those measure the effectiveness of the employed process in introducing the errors in the direct-English translation. The missed word rate quantifies the amount of the missed English words in the translated data.

Retranslated sentences (%)	0	10	20	30	50
Sentence error rate (%)	0.00	03.96	08.15	16.19	24.39	36.06
Missed word rate (%)	04.05	04.60	05.34	06.28	07.99	09.61
PPL	24.80	24.88	24.97	25.56	26.64	28.91

5.6 Revalidation on Mandarin-English Code-Switching Data

For revalidation of our findings, a freely available [8] subset of SEAME: Mandarin-English code-switching text corpus has been used. Similar to that of the Hindi-English code-switching case, the evaluations of proposed features for Mandarin-English code-switching data have also been done by following both the direct training of Mandarin-English LMs and the interpolation of monolingual Mandarin LMs. For this study, the database has been prepared in a similar manner, as described in Section 5.4 for the Hindi-English case.

At first, the available text data has been partitioned in two sets: one containing 1008 monolin-

[8] https://github.com/zengzp0912/SEAME-dev-set/blob/master/dev_man/text

Table 5.12: The details of the vocabulary size and the word count of the training, development and evaluation datasets created for re-evaluating the proposed features under two di erent experimental conditions.

Experiment	Dataset	Data type	# sent.	# words		# unique words	
				Mandarin	English	Mandarin	English
Exp-1	Train-1	Man-Eng	5,000	48,384	19,369	1,294	2,882
	Dev-1		500	9.965	1,426	577	538
	Eval-1		1,031	13,461	2,951	660	993
Exp-2	Train-2	Mandarin	1,008	14,656	557	1,560	163
	Dev-2a	Mandarin	1,521	18,338	735	1,662	184
		Man-Eng	1,521	15,675	3,438	735	1,143
	Dev-2b	Man-Eng	101	2,017	209	331	138
	Eval-2	Man-Eng	141	2,291	276	372	201

gual Mandarin sentences, and the other containing 6531 intra-sentential code-switching Mandarin-English sentences. For conducting Exp-1, the 6531 Mandarin-English sentences are further partitioned into Train-1, Dev-1 and Eval-1 datasets consisting of 5000, 500, and 1031 sentences, respectively. Unlike the Hindi-English case, for conducting Exp-2, we could not find a separate Mandarin corpus to train the monolingual LM. Therefore, the earlier obtained 1008 monolingual Mandarin sentences are used as the Train-2 dataset. On account of that, the use of the Eval-1 set is no longer feasible in Exp-2 setup since it has a very high OVV rate (27.69%) with respect to the Train-2 set and, therefore a new evaluation set is required to be defined. Further, due to the OVV issue, instead of a single development set, we require two development sets: one lager set for training the CSL tagger and the other smaller set for LM tuning purposes which has a comparable OOV to that of the newly defined evaluation set. For those purposes, a few sentences from the available Mandarin-English dataset are selected randomly and partitioned into Dev-2a, Dev-2b and Eval-2 datasets consisting of 1521, 101 and 141 sentences, respectively. Later, for developing the CSL tagger, the Mandarin-English sentences in the Dev-2a dataset are translated to Mandarin sentences by following the similar procedure that has already been described in Section 5.4.1. The salient details of these datasets are given in Table 5.12.

The distributions of the English and Mandarin words present in the Mandarin-English text corpus with respect to their POS labels are shown in Figure 5.9. On comparing Figures 5.7 and 5.9, it can be noted that similar to Hindi-English corpus, even in Mandarin-English corpus, the majority of the English words that are being code-switched are nouns, proper nouns, and adjectives. The

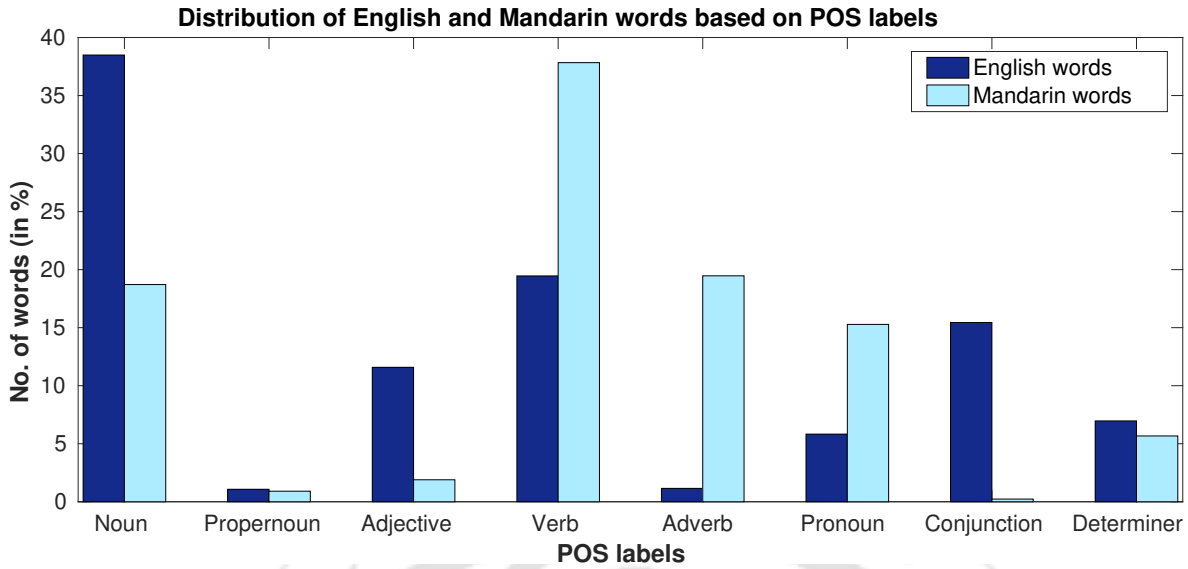


Figure 5.9: Distributions of the English and Mandarin Words in the Mandarin-English text corpus based on their parts of speech (POS) labels. Note that, for the ease in display, the POS tags for Mandarin and English words have been normalized separately, i.e., all POS labels corresponding to English/Mandarin words sum up to 100%.

POS and CSL features are extracted by following the similar procedure described in Sections 5.2 and 5.3, respectively. The POS tags for the Mandarin and English words are extracted using the Stanford log-linear part-of-speech tagger [150]. The procedure for training both the simple and factored LMs remains the same as that of the earlier case.

5.6.1 Evaluation by Incorporating Proposed Features in Code-Switching Mandarin-English Data

In this subsection, the evaluations are done on both 5-gram and RNN-based LMs trained on the Mandarin-English Train-1 dataset of Exp-1 defined in Table 5.12. On evaluating, we notice a similar trend in the PPL measures for both POS and CSL features as that of the Hindi-English LMs case. The evaluation results are shown in Table 5.13. Note that, the OOV rate of the Mandarin-English Eval-1 and Dev-1 datasets with respect to the Mandarin-English Train-1 dataset turns out to be 5.53% and 2.67%, respectively.

Table 5.13: Evaluation of Exp-1 setup i.e., the incorporation of proposed textual features in the modeling of code-switching Mandarin-English data. PPL scores for various FLMs trained on Mandarin-English training dataset incorporating di erent combinations of features when evaluated on Mandarin-English Eval-1 dataset. The best performances obtained are highlighted.

Features	**PPL**	
	5-gram LM	**RNNLM**
Word	98.19	87.76
Word + POS_E	24.03	14.99
Word + POS_P	20.84	13.89
Word + CSL	20.38	14.40
Word + POS_E + CSL	20.09	13.22
Word + POS_P + CSL	**20.03**	**12.50**
Word + LID	21.78	14.94
Word + POS_E + LID	20.56	14.27
Word + POS_P + LID	20.17	13.63

Table 5.14: Evaluation of Exp-2 setup i.e., the incorporation of proposed textual features in the modeling of monolingual Mandarin data. PPL scores of the POS and the CSL features in training the Mandarin RNNLM in context of Mandarin-English code-switching task. Since, the Mandarin RNNLMs are being tuned on Mandarin-English Dev-2 dataset, those performances are for reference purpose only. Also, the performances on 5-gram LM are given for contrast purposes.

LM Type	**Features**	**PPL**	
		Man-Eng (Eval-2)	**Man-Eng (Dev-2b)**
RNNLM	Word	104.9	67.13
	Word + POS	19.13	12.09
	Word + CSL	19.80	13.40
	Word + POS + CSL	**13.02**	**10.67**
5-gram	Word	120.88	82.67

5.6.2 Evaluation by Incorporating Proposed Features in Monolingual Mandarin Data

On evaluating the proposed features on monolingual Mandarin LMs, we notice a similar trend in the PPL scores for both POS and CSL features as that of the Hindi LMs case. The evaluation results are shown in Table 5.14. Note that, the OOV rate of the Mandarin-English Eval-2 and Dev-2b datasets with respect to the Mandarin Train-2 dataset turns out to be 6.61% and 2.24%, respectively.

Table 5.15: Assessment of textual feature augmented Hindi-English code-switching FLMs on the proposed T2W transduction scheme for E2E ASR systems trained on reduced/combined target set. The recognition performances are reported in terms of percentage word error rate (% WER) along with the 95% con dence interval in brackets

E2E System	Features for FLM	% WER (95% confidence interval)	
		Reduced	Combined
Attention	Word	31.1 (0.47)	32.3 (0.47)
	Word + POS	30.43 (0.46)	31.1 (0.47)
	Word + CSL	30.8 (0.47)	31.7 (0.47)
	Word + CSL + POS	**29.8 (0.46)**	30.8 (0.47)
CTC	Word	35.1 (0.48)	38.0 (0.49)
	Word + POS	34.1 (0.48)	35.2 (0.48)
	Word + CSL	33.8 (0.48)	35.8 (0.48)
	Word + CSL + POS	**33.2 (0.47)**	34.6 (0.48)

5.7 Assessment of the Proposed Textual Features in T2W Transduction

As argued earlier, the T2W transduction performance is highly dependent on the quality of the context information provided by the LM. Therefore, the fRNNLM trained on the proposed POS and CSL textual features is incorporated in T2W transduction for improving the context modeling of code-switching LM. The evaluation has been done for two different conditions, where, the fRNNLM is trained on (i) code-switching data, and (ii) monolingual native language data. The recognition performances of those experiments are reported in terms of WER in Tables 5.15 and 5.16, respectively, for both attention- and CTC -based E2E ASR frameworks.

5.7.1 Evaluation of T2W Transduction with Code-Switching fRNNLM

From Table 5.15, it can be noted that, with the inclusion of the FLM trained on the proposed CSL textual features, about 4.2% and 5.4% relative reductions have been achieved in the WER scores in comparison to default RNNLM for attention- and CTC-based E2E frameworks, respectively. This improvement is attributed to (i) the binary categorization of code-switching, and (ii) tagging of code-switching (English) words in the training data to their corresponding native (Hindi) words. For the code-switched words having a little or no evidence in the training data, the FLM falls back to their equivalent Hindi words, if those exist in the vocabulary. Note that, unlike the CSL

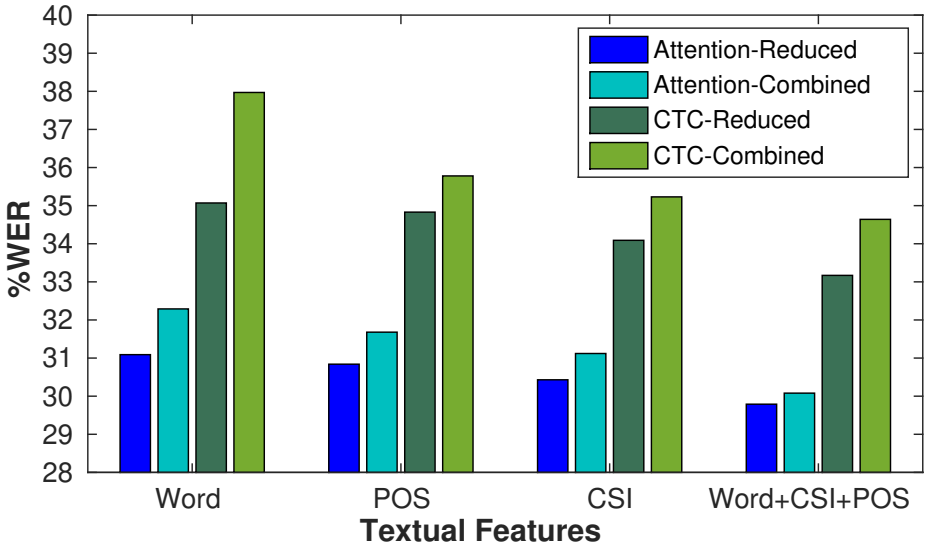


Figure 5.10: Assessment of the impact of proposed context-dependent T2W transduction with/without textual features on attention- and CTC-based E2E ASR systems.

feature, the grouping induced by the POS features are targeted towards sentence structure rather than code-switching. This could be the reason why the CSL feature not only outperformed the POS features but also provided an additive improvement when combined with the POS features.

For the ease of assessing the relative impact of the proposed context-dependent T2W transduction with/without textual features, the performances for the attention- and CTC-based E2E ASR systems are summarized in Figure 5.10. It can be noted that a similar trend in the performances is observed for both E2E frameworks. At the same time, we wish to point out that the developed CTC-based E2E system does not incorporate any character-level LM while decoding the combined/reduced targets. Therefore, the performances of the attention- and CTC-based E2E ASR systems cannot be directly compared.

5.7.2 Evaluation of T2W Transduction with Monolingual fRNNLM

Table 5.16 presents the evaluation results for the T2W transduction scheme that employs the monolingual Hindi LM trained by incorporating the proposed textual features. From Table 5.16, it can be noted that the proposed CSL textual features has shown significant improvements in the WER measures on both attention- and CTC-based systems. These improvements are due to the fact that the CSL feature not only identifies the code-switching locations but also provides information about the possible native language word that has been switched. This information

Table 5.16: Assessment of textual feature augmented monolingual Hindi FLMs on the proposed T2W transduction scheme for E2E ASR systems trained on reduced/combined target set. The recognition performances are reported in terms of percentage word error rate (% WER) along with the 95% con dence interval in brackets

E2E system	**Features for FLM**	**%WER (95% confidence interval)**	
		Reduced	**Combined**
Attention	Word	37.5 (0.49)	38.8 (0.49)
	Word + POS	36.2 (0.48)	37.6 (0.49)
	Word + CSL	35.7 (0.48)	37.2 (0.49)
	Word + CSL + POS	**35.5 (0.48)**	37.1 (0.49)
CTC	Word	39.8 (0.49)	40.7 (0.50)
	Word + POS	38.3 (0.49)	39.1 (4 0.49)
	Word + CSL	37.6 (0.49)	38.7 (0.49)
	Word + CSL + POS	**37.4 (0.49)**	38.2 (0.49)

helps in providing better context information for the LM while generating the desired word sequence during the transduction.

5.8 Conclusion

In this chapter, we have explored factored language models in the context of code-switching task. We first explored a strategy for efficient POS tagging of the code-switching data. Additionally, the CSL feature has been proposed to handle code-switching data. The proposed CSL feature is simple to estimate yet found to be quite effective. The evaluations are done for two code-switching language pairs, namely, Hindi-English and Mandarin-English. The significance of this work lies in two aspects. First, the approach attempts to enhance the recognition ability of the FLMs in the context of code-switching task. Second, we attempt to address the code-switching data scarcity issue by enhancing the ability of native language LM without the need to incorporate the code-switching data by employing the proposed features. Further, we have employed the developed FLMs into the earlier proposed T2W transduction scheme for enhancing the E2E ASR system performance.

From the experimental evaluations, we note that the performances of the code-switching E2E ASR systems have been significantly improved by incorporating FLMs into the T2W transduction scheme. These improvements in WER measure are attributed to the ability of the proposed CSL feature to model the context information associated to the code-switching words effectively.

6

Conclusions

Contents

6.1 Summary of the Work 100
6.2 Summary of Contributions 103
6.3 Future Directions 104

6.1 Summary of the Work

There are three major goals that are addressed in this thesis. The first one is the development of a robust end-to-end (E2E) Hindi-English code-switching automatic speech recognition (ASR) system, in low-resource condition. Towards that end, a reduced target set, and a context-dependent target-to-word (T2W) transduction scheme have been proposed. The second one is the proposition of code-switching location (CSL) textual feature that helps both the code-switching and monolingual language models (LMs) to efficiently handle the code-switching phenomenon. And, the third one is the creation of Hindi-English code-switching corpus for addressing the data scarcity issue.

The thesis begins with an introduction to ASR and a brief historical review on its evolution. It is followed by a discussion on the different ASR paradigms such as hybrid and E2E. The hybrid ASR systems employ a modular framework, where each module is trained and optimized independently with different objective functions. Hence, the resulting system can be sub-optimal. Also, in the hybrid framework, the time information regarding the start and end of phones is required for training the ASR system. For addressing those issues, the E2E framework has emerged as a viable alternative to conventional hybrid systems in the ASR domain. In this framework, the network is trained with characters as the output targets and hence does not include any explicit pronunciation model. Also, the E2E ASR framework does not require the phonetically labeled training data. However, both these frameworks are mainly developed for monolingual ASR task, and cannot effectively handle code-switching. The focus of this thesis is on addressing the challenges caused by the code-switching phenomenon while developing the ASR systems on a limited code-switching data.

Code-switching has become a widespread phenomenon in our day-to-day life, and hence there is a greater need to handle the code-switching by the spoken input systems. In chapter 2, we discuss the code-switching phenomenon along with a brief review on the available code-switching resources and the existing research works done in the fields of linguistics, speech, and language processing. The scope of this thesis is limited to only the acoustic and language modeling of code-switching data. From literature, it is to be noted that a very small sized code-switching speech database is available in the Indian context. Towards addressing that constraint, as a part of this work, we have

developed a Hindi-English code-switching text and speech corpora referred to as HingCoS corpus. The details of this database and the protocol followed for its development have been provided in Chapter 3. This corpus consists of Hindi-English code-switching text data having 26k sentences. Also, in addition to that, the corpus contains 25 hours of matching speech data collected from 101 (61 male and 40 female) speakers between the age group of 19 40 years. The speech data is recorded in read-style, while the transcripts collected from the web sources are written in a conversational style. Another unique feature of the HingCoS corpus is the diversity in the linguistic background of the speakers who contributed the speech data. The native language of only 42.57% of the speakers is Hindi, while the remaining ones come from other Indian language backgrounds. Among the publicly accessible code-switching corpora created in different contexts, the SEAME corpus happens to be the largest one. It is followed by the HingCoS corpus created in this work. For the sake of sanity check, the corpus has also been evaluated for language modeling and ASR tasks and their recognition performances were reported. Among all the ASR systems developed, the TDNN-HMM based system with the 5-gram LM yields the best WER of 19.5%. For promoting more research in the code-switching domain, the developed HingCoS corpus is made public, and the details for sharing the same are given on the corpus webpage [1].

In Chapter 4, the E2E framework for ASR of Hindi-English code-switching data has been explored by employing the developed HingCoS corpus. The conventional E2E ASR systems are trained directly from speech data (filterbank energies) with characters as the target labels. In the context of code-switching, a conventional E2E ASR system models the unified character set of the underlying languages. With the unified character set modeling, such a system would suffer from the following issues: (i) more than double expansion in the target set and enhanced confusability among them due to acoustic similarity, and (ii) requirement of more data and compute resources for reliable modeling. Towards addressing those challenges, first, a reduced target set labels were proposed for training the E2E ASR system for low-resourced code-switching data. Interestingly, the reduced target set based system outperformed the combined target set one in terms of the target error rate (TER). But, when those target sequences were converted to word sequences, the word error rate (WER) score is noted to be degraded. This degradation is due to enhanced confusability among the

[1] https://www.iitg.ac.in/eee/emstlab/HingCoS_Database/HingCoS.html

homophones within or across the languages involved. For addressing the same, a context-dependent T2W transduction scheme based on the modularized decoding framework was proposed in this work. This scheme employs an explicit error model (EM) along with an LM for better context modeling. The proposed transduction scheme is noted to achieve a relative improvement of 22% in WER over the naive transduction scheme in the context of the reduced target set based Hindi-English code-switching E2E ASR. This improvement in WER measure is mainly due to the better context information provided the employed LM.

Motivated by the above cited reason, in Chapter 5, the language modeling of code-switching data has been explored. The aim of an LM is to predict the probability of occurrence of a given sequence of words from the training data. The traditional LMs those employed for monolingual case suffers from lack of generalizability between training and testing contexts for code-switching case. This is because the conventional LMs simply captures the sequence information while the semantic information is ignored. This issue has been addressed by modeling the semantics of the sentence along with the syntax using a factored language modeling technique. For capturing the semantics of the code-switching sentence, a novel textual feature referred to as a CSL textual feature has been proposed. This feature helps the LM in identifying the possible code-switching locations. Also, a modified parts-of-speech (POS) tagger has been proposed for Hindi-English code-switching data. The proposed features have been evaluated on the FLMs trained on (i) Hindi-English code-switching data when sufficient amount of such data is available for training purpose, and (ii) monolingual Hindi data when limited amount of code-switching data is available for adaptation/augmentation purpose. The evaluation of the proposed features has also been done on two code-switching datasets: Hindi-English and Mandarin-English. On experimental evaluation, the proposed CSL features provide an independent and additive improvement over the POS features in terms of perplexity. Finally, the FLMs trained on the proposed textual features are incorporated into the proposed T2W transduction scheme and noted to significantly improve the E2E ASR system performance in terms of WER.

The main aim of this thesis is to explore ways for improving the recognition performance of the Hindi-English code-switching E2E ASR systems. Though, we have evaluated the performances for the code-switching hybrid ASR systems, they are not for the contrast purposes. Unlike the hybrid ASR systems, the E2E ASR systems are still in the evolution phase. So, it would be

more appropriate to contrast the performances of the Hindi-English code-switching ASR systems developed in this thesis with the Mandarin-English code-switching ASR systems reported recently in [158]. In that work, the authors have used the SEAME corpus [75] and the performances of both hybrid and E2E ASR systems are reported in terms of mixed error rate (MER). The MER is defined as the combination of WER for English and CER for Mandarin. In the hybrid setups, the WER is 19.47% for Hindi-English case, while the MER is 31.7% for Mandarin-English case. Similarly, in the E2E setups, those are 29.79% and 39.1% for Hindi-English and Mandarin-English cases, respectively. Thus, the developed ASR systems for Hindi-English code-switching data follow the similar trends as noted for Mandarin-English data as reported in the literature.

6.2 Summary of Contributions

The salient contributions of this thesis are listed below.

Developed a Hindi-English code-switching text and speech corpora referred to as the *HingCoS* corpus. The corpus consists of 25 hours of speech data corresponding to 101 speakers. For the sake of sanity check, the baseline ASR systems are also developed by employing the hybrid framework. To promote more research in the code-switching domain, the HingCoS corpus is made publicly available.

Explored the E2E framework for developing an ASR system for Hindi-English code-switching data. Also, a reduced target set has been created for training the low-resourced Hindi-English code-switching E2E ASR system by exploiting the acoustic similarity. The reduced target set based E2E ASR system avoids confusability among the cross-lingual targets. It requires much lesser memory and computational time when compared to the existing combined target set-based system.

Proposed a novel context-dependent T2W transduction scheme which efficiently converts the character/phoneme sequences outputted by the E2E ASR system into the desired word sequences. This scheme employs a modularized decoding approach with an explicit EM and an LM to provide context information. On evaluating, the proposed scheme resulted in a WER of 31.1%.

To facilitate the LMs to predict the code-switching instances, a novel textual feature referred to as the code-switching location (CSL) feature has been proposed. Additionally, we proposed an improved POS labeling scheme for accurate tagging of non-native words embedded in the code-switching data. Both these textual features have shown significant improvements in LM recognition performance in terms of perplexity when evaluated by directly incorporating them into code-switching data. Also, similar improvements in perplexity have been noted when those textual features are evaluated on a more viable approach that adapts an existing native monolingual LM to handle the code-switching data. Further, when the FLMs trained on the proposed textual features are incorporated into T2W transduction scheme, a relative improvement of 4.2% in WER scores has been achieved when compared to the RNNLM based T2W transduction.

It is worth highlighting that, though the proposed approaches are evaluated for the Hindi-English code-switching case, they are generic enough to be applied for any other valid code-switching language pair across the globe.

6.3 Future Directions

This work highlights the creation of a moderate size Hindi-English code-switching corpus along with the challenges and proposed enhancements in developing the E2E ASR systems under the low-resourced condition. Despite the success of the proposed approaches, they still face a few challenges. In the following, we have discussed those issues and the possible future directions that can address those challenges.

From Tables 3.11 and 5.15, it is observed that the performance of the baseline hybrid ASR systems (WER of 19.47%) is superior to that of the E2E ASR systems (WER of 29.79%). The possible reason for such behavior could be the use of the weighted finite state transducer (WFST) in hybrid approach [159]. In that approach, the WFSTs provide a common and natural representation for HMMs, context-dependency, pronunciation dictionaries, grammars, and alternative recognition outputs. Furthermore, a general transducer operations combine these representations flexibly and efficiently. Weighted determinization and minimization algorithms optimize their time and space requirements, and a weight pushing algorithm distributes the weights along the paths of a weighted

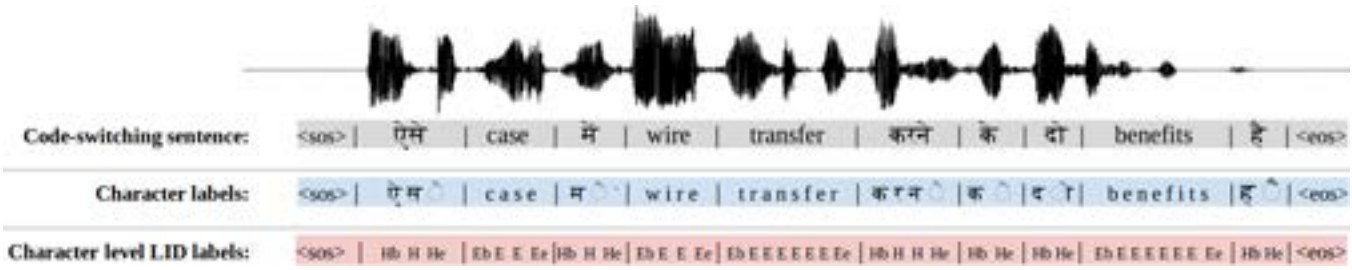


Figure 6.1: Creation of character-level LID tags for the training data towards conditioning the E2E networks to perform LID task on code-switching speech. The `*H=E*' denotes the Hindi/English LID tag. The `*b=e*' label is appended to the `*H=E*' LID tag to mark the begin/end characters.

transducer optimally for speech recognition. Unlike the hybrid ASR framework, in the proposed context-dependent T2W transduction based E2E ASR framework, the acoustic scores are not taken into consideration while decoding. Therefore, the first extension for this work can be to explore the ways to incorporate the WFST into the E2E ASR framework. Recently, in the similar lines, the authors in [160] proposed an approach to optimize the E2E system directly at word-level instead of characters.

The second one is that the efficacy of the proposed CSL features is dependent on the quality of the employed online machine translator. Hence, this work can be extended by exploring alternate approaches, which makes those features independent of the machine translator. For extracting the CSL features for code-switching data, one way could be to employ an English-to-Hindi dictionary to get the desired semantic Hindi word corresponding to the English word and vice-versa.

Also, in literature,a few of the existing code-switching E2E ASR works [60, 61, 97], employ the multi-task learning (MTL) framework involving the frame-level language identification (LID) along with the character set at the output. Those works have shown that the incorporation of LID information while training the code-switching E2E ASR system could enhance the recognition performance. Motivated by that, in our recent work [161] on joint language identification of Hindi-English code-switching speech, we have proposed a novel LID tagging scheme that can predict not only the word-level LID information but also the instances (or boundaries) where the code-switching occurs. For achieving that, for each of the training utterances, first, the given orthographic transcription is transformed into character level transcription. Later, each character in the transcription is mapped to the corresponding LID tags. This process is illustrated in Figure 6.1. This is to highlight that,

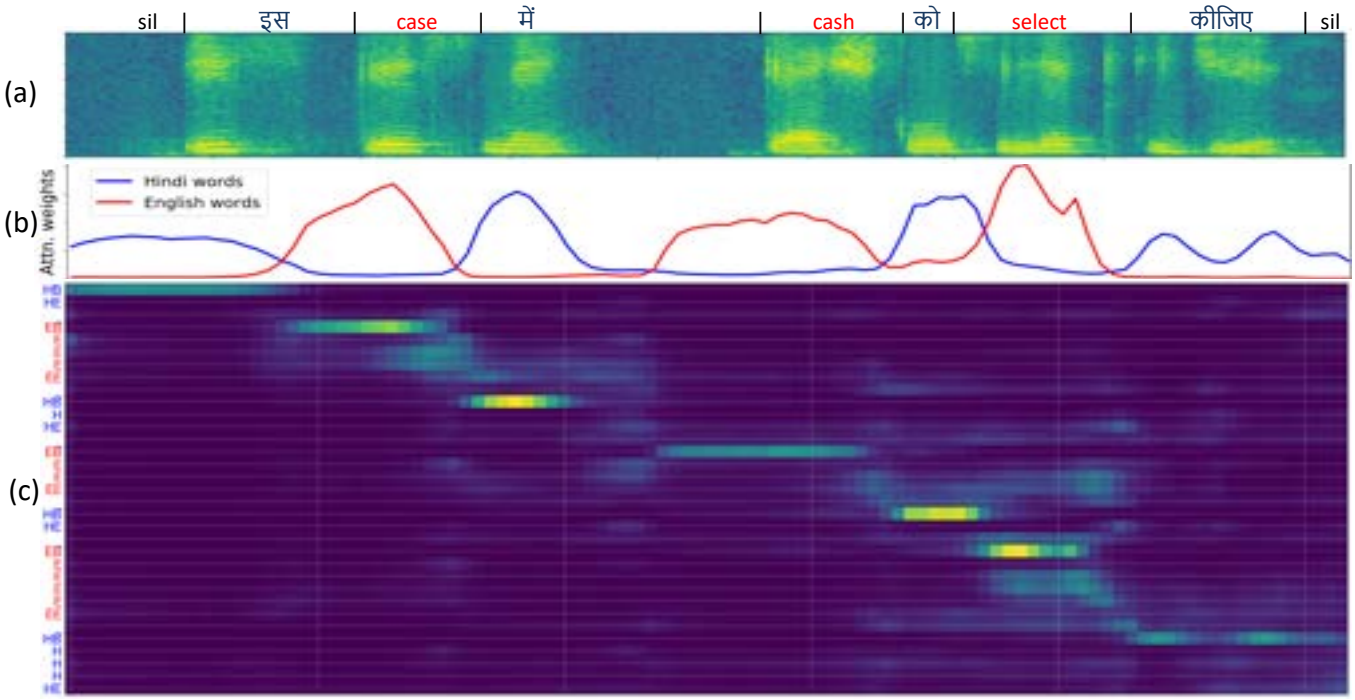


Figure 6.2: Visualization of attention mechanism for LID task. For a given Hindi-English code-switching utterance: a) spectrogram labeled with Hindi and English word boundaries for reference purpose. (b) variation of attention weights with respect to time for Hindi and English languages, and (c) alignment produced by the attention network for the input speech and the decoded output LID labels.

in the orthographic transcription of the training data, the Hindi and English words are written in their respective scripts. So, the character-level LID tags as 'H/E' (Hindi/English) are produced in a straight forward manner, except that additional labels 'b' and 'e' are appended to the tags of *begin* and *end* characters of each word, respectively. Also, a blank symbol 'j' has been inserted between words to ease the marking of the word boundary. For training the E2E models, a total of 7 labels which include 6 LID tags (Hb, H, He, Eb, E, Ee), one blank label (j) are given as targets to generate the output posterior probabilities. On evaluating the proposed target labeling scheme, the attention-based E2E system predicts the language boundaries more accurately. This is attributed to the ability of attention mechanism in LAS network to accurately predict the languages switching in the data. To highlight that, we have computed the language-specific averaged attention weights with respect to the decoded LID label sequence and the plot for the same is shown in Figure 6.2. The description of each of the subplots in Figure 6.2 is presented next.

Figure 6.2(a) shows the spectrogram of a typical Hindi-English speech utterance in the test set. Note that, the spectrogram is manually labeled with spoken words and their boundaries for

the reference purposes. The variations of the averaged attention weights for Hindi and English language targets present in the input speech data with respect to time, are shown in Figure 6.2(b). The sequence alignment produced by the attention network for the input speech data (on the x-axis) and the decoded output LID labels (on the y-axis) is plotted in Figure 6.2(c). From Figures 6.2(b) and 6.2(c), we observe that the attention weights for Hindi and English languages mostly peak around the corresponding word locations.

As the third direction, this work can be further extended by exploring the proposed LID labeling scheme as a supervision in the MTL framework. It is also worth highlighting that, the sequence alignment produced by the attention weight profiles may also carry the speaking style information that can be exploited in speaker recognition.

A

Neural Network Architectures: RNN and LSTM

Contents

A.1 RNN Architecture 110
A.2 LSTM Architecture 110

A.1 RNN Architecture

The feed-forward neural networks are not well suited to capture the long-term dependencies. Whereas, the RNN, with its feedback connection is able to model the long term dependencies as well as the temporal variability present in the signal. The block diagram of the unfolded version of the RNN architecture is given in Figure A.1 (a). Given an input sequence $x = (x_1, x_2, \ldots, x_T)$, a standard RNN computes the hidden vector sequence $h = (h_1, h_2, \ldots, h_T)$ and output vector sequence $y = (y_1, y_2, \ldots, y_T)$ by iterating the following equations from $t = 1$ to T.

$$h_t = H(W_{xh}.x_t + W_{hh}.h_{t1} + b_h) \tag{A.1}$$

$$y_t = W_{hy}.h_t + b_y \tag{A.2}$$

where, the W, b, and H terms denote the weight matrices, the bias vectors, and the hidden layer activation function (for example, W_{xh} is the input-hidden weight matrix, b_h is hidden bias vector, and H is an element-wise application of a sigmoid function). These weights in the RNN architecture are updated by employing a special training algorithm referred to as BPTT.

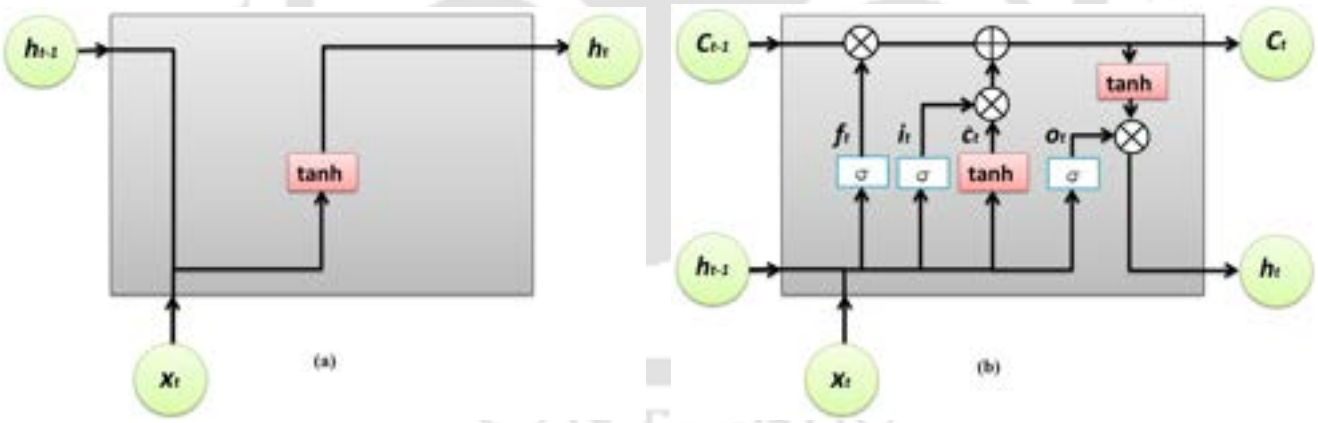


Figure A.1: Block diagrams of the unfolded network architecture of (a) the RNN, and (ii) the LSTM.

A.2 LSTM Architecture

Though the RNNs can model the long-term dependencies, they suffer from the well-known vanishing gradient problem caused due to the BPTT algorithm. It means that over a period of time, the backpropagated gradient of the error function either exponentially blows up or decays. This results in improper adaptation of weights in the next time steps. To overcome this issue, a

modified RNN architecture known as long short-term memory (LSTM) network has been proposed in the literature. In LSTM architecture, the recurrent layer contains memory cells that can store the temporal state of the neural network along with three special gates to control the information flow. The block diagram of the LSTM architecture is given in Figure A.1 (b).

For the input signal x_t at the time instant t, the flow of information to the memory cell c_t is decided with help of *input* and *forget* gates controlling how much information the network needs to remember and forget. Let i_t and f_t denote the information that the network remembers and forgets, respectively. Also, let the output corresponding to the conventional RNN be denoted as $\tilde{c}_t$. Combining these three information the contribution to the memory cell c_t is determined. Further, the information o_t from the memory cell c_t that the network passes to the next stage is controlled by an *output* gate. These operations are mathematically represented as,

$$f_t = \sigma(w_f.[h_{t-1}, x_t] + b_f) \tag{A.3}$$

$$i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i) \tag{A.4}$$

$$\tilde{c}_t = \tanh(w_c.[h_{t-1}, x_t] + b_c) \tag{A.5}$$

$$c_t = f_t \quad c_{t-1} + i_t \quad \tilde{c}_t \tag{A.6}$$

$$o_t = \sigma(w_o.[h_{t-1}, x_t] + b_o) \tag{A.7}$$

$$h_t = o_t \quad \tanh(c_t) \tag{A.8}$$

where $w_{\{f,i,c,o\}}$ and $b_{\{f,i,c,o\}}$ denote the weight and the bias associated with the respective networks. Similar to the feed-forward DNN, the LSTM layers can also be stacked to build the deeper architecture. Though the single LSTM layer can itself capture long-term dependencies, the use of deep LSTM (DLSTM) can help in distributing the parameters over multiple layers in the DLSTM network instead of increasing the model size of a single LSTM network.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

Bibliography

[1] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. PTR Prentice Hall, 1993.

[2] F. Jelinek, \Continuous speech recognition by statistical methods," in *Proceedings of the IEEE*, vol. 64, no. 4, 1976, pp. 532{556.

[3] S. Ghai, \Addressing pitch mismatch for childrens automatic speech recognition," Ph.D. dissertation, Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, India, 1995.

[4] R. P. Lippmann, \Speech recognition by machines and humans," *Journal of Speech Communication*, vol. 22, no. 1, pp. 1{15, 1997.

[5] O. Scharenborg, \Reaching over the gap: A review of e orts to link human and automatic speech recognition research," *Journal of Speech Communication*, vol. 49, no. 5, pp. 336{347, 2007.

[6] J. J. Gumperz, *Discourse Strategies*. Cambridge University Press, 1982.

[7] C. M. Eastman, \Codeswitching as an urban language-contact phenomenon," *Journal of Multilingual & Multicultural Development*, vol. 13, no. 1-2, pp. 1{17, 1992.

[8] C. M. Scotton, \Comparing codeswitching and borrowing," *Journal of Multilingual & Multicultural Development*, vol. 13, no. 1-2, pp. 19{39, 1992.

[9] K. Bali, J. Sharma, M. Choudhury, and Y. Vyas, \I am borrowing ya mixing? An analysis of English-Hindi code mixing in Facebook," in *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 2014, pp. 116{126.

[10] A. Das and B. Gambäck, \Code-mixing in social media text: The last language identi cation frontier?" *Traitement Automatique des Langues (TAL), Special Issue on Social Networks and NLP*, vol. 54(3), 2015.

[11] B.-H. Juang and L. R. Rabiner, \Automatic speech recognition{A brief history of the technology development," *Georgia Institute of Technology, Atlanta Rutgers University, and University of California. Santa Barbara*, vol. 1, p. 67, 2005.

[12] K. H. Davis, R. Biddulph, and S. Balashek, \Automatic recognition of spoken digits," *Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637{642, 1952.

[13] H. F. Olson and H. Belar, \Phonetic typewriter," *Journal of the Acoustical Society of America*, vol. 28, no. 6, pp. 1072{1081, 1956.

[14] J. W. Forgie and C. D. Forgie, \Results obtained from a vowel recognition computer program," *Journal of the Acoustical Society of America*, vol. 31, no. 11, pp. 1480{1489, 1959.

[15] | | , \Recognition of Japanese vowels{Preliminary to the recognition of speech," *Journal of the Radio Research Laboratory*, vol. 37, no. 8, pp. 193{212, 1961.

[16] K. Nagata, Y. Kato, and S. Chiba, \Spoken digit recognizer for the Japanese language," *Audio Engineering Society Convention*, vol. 12, no. 4, pp. 336{342, 1964.

[17] J. Baker, \The DRAGON system: An overview," *Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 24{29, 1975.

[18] L. E. Baum, \An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, no. 1, pp. 1{8, 1972.

[19] H. White, \Maximum likelihood estimation of misspeci ed models," *Econometrica: Journal of the Econometric Society*, pp. 1{25, 1982.

[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Elsevier, 2013.

[21] L. R. Rabiner, \A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, no. 2, 1989, pp. 257{286.

[22] D. B. Roe and J. G. Wilpon, \Whither speech recognition: The next 25 years," *IEEE Communications Magazine*, vol. 31, no. 11, pp. 54{62, 1993.

[23] W. S. McCulloch and W. Pitts, \A logical calculus of the ideas immanent in nervous activity," *The bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115{133, 1943.

[24] R. P. Lippmann, \Review of neural networks for speech recognition," *Journal of Neural Computation*, vol. 1, no. 1, pp. 1{38, 1989.

[25] H. Bourlard and C. J. Wellekens, \Links between Markov models and multilayer perceptrons," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1167{1178, 1990.

[26] S. Kurogi, \Speech recognition by an arti cial neural network using ndings on the a erent auditory system," *Journal of Biological Cybernetics*, vol. 64, no. 3, pp. 243{249, 1991.

[27] C. P. Lim, S. C. Woo, A. S. Loh, and R. Osman, \Speech recognition using arti cial neural networks," in *Proceedings of the 1st International Conference on Web Information Systems Engineering*, vol. 1, 2000, pp. 419{423.

[28] J. Tebelskis, \Speech recognition using neural networks," Ph.D. dissertation, Carnegie Mellon University, 1995.

[29] E. Trentin and M. Gori, \A survey of hybrid ANN/HMM models for automatic speech recognition," *Journal of Neurocomputing*, vol. 37, no. 1-4, pp. 91{126, 2001.

[30] S. J. Young and S. Young, *The HTK Hidden Markov Model Toolkit: Design and Philosophy*. University of Cambridge, Department of Engineering Cambridge, England, 1993.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, \The Kaldi speech recognition toolkit," in *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE Signal Processing Society, 2011.

[32] T. E. Oliphant, \Python for scienti c computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10{20, 2007.

[33] A. Graves and N. Jaitly, \Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, 2014, pp. 1764{1772.

[34] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, \Tensor ow: A system for large-scale machine learning," in *Proceedings of the 12th Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265{283.

[35] V. Renkens, \Nabu: An end-to-end speech recognition toolkit," [Online] https://vrenkens.github.io/nabu/, accessed: 2019-03-24.

[36] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, \ESPnet: End-to-end speech processing toolkit," *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2018.

[37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, \Automatic di erentiation in PyTorch," in *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[38] S. Davis and P. Mermelstein, \Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357{366, 1980.

[39] J. Makhoul, \Spectral linear prediction: Properties and applications," *Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 3, pp. 283{296, 1975.

[40] H. Hermansky, \Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738{1752, 1990.

[41] H. J. Kelley, \Gradient theory of optimal ight paths," *Journal of American Rocket Society*, vol. 30, no. 10, pp. 947{954, 1960.

[42] G. E. Dahl, D. Yu, L. Deng, and A. Acero, \Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30{42, 2012.

[43] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, \Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82{97, 2012.

[44] M. Wester, \Pronunciation modeling for ASR{knowledge-based and data-derived methods," *Journal of Computer Speech & Language*, vol. 17, no. 1, pp. 69{85, 2003.

[45] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, \Class-based n-gram models of natural language," *Journal of Computational Linguistics*, vol. 18, no. 4, pp. 467{479, 1992.

[46] M. Sundermeyer, I. Oparin, J.-L. Gauvain, B. Freiberg, R. Schluter, and H. Ney, \Comparison of feedforward and recurrent neural network language models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8430{8434.

[47] M. Sundermeyer, R. Schluter, and H. Ney, \RWTHLM-The RWTH aachen university neural network language modeling toolkit." in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2014, pp. 2093{2097.

[48] T. Mikolov, M. Kara at, L. Burget, J. Cernocky, and S. Khudanpur, \Recurrent neural network based language model." in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, vol. 2, 2010, p. 3.

[49] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, \End-to-end continuous speech recognition using attention-based recurrent NN: First results," in *Proceedings of the Workshop on Deep Learning and Representation Learning*, 2014.

[50] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, \Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960{4964.

[51] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, \Towards end-to-end automatic code-switching speech recognition," *arXiv preprint arXiv:1810.12620*, 2018.

[52] D. Wang, X. Wang, and S. Lv, \An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.

[53] A. Graves, \Sequence transduction with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, 2012.

[54] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, \Exploring neural transducers for end-to-end speech recognition," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 206{213.

[55] D. C. Lyu, R. Y. Lyu, Y. C. Chiang, and C. N. Hsu, \Speech recognition on code-switching among the Chinese dialects," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2006.

[56] B. H. Ahmed and T.-P. Tan, \Automatic speech recognition of code switching speech using 1-best rescoring," in *Proceedings of the International Conference on Asian Language Processing (IALP)*, 2012, pp. 137{140.

[57] T. Lyudovyk and V. Pylypenko, \Code-switching speech recognition for closely related languages," in *Proceedings of the Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2014.

[58] H. Seki, S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, \An end-to-end language-tracking speech recognizer for mixed-language speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4919{4923.

[59] N. Luo, D. Jiang, S. Zhao, C. Gong, W. Zou, and X. Li, \Towards end-to-end code-switching speech recognition," *arXiv preprint arXiv:1810.13091*, 2018.

[60] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, \Investigating end-to-end speech recognition for Mandarin-English code-switching," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6056{6060.

[61] Z. Zeng, Y. Khassanov, V. T. Pham, H. Xu, E. S. Chng, and H. Li, \On the end-to-end solution to Mandarin-English code-switching speech recognition," in *Proceedings of the Interspeech, an Annual Conference of the International Speech Communication Association*, 2019.

[62] E. Boztepe, \Issues in code-switching: Competing theories and models," *Working Papers in TESOL & Applied Linguistics*, vol. 3, no. 2, 2005.

[63] P. C. Muysken, \Code-switching and grammatical theory," *The Handbook of Bilingualism*, pp. 283{311, 1995.

[64] C. Nilep, \`Code switching' in sociocultural linguistics," *Colorado Research in Linguistics*, vol. 19(1), pp. 1{22, 2006.

[65] L. Malik, *Socio-linguistics: A Study of Code-Switching*. Anmol Publications, 1994.

[66] L. Milroy and P. Muysken, *One Speaker, Two Languages: Cross-disciplinary Perspectives on Code-switching*. Cambridge University Press, 1995.

[67] H.-Y. Su, \Code-switching between Mandarin and Taiwanese in three telephone conversation: The negotiation of interpersonal relationships among bilingual speakers in Taiwan," in *Proceedings of the Symposium about Language and Society*, 2001.

[68] W. Craig, Y. Harel-Fisch, H. Fogel-Grinvald, S. Dostaler, J. Hetland, B. Simons-Morton, M. Molcho, M. G. de Mato, M. Overpeck, P. Due *et al.*, \A cross-national pro le of bullying and victimization among adolescents in 40 countries," *International Journal of Public Health*, vol. 54, no. 2, pp. 216{ 224, 2009.

[69] A. Dey and P. Fung, \A Hindi-English code-switching corpus." in *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2014, pp. 2410{2413.

[70] C. Myers-Scotton, \Social motivations for code-switching: evidence from Africa. Clarendon," 1993.

[71] T. Solorio and Y. Liu, \Part-of-speech tagging for English-Spanish code-switched text," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 1051{1060.

[72] I. Hamed, M. Elmahdy, and S. Abdennadher, \Building a rst language model for code-switch Arabic-English," *Procedia Computer Science*, vol. 117, pp. 208{216, 2017.

[73] D. Imseng, H. Bourlard, H. Caesar, P. N. Garner, G. Lecorve, and A. Nanchen, \MediaParl: Bilingual mixed language accented speech database," in *Proceedings of the Spoken Language Technology Workshop (SLT)*, 2012, pp. 263{268.

[74] E. Yilmaz, M. Andringa, S. Kingma, J. Dijkstra, F. Van der Kuip, H. Van de Velde, F. Kampstra, J. Algra, H. Heuvel, and D. Van Leeuwen, \A longitudinal bilingual Frisian-Dutch radio broadcast database designed for code-switching research," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2016.

[75] D.-C. Lyu, T.-P. Tan, E. S. Chng, and H. Li, \SEAME: A Mandarin-English code-switching speech corpus in South-East Asia," in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2010.

[76] D. C. Lyu and R. Y. Lyu, \Language identi cation on code-switching utterances using multiple cues," in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2008.

[77] H. Cao, P. Ching, T. Lee, and Y. T. Yeung, \Semantics-based language modeling for Cantonese-English code-mixing speech recognition," in *Proceedings of the 7th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2010, pp. 246{250.

[78] E. van der Westhuizen and T. Niesler, \A rst South African corpus of multilingual code-switched Soap Opera speech." in *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2018.

[79] A. C. Chandola, \Some linguistic in uences of English on Hindi," *Journal of Anthropological Linguistics*, pp. 9{13, 1963.

[80] S. Malhotra, \Hindi-English, code switching and language choice in urban, uppermiddle-class Indian families," *Kansas Working Papers in Linguistics*, 1980.

[81] A. Kumar, \Certain aspects of the form and functions of Hindi-English code-switching," *Journal of Anthropological Linguistics*, pp. 195{205, 1986.

[82] S. Sinha, \Code switching and code mixing among Oriya trilingual children { A study," *Journal on Language in India*, vol. 9(4), p. 274, 2009.

[83] J. MacSwan, \Code-switching and grammatical theory," *The Handbook of Bilingualism and Multilingualism*, pp. 321{350, 2012.

[84] C. Myers-Scotton, \Codeswitching with English: Types of switching, types of communities," *World Englishes*, vol. 8, no. 3, pp. 333{346, 1989.

[85] K. A. H. Zirker, \Intrasentential vs. intersentential code switching in early and late bilinguals," Master's thesis, Brigham Young University, 2007.

[86] C. F. Yeh, C. Y. Huang, L. C. Sun, C. Liang, and L. S. Lee, \An integrated framework for transcribing Mandarin-English code-mixed lectures with improved acoustic and language modeling," in *Proceedings of the 7th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2010, pp. 214{219.

[87] M. Dhar, V. Kumar, and M. Shrivastava, \Enabling code-mixed translation: Parallel corpus creation and MT augmentation approach," in *Proceedings of the 1st Workshop on Linguistic Resources for Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 131{140.

[88] H. Adel, K. Kirchho , D. Telaar, N. T. Vu, T. Schlippe, and T. Schultz, \Features for factored language models for code-switching speech," in *Proceedings of the Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2014.

[89] 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 Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[90] S. Garg, T. Parekh, and P. Jyothi, \Dual language models for code switched speech recognition," in *Proceedings of the Interspeech, an Annual Conference of the International Speech Communication Association*, 2018.

[91] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury, \POS tagging of English-Hindi code-mixed social media content," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 974{979.

[92] K. Bhuvanagirir and S. K. Kopparapu, \Mixed language speech recognition without explicit identi- cation of language," *American Journal of Signal Processing*, vol. 2, no. 5, pp. 92{97, 2012.

[93] D.-C. Lyu, E.-S. Chng, and H. Li, \Language diarization for code-switch conversational speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7314{7318.

[94] S. Sitaram and A. W. Black, \Speech synthesis of code-mixed text." in *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2016.

[95] J. Y. Chan, P. Ching, T. Lee, and H. M. Meng, \Detection of language boundary in code-switching utterances by bi-phone probabilities," in *Proceedings of the International Symposium on Chinese Spoken Language Processing*, 2004, pp. 293{296.

[96] S. Watanabe, T. Hori, and J. R. Hershey, \Language independent end-to-end architecture for joint language identi cation and speech recognition," in *Proceeding of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017, pp. 265{271.

[97] K. Li, J. Li, G. Ye, R. Zhao, and Y. Gong, \Towards code-switching ASR for end-to-end CTC models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6076{6080.

[98] D.-C. Lyu, R.-Y. Lyu, Y.-c. Chiang, and C.-N. Hsu, \Speech recognition on code-switching among the Chinese dialects," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2006.

[99] J. C. Franco and T. Solorio, \Baby-steps towards building a Spanglish language model," in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2007, pp. 75{84.

[100] N. T. Vu, D.-C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, \A rst speech recognition system for Mandarin-English code-switch conversational speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4889{4892.

[101] H.-P. Shen, C.-H. Wu, Y.-T. Yang, and C.-S. Hsu, \Cecos: A Chinese-English code-switching speech database," in *Proceedings of the International Conference on Speech Database and Assessments (Oriental COCOSDA)*, 2011, pp. 120{123.

[102] T. I. Modipa, M. H. Davel, and F. De Wet, \Implications of Sepedi/English code switching for ASR systems," in *Proceedings of the 24th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, 2013.

[103] D. Amazouz, M. Adda-Decker, and L. Lamel, \The French-Algerian Code-Switching Triggered audio corpus (FACST)." in *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2018.

[104] I. Hamed, M. Elmahdy, and S. Abdennadher, \Collection and analysis of code-switch Egyptian Arabic-English speech corpus." in *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2018.

[105] S. V. G. Ayushi Pandey, Brij Mohan lal, \Adapting monolingual resources for code-mixed hindi-english speech recognition," in *Proceedings of the 21st International Conference on Asian Language Processing (IALP)*, 2017.

[106] S. Sivasankaran, B. M. L. Srivastava, S. Sitaram, K. Bali, and M. Choudhury, \Phone merging for code-switched speech recognition," in *Proceedings of the 3rd Workshop on Computational Approaches to Linguistic Code-switching*, 2018.

[107] J. E. Flege, \Second-language speech learning: Theory, ndings, and problems," *Speech Perception and Linguistic Experience*, pp. 233{272, 1995.

[108] S. Shahnawazuddin, D. Thotappa, A. Dey, S. Imani, S. Prasanna, and R. Sinha, \Improvements in IITG Assamese spoken query system: Background noise suppression and alternate acoustic modeling," *Journal of Signal Processing Systems*, vol. 88, no. 1, pp. 91{102, 2017.

[109] B. Ramani, S. L. Christina, G. A. Rachel, V. S. Solomi, M. K. Nandwana, A. Prakash, S. A. Shanmugam, R. Krishnan, S. K. Prahalad, K. Samudravijaya, P. Vijayalakshmi, T. Nagarajan, and H. A. Murthy, \A common attribute based uni ed HTS framework for speech synthesis in Indian languages," in *Proceedings of the 8th ISCA Workshop on Speech Synthesis*, 2013.

[110] S. Ganji, K. Dhawan, and R. Sinha, \Novel textual features for language modeling of intra-sentential code-switching data," *Journal of Computer Speech & Language*, vol. 65, p. 101099, 2020.

[111] | | , \IITG-HingCoS corpus: A Hinglish code-switching database for automatic speech recognition," *Journal of Speech Communication*, vol. 110, pp. 76{89, 2019.

[112] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, \Front-end factor analysis for speaker veri cation," *Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788{798, 2011.

[113] S. Garimella, A. Mandal, N. Strom, B. Ho meister, S. Matsoukas, and S. H. K. Parthasarathi, \Robust i-vector based adaptation of DNN acoustic model for speech recognition," in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2015.

[114] R. Haeb-Umbach and H. Ney, \Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 1992, pp. 13{16.

[115] M. J. Gales *et al.*, \Maximum likelihood linear transformations for HMM-based speech recognition," *Journal of Computer Speech & Language*, vol. 12, no. 2, pp. 75{98, 1998.

[116] C. J. Leggetter and P. C. Woodland, \Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Journal of Computer Speech & Language*, vol. 9, no. 2, pp. 171{185, 1995.

[117] T. Anastasakos, J. McDonough, and J. Makhoul, \Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 1997, pp. 1043{1046.

[118] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Kara at, A. Rastrow *et al.*, \Subspace Gaussian mixture models for speech recognition," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 4330{4333.

[119] V. Peddinti, D. Povey, and S. Khudanpur, \A time delay neural network architecture for e cient modeling of long temporal contexts," in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2015.

[120] M. Federico, N. Bertoldi, and M. Cettolo, \IRSTLM: an open source toolkit for handling large scale language models," in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2008.

[121] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, \RNNLM: Recurrent neural network language modeling toolkit," in *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 196{201.

[122] M. Song, Y. Zhao, and S. Wang, \Exploiting di erent word clusterings for class-based RNN language modeling in speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5735{5739.

[123] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, \Extensions of recurrent neural network language model," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5528{5531.

[124] Y. Bengio, P. Simard, and P. Frasconi, \Learning long-term dependencies with gradient descent is di cult," *Transactions on Neural Networks*, vol. 5, no. 2, pp. 157{166, 1994.

[125] D. Bahdanau, K. Cho, and Y. Bengio, \Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, 2015.

[126] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, \A comparison of sequence-to-sequence models for speech recognition." in *Proceedings of the Interspeech, an Annual Conference of International Speech Communication Association*, 2017, pp. 939{943.

[127] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, \Connectionist temporal classi cation: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369{376.

[128] A. Graves, N. Jaitly, and A.-R. Mohamed, \Hybrid speech recognition with deep bidirectional LSTM," in *Proceedings of the Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273{278.

[129] S. Tong, P. N. Garner, and H. Bourlard, \An investigation of multilingual ASR using end-to-end LF-MMI," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6061{6065.

[130] B. M. L. Srivastava and S. Sitaram, \Homophone identi cation and merging for code-switched speech recognition," in *Proceedings of the Interspeech, an Annual Conference of the International Speech Communication Association*, 2018, pp. 1943{1947.

[131] K. Demuynck, T. Laureys, D. V. Compernolle, and H. V. Hamme, \Flavor: A exible architecture for LVCSR," in *Proceedings of the European Conference on Speech Communication and Technology*, 2003.

[132] K. Demuynck, D. Van Compernolle *et al.*, \Robust phone lattice decoding," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, vol. 4, 2006, pp. 1622{1625.

[133] G. Zweig and J. Nedel, \Empirical properties of multilingual phone-to-word transduction," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4445{4448.

[134] J. Y. Chan, P. Ching, and T. Lee, \Development of a Cantonese-English code-mixing speech corpus," in *Proceedings of the 9th European Conference on Speech Communication and Technology*, 2005.

[135] G. Sreeram and R. Sinha, \Exploiting parts-of-speech for improved textual modeling of code-switching data," in *Proceedings of the 24th National Conference on Communications (NCC)*, 2018, pp. 1{6.

[136] | |, \A novel approach for e ective recognition of the code-switched data on monolingual language model." in *Proceedings of the Interspeech, an Annual Conference of the International Speech Communication Association*, 2018, pp. 1953{1957.

[137] J. Gebhardt, \Speech recognition on English-Mandarin code-switching data using factored language models," Master's thesis, Department of Informatics, Karlsruhe Institute of Technology, 2011.

[138] B. Harb, C. Chelba, J. Dean, and S. Ghemawat, \Back-o language model compression," in *Proceedings of the Interspeech, an Annual Conference of the International Speech Communication Association*, 2009.

[139] A. Axelrod, \Factored language models for statistical machine translation," Master's thesis, Division of Informatics University of Edinburgh, 2006.

[140] K. Duh and K. Kirchho , \Automatic learning of language model structure," in *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, 2004.

[141] I. Oparin, M. Sundermeyer, H. Ney, and J. L. Gauvain, \Performance analysis of neural networks in combination with n-gram language models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5005{5008.

[142] T. Mikolov, K. Chen, G. Corrado, and J. Dean, \E cient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[143] X. Chen, X. Liu, Y. Qian, M. Gales, and P. C. Woodland, \CUED-RNNLM{An open-source toolkit for e cient training and evaluation of recurrent neural network language models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6000{6004.

[144] S. Ganji and R. Sinha, \Exploring recurrent neural network based acoustic and linguistic modeling for children's speech recognition," in *Proceedings of the TENCON, an International Conference of IEEE Region 10*, 2017.

[145] Y. Wu, X. Lu, H. Yamamoto, S. Matsuda, C. Hori, and H. Kashioka, \Factored language model based on recurrent neural network," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2012, pp. 2835{2850.

[146] H. Adel, N. T. Vu, F. Kraus, T. Schlippe, H. Li, and T. Schultz, \Recurrent neural network language modeling for code switching conversational speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8411{8415.

[147] Y. Shi, P. Wiggers, and C. M. Jonker, \Towards recurrent neural networks language models with linguistic and contextual features," in *Proceedings of the Interspeech, an Annual Conference of the International Speech Communication Association*, 2012.

[148] M. Schuster and K. K. Paliwal, \Bidirectional recurrent neural networks," *Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673{2681, 1997.

[149] J. Niehues, T.-L. Ha, E. Cho, and A. Waibel, \Using factored word representation in neural network language models," in *Proceedings of the 1st Conference on Machine Translation*, vol. 1, 2016, pp. 74{82.

[150] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, \Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the Conference on Computational Linguistics on Human Language Technology*, vol. 1, 2003, pp. 173{180.

[151] A. Bharati, R. Sangal, D. M. Sharma, and L. Bai, \Anncorra: Annotating corpora guidelines for POS and chunk annotation for Indian languages," *Language Technologies Research Centre (LTRC), Hyderabad*, 2006.

[152] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, \Building a large annotated corpus of English: The Penn Treebank," *Journal of Computational Linguistics*, vol. 19, no. 2, pp. 313{330, 1993.

[153] S. Reddy and S. Sharo , \Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources," in *Proceedings of the 5th International Workshop on Cross Lingual Information Access*, 2011, pp. 11{19.

[154] K. Kirchho , J. Bilmes, and K. Duh, \Factored Language Models Tutorial," Dept of EE, University of Washington, Tech. Rep. UWEETR-2007-0003, 2007.

[155] A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, \The IIT Bombay English-Hindi parallel corpus," *arXiv preprint arXiv:1710.02855*, 2017.

[156] A. Stolcke, \SRILM: An extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, vol. 2, 2002, pp. 901{904.

[157] D. B. Paul and J. M. Baker, \The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the Workshop on Speech and Natural Language*, 1992, pp. 357{362.

[158] S. Zhang, J. Yi, Z. Tian, J. Tao, and Y. Bai, \RNN-transducer with language bias for end-to-end Mandarin-English code-switching speech recognition," *ArXiv*, vol. abs/2002.08126, 2020.

[159] M. Mohri, F. Pereira, and M. Riley, \Weighted nite-state transducers in speech recognition," *Journal of Computer Speech & Language*, vol. 16, no. 1, p. 6988, 2002.

[160] R. Collobert, A. Hannun, and G. Synnaeve, \A fully di erentiable beam search decoder," *arXiv preprint arXiv:1902.06022*, 2019.

[161] S. Ganji, K. Dhawan, K. Priyadarshi, and R. Sinha, \Joint language identi cation of code-switching speech using attention-based E2E network," in *Proceedings of the International Conference on Signal Processing and Communications (SPCOM)*, 2020.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

List of Publications Related to Thesis

Journal Publications

1. **Ganji Sreeram** and R. Sinha, "Exploration of End-to-End Framework for Code-Switching Speech Recognition Task: Challenges and Enhancements," *IEEE Access*, vol. 8, pp. 68146-68157, 2020.

2. **Ganji Sreeram**, Kunal Dhawan and R. Sinha, "Novel Textual Features for Language Modeling of Intra-Sentential Code-Switching Data," *Elsevier Journal on Computer Speech and Language*, vol. 64, p. 101099, 2020.

3. **Ganji Sreeram**, Kunal Dhawan and R. Sinha, "IITG-HingCoS corpus: A Hinglish code-switching database for automatic speech recognition," *Elsevier Journal on Speech Communication*, , vol. 110, pp. 76-89, 2019.

Conference Publications

1. **Ganji Sreeram**, Kunal Dhawan, Kumar Priyadarshi, and R. Sinha, "Joint Language Identification of Code-Switching Speech using Attention based E2E Network," in *Proceedings of the International Conference on Signal Processing and Communications (SPCOM)*, 2020.

2. Kunal Dhawan, **Ganji Sreeram**, Kumar Priyadarshi, and R. Sinha, "Investigating Target Set Reduction for End-to-End Speech Recognition of Hindi-English Code-Switching Data," in *Proceedings of the National Conference on Communications (NCC)*, 2020.

3. **Ganji Sreeram**, and R. Sinha, "Exploiting Parts-of-Speech for Improved Textual Modeling of Code-Switching Data," in *Proceedings of the National Conference on Communications (NCC)*, 2018.

4. **Ganji Sreeram**, and R. Sinha, "A Novel Approach for Effective Recognition of the Code-Switched Data on Monolingual Language Model," in *Proceedings of the Interspeech, Annual Conference of the International Speech Communication Association*, 2018.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

List of Other Publications

1. Nagendra Kumar, **Ganji Sreeram**, and R. Sinha, "Exploring Dictionary Diversity for Improved Sparse Coding Based Speaker Verification," in *Proceedings of the Annual IEEE India Conference (INDICON)*, 2017.

2. **Ganji Sreeram**, and R. Sinha, "Exploring Recurrent Neural Network based Acoustic and Linguistic Modeling for Children's Speech Recognition," in *Proceedings of the International Technical Conference of IEEE Region 10 (TENCON)*, 2017.

3. **Ganji Sreeram**, and R. Sinha, "Semi-Coupled Dictionary Based Automatic Bandwidth Extension Approach for Enhancing Children's ASR," in *Proceedings of the Interspeech, Annual Conference of the International Speech Communication Association*, 2016.

4. **Ganji Sreeram**, and R. Sinha, "Improved Speaker Verification using Block Sparse Coding over Joint Speaker-Channel Learned Dictionary," in *Proceedings of the International Technical Conference of IEEE Region 10 (TENCON)*, 2015.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati