THE UNIVERSITY *of* EDINBURGH

Understanding and Modeling Code-Switching: Metrics, Triggers, and Applications in Multilingual NLP

Jie Chi

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2025

Abstract

Code-switching, the phenomenon of alternating between two or more languages within a single conversation or discourse, has been commonly observed in the growing context of multilingual communities. Decades of research across various disciplines have focused on understanding its underlying principles and modeling its patterns. This doctoral thesis contributes to this ongoing research from both theoretical and applied perspectives.

Firstly, existing popular metrics for measuring code-switching richness rely on counts of tokens, switching points, or the distribution of language spans without considering differences in morpho-syntax and orthographic conventions across languages. Consequently, metrics calculated for different language pairs are not comparable. This thesis proposes a framework that leverages linguistic findings as makeshift ground truths to assess the quality and sufficiency of existing metrics after normalizing them to factor out token differences. Additionally, it introduces the T-index, which utilizes machine translation systems to capture properties of code-switched words in relation to the participating language pairs.

Building on the existing hypothesis that part-of-speech (POS) facilitates code-switching occurrence, this thesis extends prior work by incorporating the impact of word positions and robustly confirms a statistically significant connection between POS and code-switching. The findings suggest that more diverse syntactical structures lead to less flexibility in code-switching. By categorizing code-switched words and investigating neighboring POS, we observe that this relationship is strongest in close proximity to switched instances, gradually diminishing as words move farther from code-switching points.

Furthermore, this thesis investigates two approaches to code-switched text generation and their applications in improving automatic speech recognition systems. Using parallel data from two languages, equivalence constraint theory can determine which segments can be replaced to produce code-switched sentences. Alternatively, a multilingual machine translation system can achieve similar results by using shared representations between languages to produce lexical replacements.

In conclusion, this research enhances the understanding of code-switching phenomena and its applications in natural language processing and speech recognition technologies. By bridging linguistic theory and computational methods, this thesis aims to offer valuable insights and practical solutions for handling code-switching in multilingual environments.

Lay summary

This thesis explores the phenomenon of code-switching, which occurs when people switch between two or more languages during a conversation. This is a common behavior in multilingual communities around the world. Researchers have been studying code-switching for many years to understand why it happens and how it works. Our research contributes to this ongoing effort by examining both the theoretical and practical aspects of code-switching.

One of the challenges in studying code-switching is figuring out how to accurately describe its style - how one individual's way of code-switching might differ from another's. Many existing methods focus on counting words or identifying where the language switch happens, but they often don't consider the differences in how languages are structured or written. This makes it difficult to compare results across different languages. Our research introduces a new framework that adjusts for these differences, making comparisons more reliable. We also developed a new measure called the T-index, which uses machine translation to better understand how code-switched words function in different languages.

Another part of our research looks at how the type of word, such as a noun or verb, influences where code-switching occurs. With analysis of both Spanish-English and Mandarin-English corpora, we confirmed that certain types of words are more likely to be involved in a language switch, especially when they are close to where the switch happens. This finding is particularly interesting when the languages involved have different grammatical structures.

Finally, we looked into two methods for creating code-switched text, which can help improve technologies like automatic speech recognition systems. One method uses rules based on linguistic theory to decide where language switches can happen in a sentence, while the other relies on a machine translation system to create realistic code-switched sentences by making use of the similarities between languages.

In conclusion, we not only seek to deepen our theoretical understanding of code-switching, including how to describe its specific characteristics and assess the influence of different types of words on code-switching, but also offer practical approaches for generating natural code-switched text to enhance multilingual technologies. This thesis contributes to the development of more robust language processing systems that can better serve our globalized world.

Acknowledgements

I am deeply grateful to Peter Bell and Catherine Lai for being my supervisors. Your guidance, insightful conversations, and detailed feedback on my writings have been invaluable throughout my research journey. I especially appreciate your patience in adjusting your supervision to suit my progress and preferences, making this experience more fulfilling.

I would also like to extend my sincere thanks to my colleagues in the CSTR community, including those who have moved on. Steve Renals introduced me to the world of ASR, while Sam Ribeiro and Aciel Eshky mentored me in my first speech recognition project. Collaborations with Electra Wellington and Ondrej Klejch, along with informal chats with other colleagues, have greatly enriched my time here.

I am also incredibly thankful to my colleagues in CSTR communities, including those who have already left. Steve Renals showed the the world of ASR to me, Sam Ribeiro and Aciel Eshky for mentoring me in my first speech recognition project. The collaboration with Electra Wellington and Ondrej and informal chats with other colleagues have also enriched my experience.

I would also like to acknowledge my colleagues in the CDT programme, who helped me navigate the early stages of my research and supported me during the challenging times of the COVID-19 pandemic. Special thanks to Rimvydas Rubavicius for the countless walks during lockdowns, and to Nikita Moghe, Dan Wells, and others for the occasional but impactful chats in the Forum. The influence of these moments on my mental well-being cannot be overstated.

I am incredibly thankful to my parents and friends for their unwavering support and understanding, particularly for resisting the urge to constantly ask when I would finish. Their patience and belief in me have meant a lot. A special thanks to Jialin Yu, Yixuan Zhang, and Jojo Ju for taking care of my cats while I was away.

I am also grateful to the colleagues I met during the JSALT workshop, whose contributions and friendship have broadened my perspective and enhanced my work. My time in Copenhagen during my internship at Apple was also enriched by the wonderful people I met there. Special thanks to Natalie for her mentorship, and to Rita and Eva for the city walks and the internal jokes that made the office feel like home.

Lastly, to my partner Danyi Liu, thank you for being my constant source of comfort and joy. Your companionship has kept me grounded throughout this process. And to our three cats, Coco, Snow, and Tina: thank you for keeping me company by lying on my desk, especially during the late nights of writing and research.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Jie Chi*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Bilingual[1] speakers have outnumbered monolingual ones, particularly when considering bilingualism as the ability to use multiple languages sufficiently for limited casual conversation, as discussed in (Myers-Scotton, 2006). In the growing context of bilingual communities, mixed language usage is commonly observed in daily life. For example, in multilingual countries like Indonesia, people from the president to children employ English phrases in their daily activities, such as

(1) Gue tadi lunch meeting samabos. [2]
I had just lunch meeting **with boss.** (Sahib et al., 2021)

Similarly, people in the Middle East often incorporate English phrases into their conversations. In multilingual educational settings, many individuals have taken language courses or been required to learn a foreign language during their formal education. As a result, they may mix familiar phrases from their first language with the newly acquired language, especially when their proficiency in the latter is still developing.

This phenomenon of mixing languages is commonly referred to as code-switching, though its definition varies widely across the literature. For example, Milroy and Muysken (1995) describes code-switching as the alternating use of two or more *lan-*

[1] We use the term *bilingual* throughout this thesis to refer to speakers using two or more languages, including those who can also be called *multilingual*

[2] Throughout the paper, all code-switched examples are presented with the original code-switched sentence on top and the monolingual translation (usually in English) below. The underscored words or phrases are the paired words in the other language within the code-switched sentence and its translation (in bold).

guages by bilinguals within the same conversation, which is a general term covering all different bilingual behaviors. Some researchers emphasize that switching can occur not only between languages but also *dialects of the same language* (Roche, 1993; Myers-Scotton, 1993). There are also less popular definitions from the perspective of second language acquisition. Simensen (1998) views code-switching as a strategy for overcoming communication challenges in a second language, where a speaker borrows words from their *first* language. Cook (2016) defines code-switching as the act of switching between languages during a conversation when both speakers are *proficient* in the same two languages. A more comprehensive definition provided by McKay (2002) describes code-switching as the act of switching between languages or language varieties by a speaker or *writer*. This definition is broader than the previous ones, as it includes shifts within a sentence, at sentence boundaries, and even across *different speakers* within the same conversation.

Although different language pairs may present varying extents or types of code-switching, they can generally be categorised as inter-sentential, intra-sentential, and tag switching, where respectively the language switches at sentence or clause boundary; within the sentence or clause; or by inserting a tag phrase (Myers-Scotton, 1993; Muysken, 2011; Hoffmann, 2014). While Chapter 2 provides a review of these diverse types, this thesis concentrates on a specific type *intra-sentential code-switching*. This type of code-switching is particularly challenging because the variation of mixed languages within a single sentence is often more complex than across sentences.

Code-switching is of great interest from both psycholinguistic and practical perspectives. On one hand, it reflects the flexibility and adaptability of individuals in their language use. It reflects the simultaneous activation of multiple language systems in the brain, known as language co-activation, where both languages are readily accessible and can be drawn upon as needed (Gardner-Chloros, 2009). Therefore, code-switching serves as a testing ground for research into the cognitive mechanisms of bilingual language production and studies emerging from this exploration have shown it involves multiple layers of linguistic processing and is influenced by the properties of the words, linguistic structures and socio-interactional considerations (Gardner-Chloros, 2009; Kootstra et al., 2020). Given its complexity, researchers have sought to quantify the richness of code-switching to better compare its patterns across different datasets (Barnett et al., 2000; Guzman et al., 2016; Srivastava and Singh, 2021; Guzmán et al., 2017). For instance, by measuring code-switching richness, researchers can determine whether a bilingual dataset from a formal interview setting

exhibits more or less frequent code-switching compared to spontaneous conversations. Such comparisons are valuable for tailoring models to specific contexts, like improving the accuracy of speech recognition systems designed to handle informal, code-switched dialogue. Another fundamental question investigated in linguistics is what factors influence bilinguals' tendency to code-switch. One prominent theory, the lexical triggering hypothesis, suggests that language-ambiguous words, such as cognates [3] facilitate the occurrence of code-switching. For example, *automobile* and *automóvil* are English-Spanish cognates, making it more likely for bilingual speakers to switch languages when these words appear. In the sentence *I bought an automobile yesterday*, the cognate *automobile* might trigger a switch to Spanish, resulting in a code-switched sentence like *I bought an automóvil yesterday*. In contrast, a sentence like *I bought a book yesterday*, without a cognate, is less likely to prompt a switch (Clyne, 1980; Broersma, 2009; Broersma et al., 2020).

On the other hand, the growing interest in Natural Language Processing (NLP) techniques within bilingual settings has led to a significant focus on code-switched NLP. To accurately capture the language use of bilingual speakers, any model designed for bilingual contexts should be capable of accommodating code-switching. As technology becomes more embedded in everyday life, the development of code-switched NLP is crucial for creating user-friendly interfaces and virtual assistants capable of understanding and responding to code-switched queries. For example, Agarwal et al. (2017) found a link between language use and sentiment, highlighting that neglecting one language or disregarding code-switching can lead to inaccurate assessments of user sentiment. As a result, code-switched NLP models are also essential for enhancing cultural sensitivity in language processing applications.

Motivated by the same interest, this thesis addresses three key issues related to code-switching. First, existing metrics (Barnett et al., 2000; Guzman et al., 2016; Srivastava and Singh, 2021; Guzmán et al., 2017) simply rely on simple counts of tokens, switching points, or the distribution of participating languages, making it difficult to interpret their effectiveness in identifying and comparing code-switching styles across different datasets. To tackle this, we introduce a method of analyzing code-switching richness metrics that leverages linguistic findings as makeshift ground-truths. We propose assessing these metrics by evaluating how well their outputs align with linguistic research, such as the observation that code-switching tends to be more complex in conversational settings than in formal or technical ones. Additionally, we introduce a

[3] Translations that overlap in phonology (and often also in orthography) across languages,

new metric, the **T-index**, which incorporates the lexical semantic properties of code-switching words and is specific to the language pair, providing a valuable addition to existing code-switching richness metrics.

Second, previous studies (Clyne, 1967; Broersma and De Bot, 2006) investigating the triggers for code-switching have primarily focused on cognates. However, not all language pairs possess cognates, and identifying them requires linguistic expertise. Given that most code-switching triggers are nouns and proper nouns (Broersma and De Bot, 2006), the role of part-of-speech in identifying the constraints of code-switching has garnered attention from researchers (Soto et al., 2018). Drawing inspiration from these studies, we hypothesize that the dependency between part-of-speech and code-switching remains significant when considering the distribution of both part-of-speech and code-switching across word positions. Our findings suggest that this relationship diminishes as the part-of-speech moves further from the code-switching points.

Finally, the lack of code-switching data is a major hindrance to the development of code-switched systems, such as language models or Automatic Speech Recognition (ASR) systems. Efforts have been made to address this challenge by exploring techniques for generating synthetic code-switched sentences to augment the text. As bilingual speakers can choose languages based on text and social dynamics, ongoing debates surround whether code-switching is a rule-governed or random behavior. (Hymes, 1971) previously concluded that *no one has been able to show that such rapid alternation is governed by any systematic rules or constraints.* Nevertheless, after decades of research, there is now a consensus that code-switching is not purely idiosyncratic; rather, it occurs at specific switch points (Boztepe, 2003). In this context, we apply *Equivalence Constraint Theory* (ECT) (Poplack, 1980), to determine the code-switching points and generate code-switched sentences from parallel sentences. To assess the quality of the generated sentences, we introduce a pipeline designed to examine the efficacy of the augmentation techniques on language models and ASR systems. Then, to further reduce the reliance on linguistic knowledge, we propose a machine-learning-based approach which leverages the emergence of shared representations in pretrained encoder-decoder models to generate texts.

1.2 Aim and Objectives

This thesis aims to advance our understanding of code-switching phenomena, especially intra-sentential code-switching, and contribute to enhancing the capabilities of models in bilingual contexts. The specific objectives guiding this research are outlined below:

Our initial objective involves presenting a comprehensive overview of the advancements in addressing both theoretical and practical questions within the study of code-switching. This includes a thorough examination of varying perspectives on crucial issues, such as definitions, triggers and NLP models, with the goal of offering insight into the multifaceted nature of code-switching and thoroughly clarifying the scope of this thesis.

Then, as the properties of words have been observed to have an influence on bilingual production in the literature, we hypothesize that part-of-speech, a universal feature across all languages, also plays a significant role in facilitating code-switching occurrences across language pairs. Given the unique status of English as a global language and its mandatory inclusion as a foreign language in numerous countries, we conduct experiments using datasets featuring Spanish-English (same language family) and Mandarin-English (different language family).

Next, we propose a framework for generating code-switched sentences from parallel data and evaluating the generation using ASR systems. Based on equivalence constraint theory, we use constituency parsers on parallel data to determine the equivalent constituents. Then after identifying the possible code-switching points, we concatenate segments from the parallel pair to generate code-switched sentences.

Similarly, instead of relying on linguistic theory, we explore if a pre-trained machine translation system without explicit exposure to code-switched data can generate code-switched sentences like a bilingual speaker.

Through these objectives, this thesis seeks to provide a deeper understanding of code-switching and develop practical approaches to improve multilingual NLP applications.

1.3 Thesis Statement

This thesis argues that intra-sentential code-switching exhibits structured and predictable patterns that can be captured through syntactic, lexical, and distributional

features. By incorporating these structural cues into the modeling and generation of code-switched data, we can improve the effectiveness of downstream multilingual NLP applications, particularly ASR. To support this claim, the thesis proceeds through a series of empirical investigations: Chapter 3 evaluates code-switching richness metrics; Chapter 4 analyzes part-of-speech distributions near switching points; Chapter 5 introduces a syntax-aware code-switched data generation pipeline using ECT; and Chapter 6 explores neural machine translation as a structure-sensitive generation approach, which implicitly learn syntactic correspondences between source and target languages. Together, these chapters demonstrate that code-switching is not random, but rule-governed and sensitive to syntactic and lexical context, and that leveraging these patterns improves both theoretical understanding and practical model performance.

1.4 Thesis Outline

The rest of the thesis is structured as follows:

Chapter 2 lays the foundation for the thesis by introducing the background and key related work on code-switching. It covers the definition of code-switching, its various types, and explores the progress made in characterizing different code-switching styles and identifying potential triggering factors. Additionally, the chapter provides an overview of advancements in code-switched NLP. This background is essential for understanding the subsequent chapters of the thesis.

Chapter 3 illustrates the diversity of this phenomenon in conversational and formal settings. We assess both existing and proposed new code-switching richness metrics by considering the extent to which their outputs align with linguistic research. In the end, we conclude that relying on any single measure in isolation for a complete characterization of code-switching is not feasible.

Chapter 4 presents approaches exploring the trigger of code-switching by analyzing the relationship between POS and code-switching. With the use of statistical tests, we not only affirm the existence of a statistically significant connection between POS and the likelihood of code-switching across language pairs, but notably find this relationship exhibits its maximum strength in proximity to switched instances, progressively diminishing as tokens distance themselves from these switching points.

Chapter 5 presents a framework for code-switched ASR task. By using phonological features for phoneme mapping, and POS tags and equivalence constraint theory for more natural code-switched text generation, we eventually achieved 2% improvement

in perplexity (PPL) as well as word error rate (WER).

Chapter 6 introduces a novel approach of forcing a multilingual MT system that was trained on non-code-switched data to generate code-switched translations. Comparing against two prior methods, we show that simply leveraging the shared representations of two languages (Mandarin and English) yields better code-switched text generation and, ultimately, better code-switched ASR.

Chapter 7 summarizes the thesis, addressing its limitations and discussing potential future research directions.

1.5 Publication

Chapter 3 is based on (Chi et al., 2024), published in the proceedings of InterSpeech 2024. This work was completed with equal contribution from Electra Wallington, who collaboratively conducted the baseline experiments, under the supervision of Peter Bell.

Chapter 4 is based on (Chi and Bell, 2024), published in the findings of the 18th Conference of the European Chapter of the Association for Computational Linguistics. This work was completed under the supervision of Peter Bell.

Chapter 5 is based on (Chi and Bell, 2022), published in the Proceedings of the 29th International Conference on Computational Linguistics. This work was completed under the supervision of Peter Bell.

Chapter 6 is based on (Chi et al., 2023), published in the proceedings of InterSpeech 2023. This work was completed with equal contribution from Brian Lu, who collaboratively designed the architecture and conducted the experiments, and under the supervision of Jason Eisner, Peter Bell, Preethi Jyothi, and Ahmed M. Ali.

Chapter 2

Background

This chapter presents the fundamental concepts necessary to understand the remainder of the thesis. Initially, we introduce the definition of code-switching and its various types, illustrating their behavior across different languages and domains. Subsequently, we discuss current research on characterizing code-switching styles and potential triggering factors. In the end, we provide an overview of the progression in NLP tasks involving code-switching.

2.1 Introduction to Code-Switching

2.1.1 Dynamics of Code-Switching and Borrowing

In this thesis, we broadly define code-switching as the active use of mixed languages within the same sentence. We argue that distinguishing code-switching from loanwords is crucial for understanding bilingual language practices, as both can appear similar on the surface. Research on language contact provides extensive insights into how lexical items from one language can enter another during language maintenance scenarios—often driven by immigration, trade, or military invasions involving a dominant group and a linguistic minority (Alvanoudi, 2018). In such scenarios, the minority group maintains its native language while undergoing various changes induced by contact with another language. Code-switching emerges as a fundamental mechanism through which forms and constructions from the source or donor language are integrated into the recipient or borrowing language (Thomason, 2001; Gardner-Chloros, 2008).

Borrowing represents a more permanent and stabilized outcome of language con-

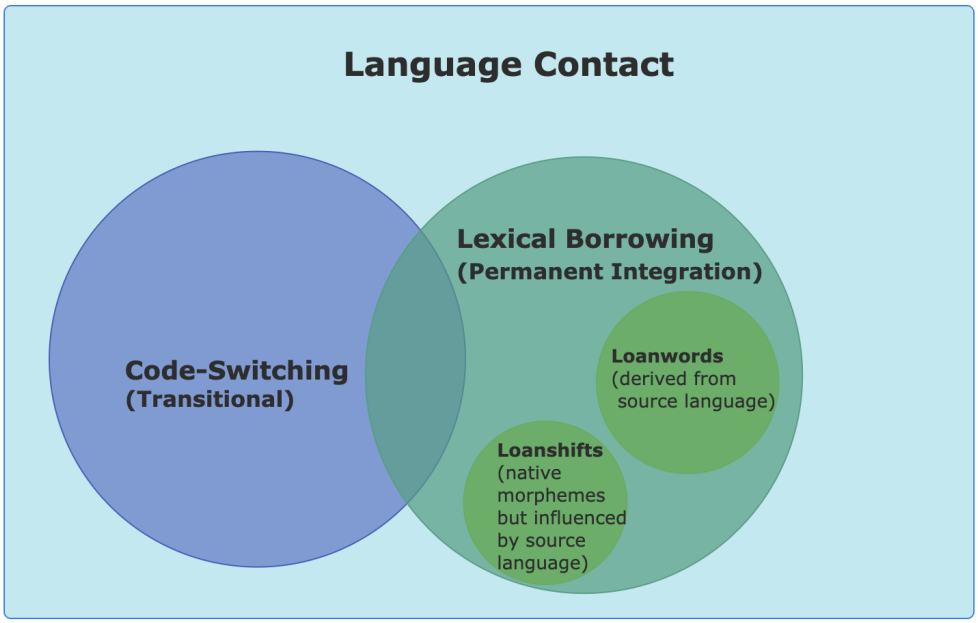


Figure 2.1: The relationships between the concepts.

tact, where foreign features become fully integrated into the recipient language (Alvanoudi, 2018). Borrowed elements may include a wide range of elements, from lexemes, pronouns, affixes, nominal categories, verbal categories, and syntactic features, to phonemes, habits of pronunciation, intonation patterns, and ways of framing discourse (Aikhenvald, 2007). Within the process of borrowing, lexical borrowing specifically refers to the transfer of lexical material and it can be categorized into two prominent types: loanwords and loanshifts (Winford, 2003). Loanwords are lexical items where all or part of the morphemic composition is derived from the source language while loanshifts are lexical items that use native morphemes but have meanings influenced by the source language. In contrast, code-switching remains a transient and situational *contact-induced speech behavior*, characterized by the seamless and often subconscious alternation between languages within a conversation or even a single sentence (Haspelmath, 2009). It is prevalent in the everyday interactions of bilinguals, showcasing their ability to navigate and negotiate multiple linguistic systems simultaneously. The relationship between code-switching, lexical borrowing, loanwords and loanshifts is shown in Figure 2.1

There is no general consensus on the relationship between lexical borrowing and code-switching. A common operational definition holds that a lexical item is considered borrowed if it is used by speakers who do not know the donor language, often without awareness of its foreign origin. In contrast, a code-switched word is typically

used by a bilingual speaker with active access to both languages. Many researchers posit that lexical borrowing is a gradual process, where lexical items are initially code-switched—meaning they are directly incorporated from the donor language—and, over time, become so widely used that they eventually blend into the recipient language as bona fide loanwords. In contrast, some argue that linguistic integration is more abrupt than gradual (Poplack and Dion, 2012). According to this perspective, code-switches do not necessarily evolve into borrowings; rather, the decision to code-switch or borrow occurs at the moment the other-language item is accessed. Despite these differing views, there is a general agreement that distinguishing between code-switched and borrowed words is challenging. Some believe that previous attempts to differentiate them have failed due to methodological limitations (Winford, 2003; Thomason, 2001; Gardner-Chloros, 2009), while others argue that the distinction is inherently fuzzy. They suggest that code-switching and borrowing might represent different points on the same continuum or be manifestations of the same underlying process (Haspelmath, 2009; Isurin et al., 2009; Poplack and Dion, 2012).

Systematic empirical research might appear as a promising solution to address these questions, yet progress in the field has been limited. One of the reasons is the nature of language contact data. Established loanwords, which have been fully integrated into the recipient language and can be recognized by dictionaries, often do not reveal how they came to be adopted. Observations usually only show how languages mix at a specific moment, and the absence of longitudinal data leaves us with an incomplete understanding of how their processes change over time. This problem is made worse by the tendency of researchers to often focus on collecting isolated examples, such as specific instances of code-switching or borrowing, rather than conducting comprehensive analyses of large datasets of spontaneous bilingual speech. Thomason (2001) once argued that loanword integration can only be studied quantitatively, by measuring how often a borrowed item appeared in a language, which is a hopeless task as it is challenging to determine the frequency without comparing them to native words (Labov, 1972). Given that most published data are derived from a limited number of language pairs and collected from a relatively small group of individuals, linguists face challenges in developing a robust, evidence-based understanding of the frequency and persistence of borrowed items. Further investigation is needed to explore how these items evolve over time and whether they can be effectively distinguished from instances of code-switching. Moreover, the field's slow progress in distinguishing between code-switching and lexical borrowing may also stem from that lexical borrowing has not

generated as much interest as the question of how bilinguals manage two grammars when switching languages within a sentence (Poplack and Dion, 2012). Often, lone incorporations from another language, which are the most common form of bilingual mixing, are treated as instances of code-switching.

2.1.2 Types

Code-switching, as defined above, covers a broad spectrum of patterns, which can be systematically classified into various types based on distinct criteria. One approach to classification focuses on the *structural* aspects of code-switching, specifically examining where the switching points are located within sentences or conversations (Myers-Scotton, 1993; Muysken, 2011; Hoffmann, 2014).

2.1.2.1 Structured classification

Inter-sentential switching , sometimes also called inter-clausal switching, refers to the case that the switching occurs between sentences or clauses, such as (1) [1] .

(1) I'm one of those weird people who loves airports. There's just something liberating yet soothing about it. Bahkan saat aku di situ untuk terbang demi urusan bisnis, bandara itu seperti tempat peristirahatan sementara.(Yusuf et al., 2020)

I'm one of those weird people who loves airports. There's just something liberating yet soothing about it. **Even when I was there to fly for the sake of business, the airport was like a temporary resting place.**

Intra-sentential switching, sometimes also called intra-clausal switching, refers to the case that the switching occurs within a single sentence or clause.

(2) Gue tadi lunch meeting samabos.

I had just lunch meeting **with boss.** (Sahib et al., 2021)

Tag switching refers to the case inserting a tag phrase in one language into an utter-

[1] Throughout the paper, all code-switched examples are presented with the original code-switched sentence on top and the monolingual translation (usually in English) below. The underscored words or phrases are the paired words in the other language within the code-switched sentence and its translation (in bold).

ance that is otherwise entirely in another language. *you know* is the tag phrase in the following example.

(3) es difícil encontrar trabajo estes dias, you know? (Romaine, 1989)
It's hard to find work these days, you know?

Intra-word switching refers to the case that the switching occurs within a single word, by applying the morphology from one language to a word from an another language, such as (4). In this case, the English verb **to hang** is combined with Spanish verbal inflection to create a mixed or codes-witched word.

(4) Voy a hangear con mis amigos. (Stefanich et al., 2019)
I'm going to **hang** with my friends.

Among these types, intra-sentential code-switching usually demands a high level of fluency, as it requires speakers to adhere to the syntactic rules of both languages mid-thought or mid-sentence (Zirker, 2007). Consequently, it is typically employed only by the most proficient bilingual speakers (Poplack, 1980). Many linguists believe that studying this type of code-switching will "yield the greatest fruits in the way of characterizing the linguistic organization of the bilingual cognitive apparatus" (Lipski, 1985), as it offers valuable insights into how bilingual individuals manage and organize their dual-language systems within the mind. However, with some language pairs, such as English-Mandarin, intra-sentential switching can exclusively involve simpler linguistic elements, such as nouns, noun phrases and sometimes discourse markers within clausal boundaries, depending on personal preference (Liu, 2019). In these cases, extensive linguistic knowledge of the donor language (English) may not be necessary. Conversely, inter-clausal switching, which involves combining conceptual and functional words in the correct order, requires a more advanced understanding of both languages involved.

The distribution of different types of code-switching is dependent on both the language pair involved and the context or domain in which it occurs. Poplack (1980) discovered that bilinguals of varying abilities—both fluent and non-fluent—frequently produce code-switches that maintain grammatical correctness in both their first (L1) and second language (L2). Additionally, they found that less fluent bilinguals tend to switch inter-sententially, whereas fluent bilinguals are more likely to switch intra-

sententially. It has also been found the age at which a child acquires the L2 affects the type and frequency of code-switching (Poplack, 1980; Jisa, 2000). (Lipski, 1985) hypothesized that a speaker who acquires a second language after the critical period will rarely engage in intrasentential switching, even if they achieve a high level of fluency in the second language. According to this hypothesis, these late bilinguals are less likely to alternate between languages within a single sentence compared with at sentence boundaries.

2.1.2.2 Functional classification

Another approach to classification is based on the **function** of or **reasons** behind the switching behavior (Abdul-Zahra, 2010). This functional classification seeks to understand why speakers alternate between languages, exploring the social, psychological, or contextual motivations driving this behavior. As this criterion is not apparent from the linguistic context itself, but also relies on the context of why it has been spoken, instead of giving isolated examples here, we give a detailed description of each situation.

Situational switching occurs when there is a change in conversational situations, such as the setting, topics and participants (Blom and Gumperz, 1972). A common example is when two people are talking in their local dialect, when a third person who does not understand the local dialect joins the conversation, they switch to the standard dialect.

Metaphorical switching occurs when there is no such apparent social situation change. Instead, the switch in language codes signifies a change in the psychological distance felt by the speaker, acting as a signal of shifting interpersonal relationships (Blom and Gumperz, 1972). For example, a speaker might switch from an *ethnically specific minority language*, associated with in-group and informal activities, to the *majority language*, associated with more formal relations (Gardner-Chloros, 2009). This switch often signifies emotional distance or annoyance. Conversely, switching from the majority language to the minority language can indicate an attempt to appeal to a shared identity or establish closeness. The following Spanish-English example is taken from a mother's call to her children. The mother starts in Spanish, a language associated with family and in-group communication, and then switches to English, the majority language, to express her annoyance at the children's lack of attention.

(5) Vena ca. Vena ca. Come here, you.

Come here. Come here. Come here, you.(Gumperz, 1982)

Conversational switching occurs naturally within the flow of conversation, independent of changes in the topic or social situation (Gumperz, 1977). It is normally present as intra-sentential code-switching, which produces instances of the two languages in roughly equal proportion.

It should be noted that different scholars may make different distinctions between situational and metaphorical switching, which introduces confusion to the concept. For example, Romaine (2000) consider the topic change contributes to situational code-switching as they identify the topic as a component of a speech event. However, Holmes (2001) think the change of topic sometimes symbolizes a change in the relationship between people, which is commonly associated with metaphorical code-switching. To give an example from Blom and Gumperz (1972): in the village of Hemnesberget, people use Bokmal, the standard Norwegian, for official transactions at the tax office and use Ranamal, the local dialect, for a friendly chat with each other. Code-switching occurs when the worker in the tax office is a personal acquaintance of a citizen and the topic changes from the citizen's spouse to the tax form. Blom and Gumperz (1972) interpret this switching as signaling a change in the participants' roles from personal friends to a citizen and an office worker, which involves the characteristics of metaphorical code-switching; while Azuma (1997) explain the switching as being triggered by situational change (topic change in this case), which is situational code-switching. Therefore, it is important to recognize discrepancies in the treatment of situational and metaphorical code-switching in textbooks on sociolinguistics (Shoko et al., 2003), and understand the complexity and possibility of multiple interpretations of code-switching.

2.2 Reasons for Code-switching

Unfortunately, some researchers assume that those who switch between languages or dialects are unable to fully master any of them, viewing code-switching as a sign of linguistic confusion or deficiency (Hymes, 1972; Redlinger and Park, 1980). The misconception may come from early research on bilingual children's code-switching behaviors and linguistic competency, which suggested that children mix languages be-

cause they are confused and cannot differentiate between the two languages, or because they lack the lexical, grammatical, and/or pragmatic competence in one or both of the languages known (Yow et al., 2017). However, recent studies have provided more complex and contradictory evidence (Chung, 2006; Jennifer A. Vu and Howes, 2010; Yow et al., 2017). These studies have shown that this behavior is evidence that children possess adequate grammatical knowledge of both languages, as like adults, they adhere to grammatical constraints in their code-mixing patterns.

Intention	Description
Quotation	Serves as direct quotations or as reported speech
Addressee specification	Serves to direct the message to one of several addressees
Interjections	Serves to mark an interjection or sentence filler for the purpose of conveying surprise, strong emotion, or to gain attention.
Reiterations	Serves to repeat a message from one code to another code either literally or in somewhat modified form
Message qualification	Serves to qualify constructions such as sentence and verb complements or predicates following a copula
Personalization versus objectivization	Serves to distinguish between talk about action and talk as action, the degree of speaker involvement in, or distance from, a message, whether a statement reflects personal opinion or knowledge, whether it refers to specific instances or has the authority of generally known fact

Table 2.1: Code-switching Functions in Semantic Model

Sociolinguistic research on adults also confirms that code-switching does not necessarily indicate linguistic incompetence, but plays an important role in social functions. The functional criteria described above can be seen as a summary of functional purposes. Gumperz (1982) lists the functions in their semantic model, shown in Table 2.1, which has been employed for analyzing communicative intention in everyday communication. After studying this behavior on students' Facebook wallposts, Ting and Yeo 2020 later added the *referential* function whereby code-switching is used to introduce words which the speaker is familiar with (e.g., cultural terms, formulaic expressions) or to compensate for words that the speaker does not have immediate access to.

2.3 Metrics for Analyzing Code-Switching

As we have seen, code-switching is a complex phenomenon, with different communities and contexts exhibiting unique behaviors. To better understand and compare

these behaviors, researchers have developed various metrics that objectively measure the extent and patterns of code-switching across different datasets. Additionally, these metrics are crucial for enhancing computational linguistic models, as they facilitate the comparison of findings across different studies, promoting consistency and applicability. Here, we classify them into three primary groups: ratio-based, distribution-based and memory measure, which focus respectively on simple word counts, language span distribution, and the ordering of those spans, following Guzmán et al. (2017). This section provides the specific background for the research presented in Chapter 3, where we explain the limitations of existing approaches and propose a new method to address these challenges.

2.3.1 Ratio-based measure

2.3.1.1 M-index

M-index(Barnett et al., 2000) is one of the earliest metrics to be proposed in this category. It quantifies the inequality of the distribution of language tags in a corpus of at least two languages, shown as follows:

$$\text{M-index} := \frac{1 - \sum p_j^2}{(m-1) \cdot \sum p_j^2} \tag{2.1}$$

Where m is the total number of languages in the corpus, p_j is the proportion of words in language j relative to the total number of words[2] in the corpus, and j ranges over the languages present in the corpus. The more balanced the word distribution across the n languages, the more code-switched the dataset, thus the closer the M-index is to 1.

2.3.1.2 Language Entropy

Language Entropy (Guzmán et al., 2017) can be considered as an alternative to the M-index. It is the Shannon entropy of the language tag distribution, and quantifies the uncertainty or unpredictability in the use of different languages within a given dataset. Specifically, it measures how many bits of information are needed to describe this distribution as follows, where we use the same conventions of notation as before:

[2]There has also been work on using Intonation Units as tokens instead of words (Pattichis et al., 2023).

$$\text{LE} := -\sum_{j=1}^{m} p_j \log_2(p_j) \tag{2.2}$$

The lower bound is 0, which represents a monolingual corpus, while the higher bound is

$$-\sum_{j=1}^{m} \frac{1}{m} \log_2(\frac{1}{m}) = \log_2(m) \tag{2.3}$$

and in such a case, each language is represented equally.

2.3.1.3 I-index

I-index (Guzman et al., 2016; Das and Gambäck, 2014; Gambäck and Das, 2016) provides a supplementary method to measure the frequency of code-switching behaviors beyond a simple language incidence ratio. First, a switch point is defined as any token in the corpus that is preceded by a token from a different language. Then I-index describes the ratio of the switch points to the number of all tokens or the approximate probability that any given token in the corpus is a switch point.

$$\text{I-index} := \frac{1}{n-1} \sum_{1<j<n} S(l_{j-1}, l_j) \tag{2.4}$$

As with M-index, we assume a corpus composed of tokens tagged by language: here, $\{l_j\}$ denotes a tag of jth token, with j ranging from 1 to n, the size of the corpus. $S(l_{j-1}, l_j) = 1$, denoting a code-switch, if $l_{j-1} \neq l_j$; 0 otherwise. The lower bound is also 0, representing monolingual usage, while the upper bound is 1, indicating a language switch at every word.

2.3.1.4 Code-mixing Index

Code-mixing index (CMI) (Das and Gambäck, 2014) was the first index proposed for social media text. It first finds the most frequent language and then measure the frequency of the words belonging to all other languages present.

$$CMI := \begin{cases} [1 - \frac{max(w_i)}{n-u}] & n > u \\ 0 & n = u \end{cases} \tag{2.5}$$

Here, w_i is the number of words of the language i, $max(w_i)$ represents the number of words of the most prominent language, n is the total number of tokens, u represents

the number of language independent tokens (such as named entities, abbreviations, mentions, and hashtags). The index ranges from 0 to 1.

Later, in (Gambäck and Das, 2016), similar to I-index, the number of switching points is introduced to CMI equation, and for $n > u$ the equation is updated as:

$$CMI_{updated} := a \cdot [1 - \frac{max(w_i)}{n-u}] + (1-a) \cdot \frac{\sum_{1<j<n} S(l_{j-1}, l_j)}{n} \tag{2.6}$$

where $\frac{\sum_{1<j<n} S(l_{j-1}, l_j)}{n}$ is the number of switching points per token while a is the weight.

2.3.2 Distribution-based measures

2.3.2.1 Burstiness

By considering language tags irrespective of their position or context, none of previous metrics consider how languages' distribution across different code-switching datasets may vary. With a *language span* being the distance between switch points, Burstiness(Guzmán et al., 2017), bounded between -1 and 1, thus considers how much the language span distribution differs from the Poisson distribution (switching occurs at random) shown here:

$$\text{Burstiness} := \frac{(\sigma_\tau / m_\tau - 1)}{(\sigma_\tau / m_\tau + 1)} = \frac{(\sigma_\tau - m_\tau)}{(\sigma_\tau + m_\tau)} \tag{2.7}$$

With σ_τ the language spans' standard deviation and m_τ the spans' mean, Burstiness encodes whether switching activity seems random (values closer to 1) or regular (closer to -1).

2.3.2.2 Span Entropy

Similar to language entropy, span entropy (Guzmán et al., 2017) is defined as the Shannon entropy of the language span distributions, which measures the unpredictability or uncertainty in the lengths of these spans. Higher entropy indicates that the span lengths are more varied and unpredictable, while lower entropy suggests more consistent span lengths.

$$\text{SE} := -\sum_{d}^{M} p_d \log_2(p_d) \tag{2.8}$$

Where M denotes all possible states of the spans and p_d denotes the sample probability of a span of length d. The lower bound is still 0, which represents all spans have the same length, and the higher bound is $\log_2(M)$, which represents that for a corpus with M possible language span states, each possible span as probability $\frac{1}{M}$. To give an example, consider a corpus where there are three possible language spans, with lengths of 2, 4, and 6 tokens, respectively. Suppose that in this corpus, the probabilities of these spans occurring are as follows:

$$p_2 = 0.5 \quad p_4 = 0.3 \quad p_6 = 0.2 \tag{2.9}$$

Here, the total number of states $M = 3$. Using the formula for span entropy LE, we can calculate:

$$LE = -(0.5\log_2(0.5) + 0.3\log_2(0.3) + 0.2\log_2(0.2)) = 1.485 \tag{2.10}$$

If all language spans were of equal length, the entropy would be 0. On the other hand, if all spans were equally likely, the maximum entropy would be:

$$LE_{max} = \log_2(M) = \log_2(3) = 1.585 \tag{2.11}$$

2.3.3 Memory

Memory (Guzmán et al., 2017), also bounded [-1,1], considers the correlation between lengths of adjacent language spans:

$$\text{Memory} := \frac{1}{n_r - 1} \sum_{i=1}^{n_r - 1} \frac{(\tau_i - m_1)(\tau_{i+1} - m_2)}{\sigma_1 \sigma_2} \tag{2.12}$$

Where n_r is the number of language spans in the distribution, τ_i is the current language span, τ_{i+1} is the following language span, σ_1 is the standard deviation of all language spans but the last, σ_2 is the standard deviation of all language spans but the first, m_1 is the mean of all language spans but the last, and m_2 is the mean of all language spans but the first. If a data-set scores closer to -1, lengths of speech spans in one language are negatively correlated with lengths of spans in the other: they tend not to be similar in length. Comparatively, values close to 1 suggest spans much more even in length.

2.4 Triggering Hypothesis for Code-Switching

Understanding the emergence of code-switching has been a significant topic for decades, with various theories proposed to explain its occurrence. One such theory is *Triggered Switching*, which suggests that code-switching is more likely to occur, or be triggered, in the presence of cross-linguistic overlap (Clyne, 1980). This overlap can include elements such as cognates (words that look and sound similar in both languages), false friends (words that look similar but have different meanings), homophones (words that sound the same but have different meanings), proper nouns, or loan words.

In Clyne (1980)'s earliest triggering hypothesis, code-switching was hypothesised to occur either just before or immediately after trigger words, suggesting an influence at the surface level of the sentence. Subsequent analyses by Broersma and De Bot (2006) tested this by examining a bilingual corpus of Dutch and Moroccan Arabic. This work analyzed code-switching frequency around trigger words, revealing that code-switching was significantly more likely to occur immediately following trigger words, thus providing empirical support for the contextual influence of trigger words on code-switching. Furthermore, assuming triggering takes place at the lemma level, Broersma and De Bot (2006) adjusted the hypothesis to the clause level by segmenting conversations into basic clauses, each containing maximally one main verb. This allowed them to establish that words, located near, not just immediately after a trigger word within the same clause, might also be more likely to be code-switched. In follow-up work, Broersma (2009) confirmed that code-switching was more frequent around trigger words or within the same clause in Dutch and English but found no significant difference in code-switching based on whether the trigger word preceded or followed the switch. This suggests that while trigger words influence the likelihood of code-switching, the position of the trigger word relative to the switch may not be a crucial factor in determining exactly when code-switching occurs. However, we note that these analyses were severely limited by the corpus size. Soto et al. (2018) tested the hypothesis on a much larger English-Spanish corpus, and confirmed cognate words facilitate code-switching when immediately preceding the code-switch. They also noted that part-of-speech tags and speaker entrainment significantly affect code-switching frequency. The hypothesis that cognate words would facilitate codeswitching, including intra-clausal and inter-clausal [3] types was further clarified in (Broersma et al.,

[3] They use clause-internal and clause-external in the original paper, where clause-internal means the clause contains words from two languages and clause-external is assesed by if the language of the finite verb in that clause differed from that of the previous or following clause

2020). Their study on Welsh-English conversations also showed that speakers who used more cognates throughout the conversation code-switch more, and that cognate-dense clauses had a higher likelihood of clause-external codeswitching. Interestingly, they found that codeswitching was still facilitated for some time after the production of a cognate, while *hearing* a cognate had no effect, aligning with the idea that syntactic priming is stronger within than across speakers. Importantly, in these studies, cognates were treated as occurring within the matrix language and immediately preceding the code-switch, rather than being part of the switched material. These findings provide the background for the work presented in Chapter 4, where we explain the limitation of current corpus-based statistical analyses and propose a new approach to test the triggering hypothesis.

It should be noted that the current understanding of the triggering hypothesis suggests that trigger words and code-switches simply co-occur in spontaneous conversations (Broersma and De Bot, 2006). This perspective marks a shift from earlier views in publications (Clyne, 1967, 1972, 1980), which implied a direct causal relationship. As Clyne (2003) argues, it is more accurate to refer to this phenomenon as *lexical facilitation* rather than triggering, since other factors, such as structural elements (like grammar) and sociolinguistic factors (like the social context) also influence when and why code-switching happens. Furthermore, the analysis of corpus data cannot confirm whether trigger words cause code-switching because such data is inherently acausal, making it impossible to definitively determine the directionality of the relationship between trigger words and code-switching. It may even be that code-switching itself enhances the likelihood of using trigger words.

To further understand the causal link between trigger words and code-switching, various experimental studies have been conducted with bilingual speakers. In sentence production tasks, as detailed by (Kootstra et al., 2012), participants initially repeated a sentence that included a code-switch immediately following either a cognate or a non-cognate. Subsequently, they were tasked with describing a provided picture using a code-switched sentence. The researchers analyzed how frequently participants initiated a code-switch at the same position in the sentence as the switch in the prime sentence, particularly when a cognate was present in both the priming sentence and the target picture. Their findings indicate that the presence of cognates facilitates code-switching at the same position primarily for high-proficiency L2 speakers. Conversely, in a shadowing task (Sybrine Bultena and van Hell, 2015), where participants repeated recorded speech immediately as quickly and accurately as possible, no effect of verb

cognates on language switching was observed. In a self-paced reading comprehension task (Bultena et al., 2014), where participants read sentences and their reading times were measured, no facilitation effect was found when a cognate preceded the switch. More recently, Kootstra et al. (2020) explored code-switching in the context of a dialogue game where participants described pictures of cognates, false friends, or control words under two conditions: either following a deliberate code-switch by a confederate or not. Again, there was no significant effect of cognate status on the likelihood of triggering code-switches, suggesting that cognates are not consistently more likely to induce code-switches than control words.

In summary, the evidence for the triggering hypothesis is mixed: corpus analyses suggest that code-switches may co-occur with cognates, and experimental studies overall do not find that cognates trigger code-switching, except potentially for highly proficient L2 speakers (Neveu et al., 2022).

2.5 Progression in code-switched NLP

In recent years, NLP has made significant progress, with growing efforts to bridge the gap between monolingual and multilingual research. This shift is crucial as NLP applications increasingly need to function in environments where multiple languages are used interchangeably, such as in social media analysis, automated customer service, and machine translation. As more researchers focus on these challenges, the field is rapidly evolving to develop models that can manage complex language use. In this section, we will review three research directions that align with the work presented in Chapter 4, 5 and 6.

2.5.1 Code-Switching Data Annotation

Data is one of the key factors in successful modeling, and while our work presented in the following chapters does not directly contribute to annotation tasks, it relies on datasets that have often been automatically annotated. Understanding the nuances of these annotation tasks is crucial, as it helps identify potential problems or limitations in the models that may arise from the quality and accuracy of the underlying datasets.

Word-level language identification task on code-switched text involves determining the language of each word or phrase within a sentence. This is more challenging than in multilingual but non-code-switched text, where languages are typically separated

into distinct blocks, making identification easier based on larger context or structural cues. The fluid and unpredictable nature of the language shifts in code-switched text demands more sophisticated models to accurately distinguish languages with minimal context. Existing approaches can broadly be categorised into dictionary-based classification, supervised training with code-mixed annotated data and training models with no or little access to code-mixed annotated data. A dictionary-based language detector typically predicts the language of a word by analyzing its frequency across pre-compiled dictionaries for multiple languages (Nguyen and Doğruöz, 2013; Barman et al., 2014). However, since code-switching often occurs in informal contexts such as conversations and social media - where word forms can vary significantly, dictionary look-up methods struggle with coverage and performance. Consequently, supervised learning with annotated data has become one of the most commonly used approaches. Early efforts employed models such as support vector machines (SVM) (Das and Gambäck, 2014), Naive Bayes (King and Abney, 2013), and Conditional Random Fields (CRF) (Barman et al., 2014) to train word-level classifiers. With the advent of neural networks, more sophisticated techniques like recurrent neural network (RNN) and convolutional neural network (CNN) have been utilized for sequence classification tasks. Apart from word-level information, character-level information has also been used to handle unseen word variations as input (Chang and Lin, 2014; Jaech et al., 2016; Samih et al., 2016). However, supervised approaches still face challenges related to data sparsity and domain adaptation. To address these issues, researchers have also explored semi-supervised and unsupervised methods combining confidence scores from various monolingual identification systems to improve the accuracy of final predictions (King and Abney, 2013; Gella et al., 2014).

Similarly, POS tagging on code-switched text can be viewed as a word-level classification task, where the labels correspond to POS tags rather than language IDs. Various strategies have been proposed to improve performance, including combining outputs from monolingual taggers and using models such as SVM (Solorio and Liu, 2008b), CRF (Aguilar et al., 2020), and LSTMs (Soto and Hirschberg, 2018). In recent years, multilingual pre-trained transformers such as XLM-R (Conneau et al., 2020a) and mBERT (Devlin et al., 2019) have also been adapted to handle code-switched POS tagging and LID tasks, often via fine-tuning. These models can implicitly leverage cross-lingual knowledge and contextual cues, reducing the reliance on explicit annotations for low-resource language pairs.

Beyond sequence labeling tasks like LID and POS tagging, recent work has also

explored the use of large language models (LLMs) such as GPT (Brown et al., 2020), mT5 (Xue et al., 2021), and mBART (Liu et al., 2020) for higher-level code-switched tasks, including sentiment analysis, named entity recognition, and question answering. In such cases, the availability and quality of word-level annotations still play a role in training or evaluation, but LLMs have shifted the burden from large labeled datasets to high-quality prompts and task formulations. However, these models are not yet fully effective at handling code-switching. For example, Zhang et al. (2023) show that multilingual LLMs often underperform compared to smaller models that have been explicitly fine-tuned on code-switched data, especially on evaluation benchmarks involving language mixing. This highlights the ongoing limitations of current LLMs in modeling the complex and fluid patterns of code-switching without targeted adaptation.

It should be noted that, like other NLP tasks, the literature reviewed here typically focuses on a small number of language pairs. The need for annotated code-switched data varies depending on the specific languages involved and their linguistic distance. Language pairs that share many homographs typically require more code-switched training data to achieve reliable performance, while more typologically distant languages often benefit more from the inclusion of additional monolingual data (AlGhamdi et al., 2016).

2.5.2 ASR

Although Automatic Speech Recognition (ASR) systems can process code-switched speech in a manner similar to monolingual data, they face challenges that overlap with those encountered in the aforementioned text-based annotation tasks. They both struggle with the limited availability of annotated data and the complexities of informal, conversational settings. Additionally, ASR systems must deal with the continuous nature of speech, which lacks the discrete boundaries present in text, adding another layer of difficulty in managing and identifying language switches accurately.

Initial approaches to handling code-switched speech often involve a language identification component that first determines the language of each speech segment before passing the data to the corresponding monolingual ASR system (Lyu et al., 2006; Wu et al., 2006; Bhuvanagiri and Kopparapu, 2010). While this method is straightforward, it assumes that code-switched speech segments can be independently identified and easily separated, leading to delays and inaccuracies during the language identification stage (Weiner et al., 2012). In contrast, joint approaches, which integrate language

identification directly into the ASR process, serve as the basis for the hybrid systems discussed henceforth. These systems operate similarly to multilingual systems, incorporating a multilingual acoustic model, a pronunciation dictionary that merges word entries from both languages, and a multilingual language model that allows language switching during speech recognition. In this approach, the choice of phone inventories is important and many studies have been conducted to merge phone sets of different language pairs using both manual and automatic methods. Manual merging involves linguists and phoneticians who carefully align and integrate phone inventories from different languages, creating a unified phone set (Kohler, 1998; Lyudovyk and Pylypenko, 2014). This approach is detailed and precise but can be labor-intensive and less scalable. Automatic merging techniques, on the other hand, often employ clustering algorithms and machine learning models to categorize similar phones from different languages based on their acoustic properties (Sivasankaran et al., 2018). By grouping phones into broader categories, automatic methods effectively increase the training data available for each phone, enhancing the system's capacity to manage pronunciation variations between languages and improving overall ASR performance. The multilingual language component can be improved by training on a larger amount of code-switched text, a topic we will return to later.

In recent years, end-to-end (E2E) models have been increasingly explored for handling the challenges of code-switched speech, effectively combining acoustic, lexicon, and language models into a single, unified framework. These models simplify the ASR pipeline but typically require large amounts of annotated data to perform well. To mitigate this data dependency, researchers have incorporated language identity information through an auxiliary module via multitask learning (Zeng et al., 2019; Shan et al., 2019; Luo et al., 2018). Additionally, various data augmentation techniques have been investigated to generate code-switching data artificially. For instance, Seki et al. (2018) and Ali et al. (2021) created synthetic code-switched speech by concatenating monolingual utterances from language-dependent corpora. However, a significant challenge remains: fine-tuning these models on code-switched data often leads to degraded performance on monolingual speech. To address this issue and improve model robustness, techniques such as Learning Without Forgetting (LWF) and adversarial training have been proposed, aiming to maintain the model's performance across both monolingual and code-switched scenarios (Shah et al., 2020; Madhumani et al., 2020).

2.5.3 Text generation

Considering that there is a much larger amount of monolingual text than code- switched text, the generation of code-switched from monolingual texts has become a key research area. One of the foundational approaches in this field involved rule-based methods, where syntactic and lexical constraints were leveraged to generate code-switched sentences that simulate realistic linguistic behaviors. For example. Shen et al. (2011) and Yu et al. (2023) simply align the words on the parallel sentences and words are replaced across languages based on frequency. Other than specifying words independently, imposing language theories to parallel monolingual texts has also been a popular research direction. The Matrix Language Theory (Myers-Scotton, 1997) describes how in bilingual code-switching, one language (the matrix language) structurally dominates the conversation, setting the grammatical framework of the sentence. This allows for segments from the other language (the embedded language) to be inserted at points that are grammatically suitable within this framework during Hindi-English code-switching (Bhat et al., 2016). Similarly, Equivalence Constraints Theory (ECT) (Poplack, 1980) has been utilized to predict switching points by identifying positions in the sentence where the grammatical structures of both languages align (Li and Fung, 2012; Pratapa et al., 2018). Our work presented in Chapter 5 also applies ECT, and we will introduce it in more detail later.

However, the limitations of rule-based approaches, particularly their reliance on predefined linguistic rules and constraints, have led researchers to explore machine learning techniques to increase the diversity and accuracy of generated code-switched text. A bilingual attention language model has been proposed to learn word embeddings that represent equivalent words across two languages, modeling cross-lingual sequential dependencies (Lee and Li, 2020). Winata et al. (2018) applied pointer-generative networks which are trained to generate code-switched text from parallel text with a copy mechanism. Generative Adversarial Networks (GANs) have been trained on real Code-switched text to modify the monolingual sentence to a code-switched sentence (Chang et al., 2019). More recently, powerful pretrained models like em-BART have been used to translate a sentence in one language into a code-switched version (Gautam et al., 2021). Again, our work presented in Chapter 6 also applies a machine learning approach to generate code-switched text, and we will introduce the details of these models later.

2.6 Summary

In summary, code-switching is a complex phenomenon that can manifest in various forms, with individuals switching languages for multiple reasons. The triggering hypothesis suggests that code-switching often co-occurs with cognates, though this has yet to be definitively confirmed through experimental studies. Additionally, most theoretical approaches have historically relied on data from single communities, potentially underemphasizing the cultural aspects of language. To address this bias and more accurately evaluate the structural accounts of code-switching, systematic cross-community comparisons within and between language pairs are essential. While NLP research has made significant progress, several challenges persist. Collecting naturalistic code-switched data remains difficult, as experimental settings may not fully capture the spontaneity and complexity of code-switching in everyday conversation. Furthermore, analyzing these behaviors involves navigating numerous linguistic, cognitive, and social variables, making it a resource-intensive task. Another key challenge is the generalizability of findings; insights from specific bilingual communities may not apply universally due to the diversity of bilingual experiences and language pairs.

Chapter 3

Characterizing code-switching

With handling code-switching becoming an increasingly important topic in speech technology, driven by the expansion of low-resource and multilingual methodologies, it is vital that we recognize the diversity of code-switching as a phenomenon. In this chapter, we propose a framework that leverages linguistic findings as makeshift ground truths to assess the quality and sufficiency of existing metrics designed to characterize differing code-switching styles across datasets. We also introduce a new metric, the **T-index**, which uses machine translation systems to quantify the semantic ambiguity of code-switched words in relation to the participating language pair. These metrics are primarily intended for researchers and practitioners aiming to compare code-switching behavior across domains, for example, conversational versus technical speech, rather than for evaluating individual system outputs. As such, while the datasets used here differ from those in later chapters, the insights drawn help frame the structural assumptions underlying our generation and modeling approaches. The rest of the thesis focuses on improving performance within a single code-switching domain, where such cross-domain metrics are less directly applicable. This chapter is based on content originally presented in our InterSpeech 2024 paper (Chi et al., 2024).

3.1 Introduction

Handling code-switching has become an increasingly important focus in speech technology, largely driven by the expansion of multilingual ASR methodologies. These advancements have enabled ASR systems to be effectively deployed across a wide range of contexts and domains where code-switching is likely to occur. However, developing such systems for real-world applications requires a nuanced understanding

of code-switching behavior. Not all code-switching is the same; it varies significantly based on factors, as described in Chapter 2, such as the languages involved, the sociolinguistic context, the speakers' language proficiency, and the specific communicative purposes of the interaction. This variation poses a challenge for ASR systems, which must be able to generalize across diverse code-switching patterns rather than rely on narrowly tuned solutions tailored to specific cases. Consider the intra-sentential examples below:

(1) bhiodh na gaidheil dìreach math fhèin air ready steady cook s tha mi a smaointinn gu bheil e mar phàirt den chultar againne cuideachd bidh thu [1]

Gaelic people would be good at ready steady cook **and I think it's part of our culture too.**

(2) और विभिन्न crystal systems के crystal structures को दिखाना उदाहरण के लिए cubic hexagonal (Diwan et al., 2021)

and showing the crystal structures **of different** crystal systems **for example** cubic, hexagonal

(3) i worry yam kengoku it's the next meeting on friday and kengoku I'm going out chomi nomfana wam (Reitmaier et al., 2022)

my worry **now** its the next meeting on Friday and **now** Im going out **friend with my man'**

All examples come from code-switching datasets[2]; however, there is a notable contrast between the singular English segment within an otherwise monolingual Gaelic sequence in (1), and the far more frequent switches between Xhosa and English observed in (3). Additionally, the use of English in (1) and (2) is largely restricted to technical concepts and named entities, whereas in (3), English is employed much more freely, both syntactically and semantically.

Now consider what would be required from an ASR system to recognize each of these examples. A system built or trained on code-switching data primarily resembling (1) would likely not be sufficient to handle input in the style of (3). We believe that treating code-switching as a monolithic phenomenon thus poses a significant risk of domain mismatch. Given such code-switching style variation, it may not be possible

[1] `https://learngaelic.scot`

[2] In this chapter, *Code-switching datasets* refers to transcripts of code-switching speech, not written text.

to guarantee that a system or methodology developed on one code-switching dataset would generalize to other languages or contexts.

Motivated by this conundrum, there have been efforts to develop metrics to measure the richness, level, or complexity of code-switching in datasets. As described in Chapter 2, popular metrics include M-index(Barnett et al., 2000), I-Index(Guzman et al., 2016), and Code-Mixing Index(Srivastava and Singh, 2021), which rely on counts of tokens or switching points, as well as the Burstiness and Memory metrics(Guzmán et al., 2017), which rely on characterizing the distribution of participating languages.

It has become standard practice to cite at least one such metric when introducing new code-switching data into research and it theoretically facilitates more robust code-switching ASR development. However, we believe caution should be warranted: relying on these metrics to gauge the complexity of a dataset assumes that they accurately and *sufficiently* characterize the space of code-switching variation. Testing this requires a clear definition of what constitutes a meaningful division of code-switching variation. However, there is a circularity issue: it is problematic to assess a metric's effectiveness at identifying code-switching styles across different datasets when the same metric is utilized to define the occurrences of code-switching.

We argue that what is needed are methods to analyze and assess the sufficiency of such metrics, whether in isolation or in combination, that are grounded in external factors. Linguistic research provides valuable insights in this regard. Extensive linguistic literature suggests that surface variability in code-switching is not random but influenced by latent variables related to meta-properties of the speech situation and setting. For instance, studies have shown that speakers of different age groups or genders systematically differ in the way they code-switch within conversation (Poplack, 1980, 2013; Post, 2015; Deuchar et al., 2016). The specific languages involved and their typological and socio-political relationship have also been shown to correlate with code-switching richness levels/styles (Poplack, 1988; Guzman et al., 2016; Guzmán et al., 2017). Specifically relevant to this paper is the understanding that conversational code-switching, especially in *informal* contexts, is generally considered to exhibit a richer, more varied style compared to code-switching in formal, monologue, or technical settings (Sulminski, 2022; Beatty-Martínez et al., 2020).

The contributions from this chapter are twofold: we introduce a method of analyzing code-switching richness metrics that leverages linguistic findings as makeshift ground-truths. We propose to assess metrics by considering the extent to which their outputs align with linguistic research, such as the observation that code-switching tends

to be more complex in conversational settings than in formal or technical settings. Our framework allows us to systematically benchmark existing metrics. It also enables investigation of alternative methods of code-switching richness measurement. Second, we introduce the **T-index**, which uses a pre-trained machine translation system to better model the relationship between languages at each switching point.

3.2 Related work

We are not the first to critique the reliance on token counts or distributions in existing code-switching metrics. This approach presents a significant limitation, as these metrics often fail to account for the syntactic, semantic, or phonological properties of the data. For instance, Srivastava and Singh (2021) conducted a study where they collected 10 Hinde-English code-switching datasets, each originally developed for various NLP tasks such as named entity recognition, sarcasm detection, language identification, sentiment analysis, hate-speech detection, machine translation, irony detection, and natural language inference. They sampled one sentence from each dataset to evaluate the effectiveness of metrics like CMI, M-index, I-index, Burstiness, and Memory. Human annotators were also employed to rank these sentences based on the degree of code-mixing (scoring from 0 to 10, ranging from monolingual to a high degree of code-mixing) and readability (also scored from 0 to 10, from unreadable due to spelling mistakes and lack of sentence structure, to highly readable with clear semantics and easily understood words). The study revealed that there was no significant relationship between the degree of code-mixing and readability. Moreover, none of the existing metrics could independently measure the human ratings effectively. Similarly, Kodali et al. (2022) showed that even when code-switched examples exhibit the same patterns as described by language tag distributions, they can vary in terms of cognitive and computational processing difficulty depending on the syntactic nature of the switched words.

Recognizing these limitations of basic token counts based metrics reviewed in Chapter 2, there have been efforts to develop more robust metrics that better reflect the linguistic complexity of code-switching. In this section, we will review recent metrics developed with similar motivations and explain how they differ from our proposed approach.

3.2.1 SyMCoM

The Syntactic Measure of Code Mixing (SyMCoM) employs the distributional differences of POS categories across languages as a metric for assessing structural complexity, as represented in Equation 3.1, where $N_{SU_{Li}}$ denotes the counts of SU in language i (Kodali et al., 2022). In this framework, *SU* represents syntactic units, which can be individual POS tags or broader *classes* of POS tags. For their study, they defined two classes: Open and Closed. The Open Class includes content words, such as nouns, adjectives, and verbs, while the Closed Class includes function words, such as pronouns, demonstratives, and other grammatical markers.

$$SyMCoM_{SU} := \frac{(N_{SU_{L1}}) - (N_{SU_{L2}})}{\sum_{i=1}^{2} N_{SU_{Li}}} \tag{3.1}$$

The $SyMCoM_{SU}$ score ranges between $[-1, 1]$, with its polarity indicating which language, among $L1$ and $L2$, contributes a higher number of tokens for a particular SU. The absolute value of the score reflects the degree of skew towards one language over the other. A $SyMCoM_{SU}$ score closer to zero suggests a balanced contribution between L1 and L2 for that specific SU, implying an equal distribution of syntactic units across both languages. It has been found that in Hindi-English code-switching, Open class categories are more likely to be switched than the Closed class. By comparing $SyMCoM_{SU}$ scores with the CMI across various corpora, it was found that datasets with similar CMI scores can have differing $SyMCoM_{SU}$ scores. This discrepancy indicates that $SyMCoM_{SU}$ captures syntactic aspects of datasets that are not reflected in CMI scores.

However, in practical terms, this metric's effectiveness is heavily dependent on the performance of the POS tagger used, which is often pretrained on monolingual data and may not be well-suited for code-switched contexts. While $SyMCoM_{SU}$ demonstrates variability across different corpora, it lacks grounding in specific linguistic phenomena to clarify the meaning of these differences. Consequently, it is unclear what constitutes a significant or minor difference in $SyMCoM_{SU}$ scores. Additionally, variations in POS distribution could arise from differences in the domains of the corpora rather than reflecting distinct code-switching behaviors.

3.2.2 CF

The Complexity Factor (CF) was proposed to enhance the CMI metric by incorporating additional considerations beyond the mere fraction of code-switched words in the

corpus (Ghosh et al., 2017). CF takes into account not only the number of languages involved but also the number of code-switching points present throughout the corpus. CF considers three different aspects: Language Factor (LF), Switching Factor (SF) and CMI.

LF quantifies the number of different languages in a sentence as a fraction of the total number of words as follows, where n is the number of words and m is the number of distinct languages in the sentence.

$$LF = \frac{n}{m} \tag{3.2}$$

SF represents the ratio of actual switching points in a sentence to the maximum possible switching points as follows, where S is the number of switching points, equivalent to the I-index.

$$SF = \begin{cases} \frac{S}{n-1} & n > 1 \\ 0 & n = 1 \end{cases} \tag{3.3}$$

Finally, CF is computed by combining all three factors as follows: where a and b are the weights for CMI and SF, respectively. f is a function of LF that can be linear or geometric.

$$CF = \frac{a \cdot CMI + b \cdot SF}{f(LF)} \tag{3.4}$$

By testing the CF with various combinations of weights and functions, the resulting range and mean values offer insights into the complexity of different corpora. However, similar to the SyMCoM metric, CF faces limitations. It can highlight variations in complexity, but it does not provide detailed insights into the specific differences in code-switching behaviors beyond broad features. Additionally, CF does not account for differences between the languages involved, which limits its ability to fully capture the nuances of code-switching.

3.2.3 CESAR

The CodE-Switching According to a Reference Language (CESAR) metric was introduced to address the limitations of the CF by considering the noise present in a document relative to a reference language (Abidi and Smaïli, 2022). Essentially, this metric assesses how *clean* code-switched documents are in comparison to a reference corpus. The CESAR metric is formulated as follows, where $P_r(C)$ represents the proportion of the number of languages in each document t_i in the corpus C, and B_r denotes

the rate of noise introduced by words that are different from the reference language r. a again is the weight that is determined empirically. n is the number of documents of C

$$CESAR(C) = a \cdot P_r(C) + (1-a) \cdot B_r(C) \tag{3.5}$$

$$P_r(C) = \frac{1}{n} \sum_{i=1}^{n} \delta_r(t_i) LF(t_i) \tag{3.6}$$

where $\delta_r(t_i)$ is defined as follows where $l(w)$ represents the language of word w and lr is the reference language:

$$\delta_r(t_i) = \begin{cases} 1 & \text{if } \exists w \in t_i \text{ where } l(w) \neq l_r \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

It should be noted that in this context, LF still represents the language factor, but it differs from the definition provided in Equation 3.2. Specifically, this version of LF assigns a value of 1 if no words in the document are written in the reference language, and a value of 0 if all words are written in the reference language, In other cases, LF is proportional to the number of languages different from the reference language.

$B_r(C)$ considers the ratio of words in a document that differ from the reference language as follows, where $N(w_{t_i})$ is the number of words of the document t_i whose language is different from the reference language. N_{t_i} is the total number of words of the document t_i. The condition $l(t_i) \neq l_r$ is applied to restrict the summation to documents whose main language differs from the reference.

$$B_r(C) = \frac{1}{n} \sum_{i=1}^{n} \frac{N(w_{t_i})}{N_{t_i}} LF(t_i) \text{ where } l(t_i) \neq l_r \tag{3.8}$$

Thus, $P_r(C)$ equals 0 if the corpus C is entirely written in the reference language, and equals 1 if all documents are entirely in other languages. In all other cases, when some documents contain a mix of languages, which gives $\delta_r(t_i) = 1$ and $LF(t_i) \in (0,1)$, leading to intermediate values of $P_r(C)$. These situations correspond to varying degrees of code-switching at the document level. As such, $P_r(C)$ captures not just the presence of language mixing, but also how multilingual each document is. This allows us to evaluate how *clean* (monolingual) or *noisy* (code-switched) the corpus is with respect to the reference language.

The idea behind CESAR aligns closely with the motivation for our work, as both aim to assess the level of code-switching relative to some reference. CESAR was designed to address some of the limitations found in CF, particularly by better capturing the *noise* introduced by other languages compared to a reference language (such as a local dialect). When CESAR and CF were applied to specific examples, CESAR was found to be more effective at identifying this noise.

3.2.4 Intonation Unit level metrics

Pattichis et al. (2023) proposed using intonation units (IUs) as tokens instead of words to account for prosodic constraints in code-switching analysis. IUs are speech segments *uttered under a single, coherent intonation contour* (Du Bois et al., 1983), meaning that a sentence or sequence of words can be converted into a prosodic sentence composed of a sequence of IUs. The IU-Boundary constraint posits that code-switching is more likely to occur at IU boundaries, meaning switching is more probable between two words at the boundary of IUs than within the same IU. This constraint is related to the equivalence constraint, which we briefly reviewed in Chapter 2 and will explore in more detail in Chapter 5. The equivalence constraint requires local equivalence of word order in the two languages around the switching point, though these two constraints are applied independently. By calculating M- and I-index at the IU-token level, Pattichis et al. (2023) demonstrated that this approach could effectively distinguish different mixing types, as validated by comparing results with visual language distribution from five speakers on a Spanish-English corpus. Additionally, they confirmed that speakers generally strongly disfavor within-IU multi-word code-switching, reinforcing the IU-Boundary constraint.

The idea of using IUs instead of words is valuable for analyzing how bilinguals tend to prosodically separate two languages during code-switching. However, this approach shares a similar limitation with using word-level tokens: it does not account for the unique characteristics of the participating languages.

3.3 Proposed Metrics

As referenced in Section 3.1, we propose a framework to compare these metrics by applying them all to a suite of code-switched datasets that vary along some meta-dimension: we choose the domain/speech-setting. This approach allows us to ana-

lyze how these metrics rank the datasets across different language pairs, with a focus on aligning their results with linguistic intuitions regarding conversational versus technical data. We begin by assessing the normalized versions of four existing code-switching metrics, which account for orthographic token differences, alongside our novel T-index within this linguistically-informed framework. To evaluate the sufficiency of these metrics, we calculate scores at the utterance level and then average them across the corpus to derive the final result for each dataset.

3.3.1 Normalized M-index and I-index

As reviewed in Chapter 2, M-index measures the ratio of token counts between two languages, while I-index counts the switch points where adjacent tokens differ in language. We stated above that the characterizing sufficiency of these metrics has been questioned, given their token-count focus. Note here another consequence of such token reliance.

(4) **Sesotho:** and ke a ba rata *(M-Index = 0.47)*

Zulu: and ngiyabathanda *(M-Index = 1.0)*

Translation: and **I like them** *(M-Index = 0.0)* (Taljard and Bosch, 2006)

Previous works utilizing these two metrics ((Guzman et al., 2016; Gambäck and Das, 2016; Guzmán et al., 2017)) claim these methods to be *language independent*. But languages differ in their morpho-syntax and orthographic conventions which, importantly, impacts token distribution across sentences and speech segments.

Consider Example (4). Though both South African languages, Sesotho uses a *disjunctive*, and Zulu a *conjunctive*, orthography. Consequently, the same semantic and syntactic content is represented by more tokens in Sotho than in Zulu. This impacts our code-switching metric computation. A Sesotho-English corpus will always have a higher Sesotho:English token ratio than a Zulu-English corpus', and hence will always receive a lower M-index score.

To factor out these orthographic token differences, we propose normalized versions of M-index and I-index in Equations 3.9 and 3.10, via calibration against parallel corpora. We use t_e as the total number of English tokens in the utterance or document/text and t_{ne} as the total number of tokens supplied by the other participating language. For I-index, similar to Chapter 2, $\{l_j\}$ denotes a language tag given to a token; $S(l_{j-1}, l_j) = 1$ thus denotes a code-switch if $l_{j-1} \neq l_j$; 0 otherwise. We add k as a normalising scaling

factor specific to the language-pair. When k is set to 1, the equations are the same as the original described in Chapter 2.

See Section 3.4.3 for how to do this. Note these formulations assume code-switching between two languages only, one of which is English; however, such methodology could easily be extended to incorporate more languages. Here, the suffix $_e$ denotes English, and $_{ne}$ denotes non-English

$$\text{M-index} := \frac{1 - (\frac{t_e}{t_e+kt_{ne}}^2 + \frac{kt_{ne}}{t_e+kt_{ne}}^2)}{\frac{t_e}{t_e+kt_{ne}}^2 + \frac{kt_{ne}}{t_e+kt_{ne}}^2} \tag{3.9}$$

$$\text{I-index} := \frac{1}{(t_e+kt_{ne})-1} \sum_{1<j<(t_e+t_{ne})} S(l_{j-1}, l_j), \tag{3.10}$$

Due to the modified denominator, the normalized I-index no longer ranges between 0 and 1. Instead, its upper bound becomes:

$$\frac{(t_e+t_{ne})-1}{(t_e+kt_{ne})-1}$$

This value depends on the choice of k. When $k > 1$, the I-index is compressed; when $k < 1$, it is inflated. In all cases, the minimum remains 0, while the maximum varies based on the weighting of token types.

3.3.2 Normalised Burstiness and Memory

Recall that, neither M- nor I-index consider variation attributed to how languages are distributed in a data-set – they do not consider a language tag's position or context. In contrast, as reviewed in Chapter 2, with a *language span* being a consecutive sequence of tokens from the same language, *Burstiness* considers how much the language span length distribution differs from the Poisson distribution and *Memory* considers the correlation between lengths of spans in one language versus another (Guzmán et al., 2017).

As above, we propose normalised versions of these metrics. If we take τ to be a sequence of the language span lengths in the text, we first define:

$$\text{span-length}_{aug} = \begin{cases} \text{span-length} \cdot k, & \text{if span language} \neq \text{English} \\ \text{span-length}, & \text{otherwise} \end{cases}$$

with k again a language-pair-specific scaling factor (Section 3.4.3). We can then define $\bar{\tau}$ as τ but with each span length in τ transformed as above. We can then use:

$$\text{Burstiness} := \frac{(\sigma_{\bar{\tau}}/m_{\bar{\tau}} - 1)}{(\sigma_{\bar{\tau}}/m_{\bar{\tau}} + 1)} = \frac{(\sigma_{\bar{\tau}} - m_{\bar{\tau}})}{(\sigma_{\bar{\tau}} + m_{\bar{\tau}})} \tag{3.11}$$

$$\text{Memory} := \frac{1}{n_{\bar{\tau}} - 1} \sum_{i=1}^{n_{\bar{\tau}}-1} \frac{(\bar{\tau}_i - m_{\bar{\tau}_1})(\bar{\tau}_{i+1} - m_{\bar{\tau}_2})}{\sigma_{\bar{\tau}_1}\sigma_{\bar{\tau}_2}} \tag{3.12}$$

where $\sigma_{\bar{\tau}}$ is the (augmented) language spans' standard deviation and $m_{\bar{\tau}}$ the spans' mean. If $n_{\bar{\tau}}$ is the number of language spans in $\bar{\tau}$, then $\bar{\tau}_i$ is the current language span and $\bar{\tau}_{i+1}$ becomes the following language span. $\sigma_{\bar{\tau}_1}$ is the standard deviation of all (augmented) language spans but the last, $\sigma_{\bar{\tau}_2}$ is the standard deviation of all language spans but the first, $m_{\bar{\tau}_1}$ is the mean of all language spans but the last, and $m_{\bar{\tau}_2}$ is the mean of all language spans but the first.

3.3.3 T-Index

We hypothesize that the reduction of code-switched utterances to language-tag sequences causes us to overlook the lexical semantic features of those words that trigger code-switching in a dataset, information which may be valuable for characterizing a dataset's code-switching style or complexity level. We also believe that such consideration of code-switched words' properties should be done with reference to the language *pair* participating in the code-switching.

Consider the Mandarin-English example (5) below. That Mandarin has no precise equivalent of the word ***idea***, with 想法/主意 ('thoughts') being the closest alternative, likely explains the switch to English at this word, as it allows for expressing the concept with a nuance of creativity or innovation that is not present in Mandarin. Conversely, imagine such a switch within a Spanish-English dataset: Spanish *does* have a direct cognate ***idea*** with connotations closer to the English; the motivation for a switch here would not be identical to the Mandarin scenario. When characterizing the semantic content of the English word 'idea', we must do so differently in the context of Spanish-English compared to Mandarin-English.

(5) 这个 idea 很 有趣

'**This** idea **is very interesting** '

We propose a novel metric, the T-index, which quantifies the translation specificity of code-switched words using a pretrained machine translation (MT) system. Specifically, we employ mBART model to obtain translation candidates for code-switched

tokens. The MT system operates in a zero-shot setting and takes as input only the individual code-switched word, without access to sentential context, in order to isolate lexical ambiguity. Following (Wintner et al., 2023) and our findings in Chapter 4, we focus on words that occur immediately after code-switch points, as these are most predictive of switching behavior. For each such word, we generate a set of translation candidates using the MT model. We then compute a translation score for the first candidate by summing the log-probabilities of all tokens in the translation and normalizing by length. See Equation **??**, where s_i denotes the log-probability of the ith token in the translation, L is the length of the translation output, and p is the length penalty factor (set to 1 by default):

$$\text{T-index} \equiv \frac{\sum s_i}{L^p} \tag{3.13}$$

This score reflects the *specificity* of the translation: a high score indicates a dominant, unambiguous translation, while a flatter distribution across candidates suggests greater lexical ambiguity or translational variability. The T-index is thus interpreted as a proxy for how semantically focused or diffuse a code-switched word is, with respect to its embedding language.

3.4 Experimental Setup and Methods

3.4.1 Code-switched Datasets

Table 3.1: Code-switched datasets

Language pair	Conversational	Technical
Mandarin-English	SEAME(Lyu et al., 2010)	Lectures[3]
Hindi-English	Bollywood(Khanuja et al., 2020)	MUCS(Diwan et al., 2021)

We collate a suite of *pairs* of datasets, each consisting of one conversational code-switching dataset and one more formal or technical code-switching dataset, across both Hindi-English and Mandarin-English language pairs, listed in Table 3.1. The details and examples of each set are given below:

[3] https://www.youtube.com/watch?v=Ye018rCVvOo&list=PLJV_el3uVTsMhtt7_Y6sgTHGHp1Vb2P2J&ab_channel=Hung-yiLee

Mandarin -English	Technical	..., 但是在 Pointwise Convolution 里面 ..., **but** within Pointwise Convolution
	Conversational	我同意 but sometimes 我觉得他就是懒惰 I agree but sometimes I think he's just lazy.
Hindi -English	Technical	नमस्कार polymorphism in java पर spoken tutorial आपका स्वागत है Hello, welcome to the spoken tutorial on polymorphism in Java.
	Conversational	aappe bahut serious charges hain You are facing very serious charges

Table 3.2: Examples from datasets

SEAME

The South East Asia Mandarin-English (SEAME) dataset was developed to study the code-switching behavior between Mandarin and English among Malaysian and Singaporean speakers. The dataset now comprises 192 hours of spontaneous speech from 156 speakers, recorded in both interview and conversation setups. In the interview setup, one interviewer interacts with an interviewee, but only the interviewee's speech is recorded. The conversation setup, on the other hand, involves two participants, with both of their speeches being collected. The topics discussed in both setups are general in nature, covering areas such as personal interests (e.g., books, movies) and family life and relationships. There are in total 162,290 sentences. All speech data in SEAME is manually transcribed at the word level, using English orthography for English words and Simplified Chinese characters for Mandarin words. Each word in the transcription is tagged with a language identifier (e.g., `ENG` or `MAN`), allowing for detailed analysis of code-switching patterns.

Lectures

To the best of our knowledge, there is no publicly available technical Mandarin-English code-switched speech corpus. Therefore, we created one by collecting transcripts from a YouTube playlist containing lectures and educational content from Hungyi Lee's machine learning course. This resulted in a dataset with 32,988 sentences. The initial transcriptions were generated using an automatic system and subsequently manually checked and corrected to ensure accuracy by Hungyi's students.

Bollywood

The Bollywood movie dataset contains code-switched sentences extracted from 18 different Bollywood films. While the data is not artificially generated, it is scripted, which may make it a less natural source of language compared to spontaneous conversations between real speakers. Notably, the Hindi tokens in this dataset are written in Roman script. This necessitates the use of an external language identification (LID) tool, since tokens cannot be distinguished by script alone, unlike in datasets where different transcription systems are used for different languages. Despite the abundance of Hindi-English code-switched speech in Bollywood media, the dataset remains relatively small. This is because it originates from Khanuja et al. (2020), which was developed for natural language inference tasks. The authors applied strict filtering and

manual curation to extract only clean, well-formed code-switched utterances. As a result, the final dataset comprises 7,000 sentences.

MUCS

The Hindi-English subset of the Multilingual and Code-Switching (MUCS) challenges includes approximately 95 hours of speech derived from spoken tutorials, covering a range of technical topics in computer science. This dataset consists of a total of 45,014 sentences.

Table 3.2 presents examples from each dataset introduced. In the technical domain, code-switching often occurs exclusively with technical terms. This pattern is particularly noticeable in the Mandarin-English comparison: in the conversational domain, the word *but* is code-switched, whereas in the technical domain, it typically is not. It is important to clarify that we are not asserting that *but* cannot be code-switched in technical contexts. However, when technical terms are present in a sentence, these terms are more commonly the focus of code-switching.

3.4.2 Data Preparation

We exclusively consider those sentences in our datasets with intra-sentential code-switching. Our aim in comparing these metric is to identify whether they adequately characterize different code-switching styles or propensity to code-switch without being influenced by differing quantities of monolingual content. Since Section 3.3.1 and 3.3.2's metrics essentially treat datasets as sequences of language-id tokens, each dataset must be tokenized, and each token assigned a language tag. We pre-process all data by removing punctuation, non-alphabetic strings, and converting all text in Latin script to lowercase, where case distinctions exist. We tokenize Hindi data-sets by white-space: however as the *lecture* transcripts lack white-space entirely, we tokenize both Mandarin data-sets at a character level to ensure intra-language consistency[4]. Importantly, valid inter-language comparisons remain possible as our *normalized* metric variants account for and factor out differences in tokenization method: see Sections 3.4.3 and 3.5.1. Regarding token language tagging, both Mandarin datasets and the

[4]Traditionally written Chinese omitted white-space and inclusion of such to mark word boundaries remains variable.

Hindi MUCS data utilize different orthographic scripts for their respective language pairs. Switching between scripts thus becomes our proxy for switching between languages. As the Bollywood data-set's Hindi tokens are written in Roman script, we apply Microsoft's LID tool[5] which was chosen after a few rounds of experimentation with different off-the-shelf LID systems.

3.4.3 Normalization Data and Methods

Section 3.3.1 and 3.3.2's metrics requires k: this constant scales a dataset's non-English spans such that the 'token rate' to 'information rate' of these spans is made equal to English (chosen on account of its presence in all datasets.) Though k could be estimated in a number of ways, we employ parallel-corpora here. For Hindi-English, we use the IIT Bombay English-Hindi corpus (Kunchukuttan et al., 2018) and for Mandarin-English, we use the ISI Chinese-English Automatically Extracted Parallel Text corpora (Munteanu and Marcu, 2007). We pre-process this data as in Section 3.4.2 (tokenizing the Mandarin ISI corpora at character level): integral for ensuring our normalization method correctly factors out tokenization method differences. We then obtain the average number of English words per sentence compared with the average number of Hindi/Mandarin words per sentence. Using parallel corpora ensures that the English's semantic content is identical to the Hindi/Mandarin: we can thus attribute differences in token distribution to the languages' orthographic/morphosyntactic conventions.

3.4.4 Machine Translation system

For calculating the T-index, we prepare the data as described above. We utilize a public multilingual machine translation model, specifically the mBART model fine-tuned on multilingual data [6](Tang et al., 2020). This model is designed to handle multiple languages and covers both the English-Mandarin and English-Hindi language pairs present in our datasets. The mBART model employs a standard sequence-to-sequence Transformer architecture and functions as a denoising auto-encoder. It is pre-trained on extensive monolingual corpora in various languages, as illustrated in Figure 3.1, where the model predicts the original text based on its corrupted version, which can either involve token masking or sentence permutation. The mBART model is then further fine-tuned on 50 languages across multiple directions, resulting in the

[5] https://github.com/microsoft/LID-tool?tab=readme-ov-file

[6] https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

mBART50 model used here.

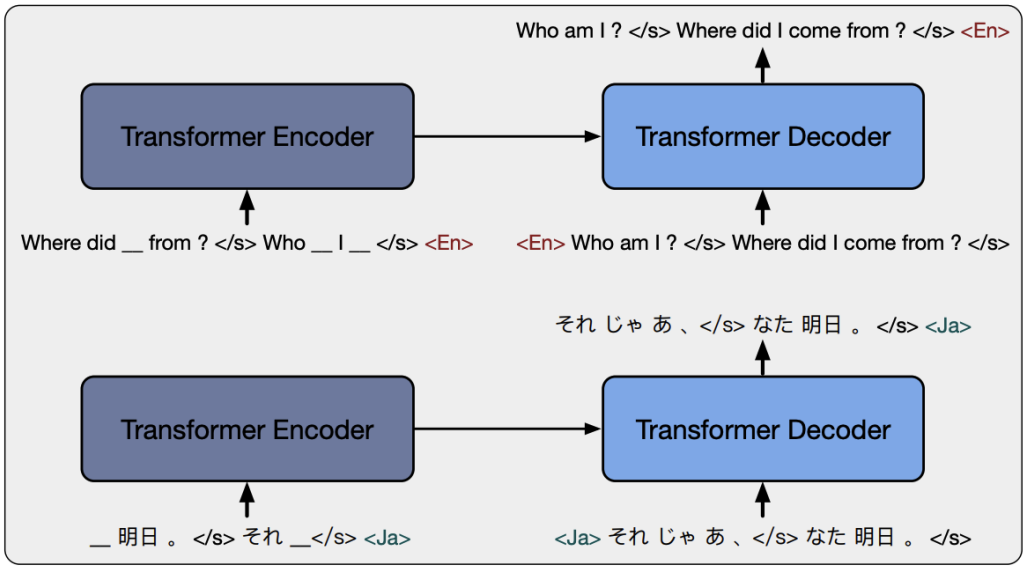


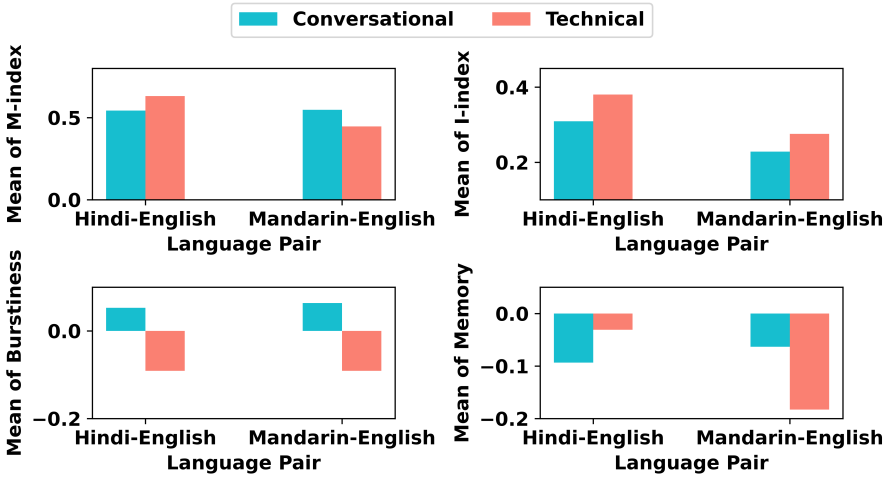
Figure 3.1: Framework for Pre-training. Adapted from (Liu et al., 2020)

Note that while we use mBART-50 here, our approach can be adapted to any machine translation system. Since all chosen datasets involve code-switching to English, we aim to compare the code-switching behavior for each language-English pair, by investigating the translation scores of the code-switched English words in each language.

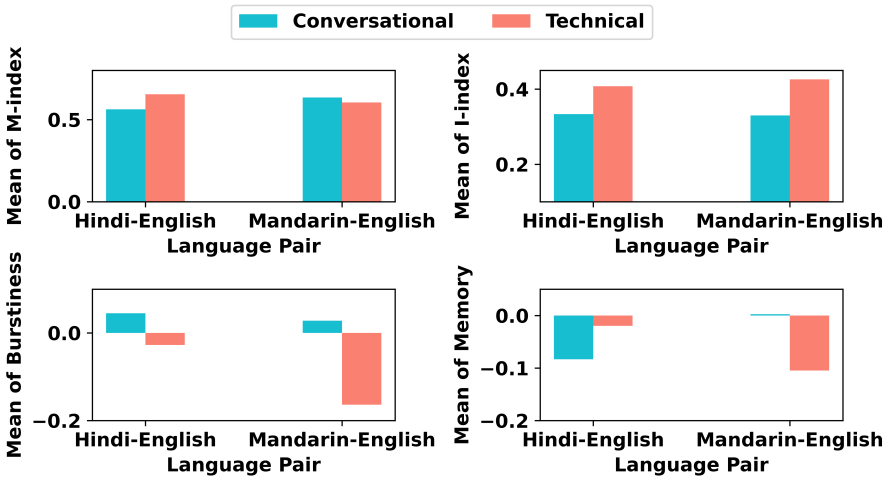
3.5 Results and Discussion

3.5.1 Normalized Metrics

Figure 3.2 presents the code-switching richness scores for the Hindi and Mandarin data-sets: 3.2a with no normalization ($k = 1$), and 3.2b having adding scaling factors, calculated as in Section 3.4.3 (0.909 for Hindi; 0.604 for Mandarin). Note that the relative ordering of the conversational/technical data-set pairs (within each language) remains the same across both sets of scores (for all metrics). However, 3.2b's results technically constitute the more experimentally valid of the two, with the normalization licensing comparison of *absolute* values across the two languages (especially important given our use of differing tokenisation methods for Hindi-English and Mandarin-



(a) before normalization



(b) after normalization

Figure 3.2: Scores obtained after applying four existing code-switching metrics to conversational and technical data-sets.

English data-sets, which our normalisation removes). Notably, with the M- and I-index metrics, which are based on token counts, the Hindi-English and Mandarin-English datasets score more similarly in 3.2b than in 3.2a: possibly suggesting the existence of baked-in typological language differences prior to normalization. Specifically, the difference in mean M- and I-index values between the two language pairs decreases after normalization, indicating more comparable switching behaviors.

If guided solely by linguistics research (Sulminski, 2022; Beatty-Martínez et al., 2020), we would expect each of the 'conversational' data-sets to evidence a richer code-switching than its counterpart (thus obtaining a higher score). Of Figure 3.2's metrics, only Burstiness results seems to align with this intuition. Recall that a Bursti-

ness score closer to 1 reflects more 'random' code-switches, whilst negative scores reflect regular, predictable, and thus likely *more constrained*, switch patterns. Comparatively, I-index scores both the Hindi and Mandarin technical data-sets more highly than their conversational counterparts . Both M-index and Memory are inconsistent across the two languages with regards to their ranking/ordering of the conversational vs technical data. Importantly, this means that the conclusions we might draw if relying on M-index or Memory would differ from those one would obtain if using I-index: in itself, supporting our argument that none of these code-switching metrics should be considered in isolation.

3.5.2 T-index

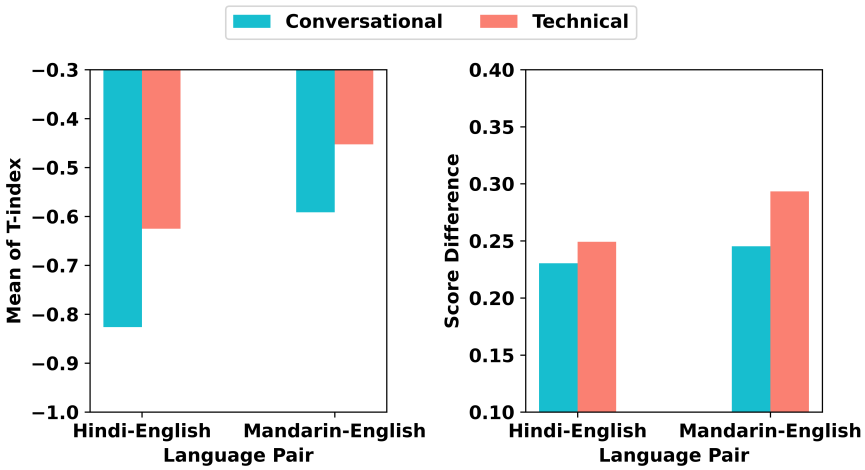


Figure 3.3: Mean translation scores of the first candidate as well as the difference between the first two candidates.

Figure 3.3's left plot presents each data-set's T-index. Since the same translation system is used for all language pairs, scores are comparable across different domains. We observe that technical domains exhibit higher scores compared to conversation domains in both languages. This can be attributed to the fact that in lecture or formal domains, individuals tend to minimize code-switching unless necessary to convey technical terms or concepts best expressed in a specific language, such as English in our case. Additionally, technical terms are highly specialized and consistently used, leading to a one-to-one mapping in our language pairs and resulting in higher first-candidate translation scores. Figure 3.3's right plot, illustrating the difference between the first two candidates' translation scores in each dataset, shows a similar trend. Score

differences are higher in technical domains than in conversational domains for both language pairs. This suggests that code-switching words in lecture datasets are more *specific* in semantic content, making MT systems more confident in selecting the first candidate when translating these words.

Unlike most of the metrics in Figure 3.2, T-index results do seem to align with expectations from the literature, with Burstiness being an exception. As stated in Section 6.1, we cannot conclude from this that T-index is a superior metric (especially given the limited number of data-set pairs examined). However, these results imply that T-index, along with the consideration of lexical features beyond mere token counts, could contribute to a more comprehensive characterization of potential code-switching variation. As such, we suggest using T-index alongside the suite of existing metrics could provide complementary information.

3.6 Conclusion

In this chapter, we illustrate a method of analyzing code-switching richness metrics grounded in the linguistics literature. By using this framework to compare existing richness metrics with our novel T-index, we conclude that relying on any single measure in isolation for a complete characterization of code-switching is not feasible. Our proposed T-index, which incorporates lexical semantic properties of code-switching words and is specific to the language pair, is a valuable addition to existing code-switching richness metrics.

Chapter 4

The Role of Part-of-speech in Code-switching

In the previous chapter, we focused on analyzing code-switching by examining stylistic variations across different corpora. In this chapter, we shift our focus to the factors that trigger code-switching, with a particular emphasis on lexical properties. Although the cognitive mechanisms underlying code-switching are complex and not yet fully understood, previous studies have identified several contributing factors. This chapter specifically investigates the influence of POS on the likelihood of bilinguals engaging in code-switching, using a detailed analysis of Spanish-English and Mandarin-English corpora. Our findings build upon prior research by confirming the statistically significant connection between part-of-speech (POS) and code-switching across these language pairs. Importantly, we observe that this relationship is strongest in close proximity to code-switching instances, gradually weakening as tokens move farther from these switching points. This chapter is adapted from content originally presented in our paper published in the findings of EACL 2024 (Chi and Bell, 2024).

4.1 Introduction

Code-switching reflects the flexibility and adaptability of individuals in their language use and therefore serves as a testing ground for research into the cognitive mechanisms of bilingual language production. The studies emerging from this exploration have shown that code-switching involves multiple layers of linguistic processing and is influenced by the properties of the words, linguistic structures and socio-interactional considerations (Gardner-Chloros, 2009; Kootstra et al., 2020).

As reviewed in Chapter 2, the triggered hypothesis suggests that certain *triggered words*, such as cognates, increase the likelihood of code-switching, underscoring the impact of specific lexical properties on language alternation. However, in this chapter, our focus shifts from exploring cognates to investigating the influence of POS, a universal linguistic feature across all languages on code-switching behavior.

Structural similarities between languages, such as word order or phrase structure, can either facilitate or constrain code-switching, depending on how closely the languages align at specific points in a sentence (Belazi et al., 1994; Berk-Seligson, 1986). Beyond linguistic considerations, socio-interactional dynamics also play a crucial role. For instance, if the preceding utterance in a conversation is code-switched, there is a higher probability that the following utterance will be code-switched as well, a trend observed in both between-person and within-person priming (Gardner-Chloros, 2009; Soto et al., 2018).

In parallel, the practical implications of understanding code-switching extend to the development of NLP techniques designed to serve multilingual communities. Recent research has attempted to integrate linguistic theories of code-switching with machine-learning approaches to train ASR and LID models. However, these theories often stem from language pairs with syntactic similarities, and their practical application is sometimes limited by the performance of dependency parsers (Berk-Seligson, 1986). While machine-learning models have shown success in specific tasks, they might be further enhanced by incorporating linguistic features from the corpus being analyzed (Adel et al., 2013b; Attia et al., 2019).

Given the critical role of word properties in bilingual language production and their potential to improve code-switching-related tasks, this chapter specifically investigates how POS tags influence code-switching behaviors. POS tags are a universal feature across languages, capturing essential grammatical categories like nouns and verbs, making them effective for analyzing linguistic structures consistently across different language datasets. examining the influence of POS tags in language pairs from both genetically related groups, such as Spanish and English, which share Indo-European roots, and genetically distinct groups, such as Mandarin and English, which belong to the Sino-Tibetan and Indo-European families respectively, this study aims to provide deeper insights into the factors that facilitate code-switching.

4.2 Analyzing lexical triggering

In the context of analyzing lexical triggering in natural language corpora, the χ^2 goodness-of-fit test (hereafter referred to as the χ^2 test) is frequently employed to determine whether there is a significant association between the occurrence of two lexical items, suggesting a potential triggering relationship. Analyses have consistently shown that code-switching occurs more frequently when language-ambiguous words, particularly cognates, are nearby (Clyne, 1967; Broersma and De Bot, 2006; Kootstra et al., 2020). These findings align with the well-established idea that cognates lead to the simultaneous activation of both languages in the speaker's mind, thereby increasing the likelihood of using both languages within a single utterance (Van Assche et al., 2012; Soares et al., 2019). However, it is important to note that not all language pairs include cognates, and identifying cognates when they do exist often requires linguistic expertise. Given that a significant portion of code-switching triggers are nouns and proper nouns (Broersma and De Bot, 2006), researchers have turned their attention to the role of POS in determining code-switching constraints (Soto et al., 2018). Inspired by the same study, which demonstrate a dependency between POS and code-switching, our work extends this line of inquiry. In this chapter, we propose a more robust hypothesis: this dependency remains significant when considering the distribution of both POS and code-switching across word positions, with its strength diminishing as the POS moves further from the switching points.

4.2.1 χ^2 test

We begin by outlining the process of the χ^2 test, followed by discussing our modifications to the underlying dependence assumption. This process includes collecting counts of lexical items to construct a contingency table. For example, to determine if there is a statistical relationship between the presence of cognates and code-switching within an utterance, we start by constructing a 2×2 contingency table as shown in Table 4.1. This table consists of four cells, defined as follows:

- O_{11}: The number of utterances where both code-switching and cognates are present.
- O_{12}: The number of utterances where code-switching occurs, but cognates do not.

	cognates	no cognates
code-switching	O_{11}	O_{12}
monolingual	O_{21}	O_{22}

Table 4.1: A 2×2 contingency table which compares the observed frequencies of utterances containing code-switching and cognates.

- O_{21}: The number of utterances where cognates are present, but code-switching does not occur.
- O_{22}: The number of utterances where neither code-switching nor cognates are present.

To test the independence of these occurrences, the expected frequency for each cell is calculated under the assumption that cognates and code-switching occur independently. The expected frequency E_{ij} for each cell is given by:

$$E_{ij} = \frac{\sum_i O_{ij} \times \sum_j O_{ij}}{\sum_{i,j} O_{ij}} \tag{4.1}$$

Once both the observed (O_{ij}) and expected E_{ij} counts are determined for each cell, the χ^2 statistic is computed using the following formula:

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4.2}$$

This χ^2 statistic is then compared to a critical value from the χ^2 distribution table, which is determined based on the degrees of freedom (which, in a 2×2 table, is 1) and the chosen significance level (typically 0.05). If the χ^2 statistic exceeds the critical value, the null hypothesis of independence is rejected, indicating a statistically significant association between the presence of cognates and code-switching.

It is important to note, however, that a significant association found through the χ^2 test only implies that the occurrence of one lexical item may be related to or potentially trigger the occurrence of the other. This test identifies correlation, not causation.

4.2.2 Our triggering hypothesis

In their work, Soto et al. (2018) defined code-switched words as those immediately following code-switching points and demonstrated a strong statistical association between POS tags and both the words preceding code-switches and the code-switched

Table 4.2: POS % in different positions

Position	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	SCONJ	VERB
Beginning	2.12	5.89	27.69	0.70	3.45	2.86	7.24	7.71	1.75	0.64	19.74	5.07	2.25	12.89
Mid	3.14	5.46	16.11	1.83	1.30	4.42	1.03	14.87	3.09	4.85	14.07	5.91	1.21	22.73
End	3.81	2.51	12.97	0.47	0.94	1.19	3.26	27.39	2.90	8.88	7.54	5.02	0.65	22.48

Table 4.3: Position % for each POS

Position	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	SCONJ	VERB
Beginning	6.81	11.29	16.31	4.43	23.28	7.26	38.60	5.00	5.94	1.31	14.09	8.82	17.89	5.92
Mid	80.96	83.92	76.07	92.61	70.39	89.74	44.04	77.29	84.22	80.37	80.53	82.46	76.98	83.77
End	12.23	4.79	7.62	2.96	6.33	3.00	17.36	17.71	9.84	18.32	5.37	8.72	5.14	10.31

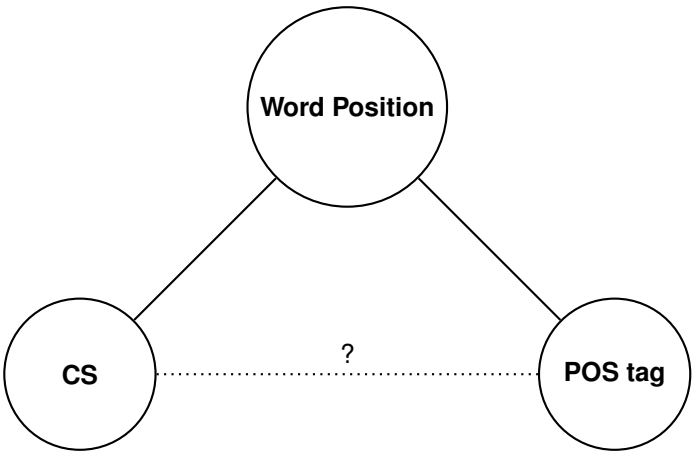
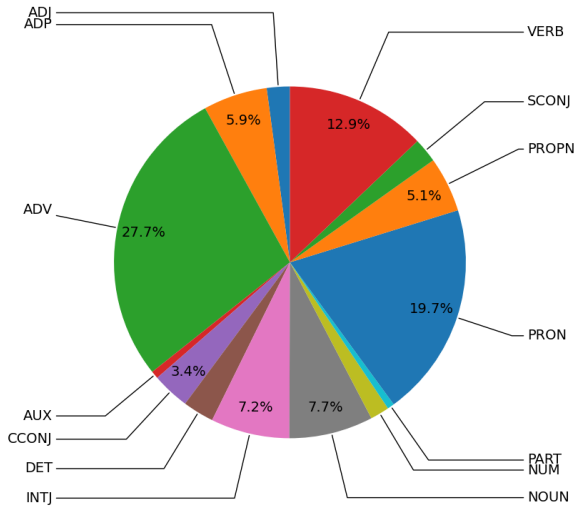


Figure 4.1: An undirected graph depicting the hypothetical connections between word position, CS, and POS.

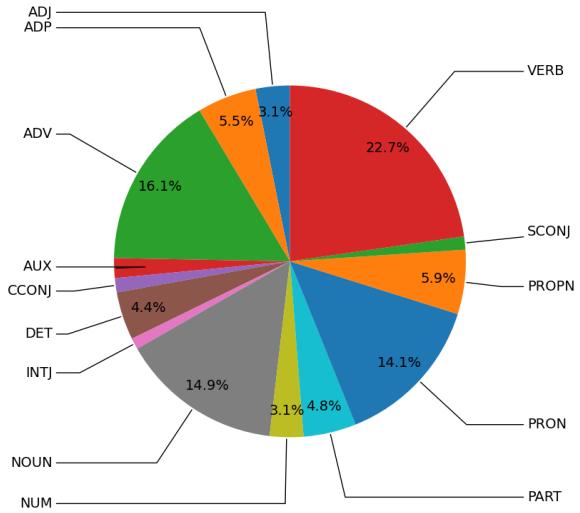
words themselves. However, this definition presents a potential issue: while the χ^2 test may confirm a dependency between POS tags and code-switched words, this relationship, as shown in Figure 4.1, could be influenced more by word positions within a sentence than by the intrinsic nature of code-switching. Specifically, code-switching points are not uniformly distributed across sentence positions and, in particular, never occur at the beginning of a sentence.

To clarify this issue, consider an extreme scenario where a particular POS tag, predominantly appears at the start of a sentence. This positional bias would make it less likely for that POS tag to be associated with code-switching simply because code-switches do not occur at sentence-initial positions. Even if that same POS tag were occasionally involved in code-switching at other positions, its overall statistical association with code-switching could appear misleadingly low due to its positional distribution. To illustrate this point, we plot the distribution of different POS tags at the beginning, middle, and end of sentences for SEAME dataset, as well as the distribution of these positions for each POS tag, in Figure 4.2 and 4.3, with detailed numbers in Table 4.2 and 4.3. Here, we define the beginning as the first word of the sentence, the end as the last word, and the middle as all remaining words in between. This visualization demonstrates that POS tags have distinct distributions across sentence positions, which can impact the results of significance tests. Therefore, while previous studies have highlighted the relationship between POS tags and code-switching, they may not have fully accounted for the positional effects.

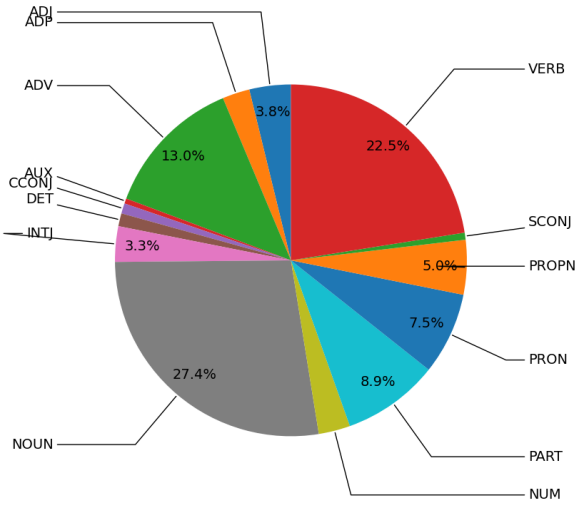
In light of these considerations, we refine our hypothesis to assert that certain POS tags maintain a statistically significant relationship with code-switching and the sur-



(a) POS distribution at the start of the sentence.



(b) POS distribution in the middle of the sentence.



(c) POS distribution at the end of the sentence.

Figure 4.2: POS distribution at different positions in the sentence.

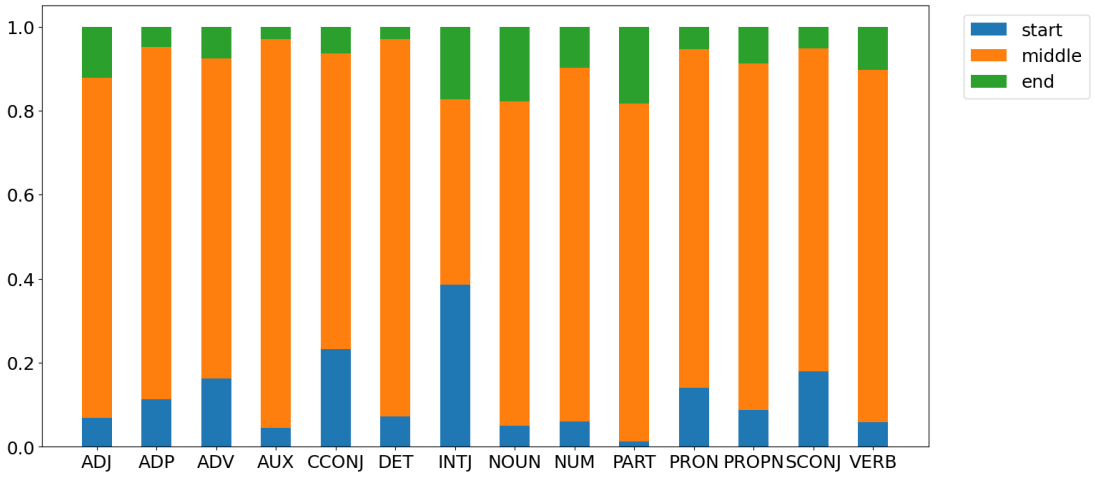


Figure 4.3: The distribution of these positions for each POS tag

rounding words, even after controlling for word positions within a sentence. Moreover, we posit that this relationship weakens as the words in question move farther from the code-switching point.

4.3 Methodology

4.3.1 Corpus

Two language pairs are investigated in this work: Spanish-English and Mandarin-English. For the Spanish-English code-switching analysis, we utilize the publicly available Bangor-Miami (BM) corpus, a rich resource that captures conversational speech from bilingual speakers in the Miami, Florida area (Deuchar et al., 2014). This corpus reflects natural, spontaneous conversations among speakers who frequently alternate between Spanish and English. The Bangor-Miami corpus was originally annotated using the native tagset provided by the Bangor Autoglosser (Donnelly and Deuchar, 2011). However, to facilitate cross-linguistic comparisons and ensure consistency across different language pairs, we use a version of the corpus that has been re-annotated with Universal POS (UPOS) tags (AlGhamdi et al., 2016). UPOS tags are part of the Universal Dependencies (UD) framework, which offers a standardized POS tagging system that is applicable across different languages. This universal annotation scheme allows for more straightforward comparisons of linguistic features across languages with different grammatical structures and typologies. The full UPOS tagset is presented in Table 4.4 and the last three categories (PUNCT, SYM, X) are excluded

because they do not carry lexical content or grammatical function in the same way that other POS categories do.

For Mandarin-English experiments, we explore the SEAME corpus which comprises conversations and interviews with bilingual speakers from Malaysia and Singapore (Lyu et al., 2010) as described before in Section 3.4.1. As there is no existing tagged version of SEAME available, we annotate it utilizing the Spacy toolkit, following the methodology in (Bhattacharya et al., 2023). Specifically, we first use taggers of both relevant languages to obtain two sets of POS tags for each sentence. Next, we perform token-level language identification to determine the language of each word in the sentence. Based on this identification, we select the appropriate POS tag from the two tagger outputs to create a final sequence of POS tags for each sentence. The distribution of POS tags in both corpora is detailed in Table 4.5.

Table 4.4: Universal POS Tagset

Category	Tag	Included in Analysis
Adjective	ADJ	Yes
Adposition	ADP	Yes
Adverb	ADV	Yes
Auxiliary verb	AUX	Yes
Coordinating conjunction	CONJ	Yes
Subordinating conjunction	SCONJ	Yes
Determiner	DET	Yes
Interjection	INTJ	Yes
Noun	NOUN	Yes
Numeral	NUM	Yes
Proper noun	PROPN	Yes
Pronoun	PRON	Yes
Particle	PART	Yes
Verb	VERB	Yes
Punctuation	PUNCT	No
Symbol	SYM	No
Other	X	No

	SEAME	BM		SEAME	BM
ADJ	3.11	4.10	ADP	5.24	6.97
ADV	16.94	8.11	AUX	1.59	3.25
CONJ	1.47	4.40	DET	3.97	8.81
INTJ	1.71	5.94	NOUN	15.42	11.04
NUM	2.95	1.51	PART	4.87	2.58
PRON	14.05	15.98	PROPN	5.73	2.49
SCONJ	1.26	3.88	VERB	21.70	20.00

Table 4.5: POS distribution (shown in percentage) within each dataset in Bangor-Miami and SEAME corpus

4.3.2 Experiments

4.3.2.1 Code-switched words

The relationship between code-switching and POS tags is analyzed using the χ^2 test for independence, with Yates' correction applied where necessary to adjust for small expected frequencies. To account for the influence of word positions within a sentence, words are categorized into three positional groups: Start, Mid, and End. In constructing the contingency tables that record the counts of all POS tags and their association with code-switching instances, we compute the expected distribution under the null hypothesis, which assumes that, *given specific word positions*, code-switching and POS are independent.

The calculation of the expected count of words being both code-switched and tagged as a specific POS, such as ADJ (adjective), is represented by $E(CS,ADJ)$ in Equation 4.3. Here, the variable i denotes the word position (Start, Mid, or End), and P_i represents the probability of a word being code-switched or tagged as ADJ at position i. The term N_i represents the total number of words at position i. This framework allows us to estimate how often code-switching should occur with specific POS tags purely by chance, under the assumption of independence.

$$\begin{aligned} E(CS,ADJ) &= \sum_{i \in s,m,e} P_i(CS,ADJ)N_i \\ &= \sum_{i \in s,m,e} P_i(CS)P_i(ADJ)N_i \end{aligned} \tag{4.3}$$

It is important to highlight that the hypothesis proposed by (Soto et al., 2018),

	ADJ	ADP	ADV	AUX	CONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	SCONJ	VERB
BM	-	-	-	√√ ↓	√√ ↑	√ ↓	√√ ↑	√√√ ↑	-	√√ ↓	√√ ↑	√ ↑	√ ↑	√√√ ↓
SEAME	√√√ ↑	√√√ ↓	√√ ↓	√√√ ↓	√√ ↑	√√√ ↓	√√ ↑	√√√ ↑	√√√ ↓	√√√ ↑	-	√√√ ↑	√ ↓	√√√ ↓

Table 4.6: The significance of running χ^2 statistical tests on each group of POS tags and code-switching words. One $\surd$ indicates $p < 0.01$, two indicate$p < 10^{-36}$ and three indicate $p < 10^{-100}$. $\uparrow$ and $\downarrow$ represent whether they more often or less often occur at the code-switching word.

which does not account for word positions, can be viewed as a special case within this broader framework. In their model, words are assumed to be uniformly distributed across the Start, Mid, and End positions, implying an equal likelihood of appearing at any point within a sentence. Our approach, by contrast, adjusts for the actual distribution of words across these positions.

4.3.2.2 Neighbour words

Previous research has primarily focused on the POS that immediately precede and follow code-switching words, using distribution analysis and χ^2 tests to evaluate their associations. However, due to the complex syntactic relationships within sentences, when examining code-switching holistically, the impact of various POS tags of code-switching words on neighbouring words may result in intricate mutual offset or amplification effects. Since these analyses rely on count-based data, detecting significant changes or patterns can be particularly challenging.

To address these complexities, we propose a novel approach that categorizes code-switching based on the POS of the code-switched words themselves. For each category, we analyze the distribution of POS tags in the words immediately before and after the code-switching word, as well as those located two to four words away. By comparing these distributions to the overall POS distribution for each category, we aim to isolate the variations that are directly attributable to code-switching behaviors, rather than those arising from general syntactic structure.

4.4 Results

4.4.1 CS words

Table 4.6 presents the results of χ^2 statistical tests performed on groups of POS tags and their association with code-switched words. A single checkmark ($\surd$) denotes a significance level of $p < 0.01$, two checkmarks indicate $p < 10^{-36}$, and three checkmarks signify $p < 10^{-100}$. The arrows ($\uparrow$ and $\downarrow$) represent whether these tags occur more or less frequently at code-switched words based on our observations than would be expected if code-switching occurs independently of POS. Although the datasets used in this study are not extremely large, the extremely small p-values (e.g., $p < 10^{-100}$) observed suggest that the association between certain POS tags and code-switching is highly systematic. These values indicate that the observed distributions deviate very strongly from what would be expected if POS and code-switching were independent. However, it's important to note that statistical significance does not necessarily equate to practical importance. The direction of the effect provides more interpretable insight into the role of POS in code-switching behavior than the raw p-value alone.

The analysis uncovers a strong statistical relationship for most POS tags, reinforcing the idea that certain syntactic categories are more amenable to code-switching than others. Notably, this contrasts with the findings of Soto et al. (2018), where certain tag pairs—such as CONJ and SCONJ, or PRON and NOUN—showed markedly different statistical behaviors with respect to code-switching in the BM corpus. In our results (Table 4.6), these categories show more comparable significance levels and directionality, indicating more uniform behavior across functionally similar POS tags. We attribute this to differences in how word position is handled that our analysis considers whether words are located at or near switch points. For instance, tags like PRON and CCONJ, which frequently appear at sentence-initial positions, may be more likely to co-occur with switches simply due to structural constraints. Our treatment of position may therefore reduce such positional biases, leading to more consistent behavior across tags.

Another significant observation is that the SEAME generally shows a stronger statistical relationship compared to the BM corpus. This suggests that Mandarin and English have more diverse syntactic structures compared to Spanish and English, resulting in less flexibility for code-switching. Additionally, we observe a notable infrequency of switches involving VERB or AUX tags in both language pairs. This can be explained by the fact that verbs are typically preceded by pronouns and re-

quire agreement in person and number, which imposes constraints on the possibility of code-switching at these points.

4.4.2 Neighbour words

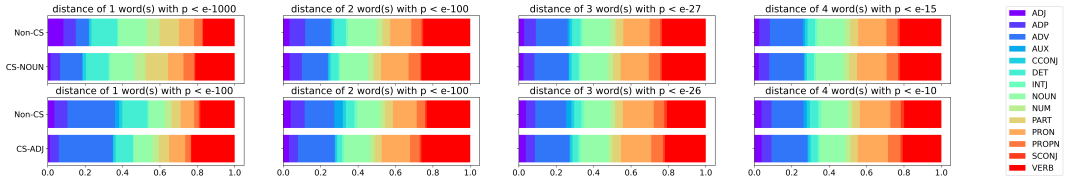
Figure 4.4 illustrates the distribution of POS tags for words located 1-4 tokens away from code-switching points, focusing specifically on the NOUN and ADJ categories for clearer visualization of variation. The full POS distribution, including all categories, is shown in Figures 4.5, 4.6, 4.7, and 4.8. While the text labels in these figures are small due to the number of categories and space constraints, the goal of these visualizations is not to read individual labels, but to draw attention to broader distributional patterns—particularly differences between code-switched and non-code-switched contexts, and how those differences change with distance from the switch point. The figures reveal that, in the SEAME corpus, POS distribution differences diminish as words move further away from code-switching points, suggesting a localized effect. In contrast, the BM corpus shows a sharper contrast at the immediate switch point, but less distinction further away, with no statistical significance observed beyond one token. Across both corpora, preceding words exert greater influence than following words, consistent with findings by Soto et al. (2018).

In SEAME, even the largest p-value among these tests is below 10^{-3}, underscoring the strength of local syntactic and contextual constraints. We also observe that ADJ occurs less frequently before switched NOUNs, supporting the tendency for noun phrases to be switched as a unit. Similarly, VERB and AUX appear more frequently before switched NOUNs, aligning with syntactic structures in which these elements precede the noun.

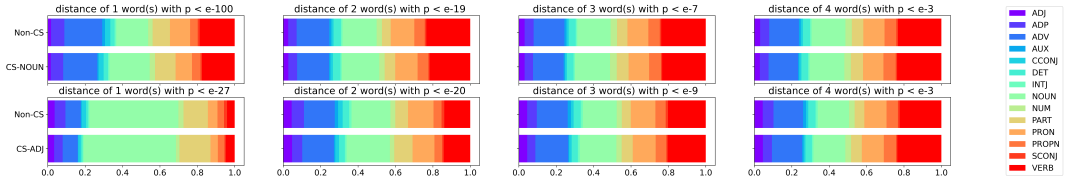
For the BM corpus, certain code-switching categories (e.g., PART) have small sample sizes. As a result, the χ^2 test results are not provided for these categories, as the test requires a minimum expected cell count in each category (commonly 5), and this assumption is violated when some categories have extremely low or zero counts.

4.5 Conclusion

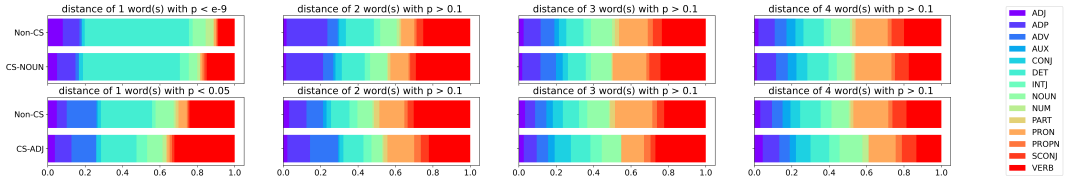
In this chapter, we have extended prior research by incorporating the influence of word positions into the analysis of the relationship between POS and code-switching across two language pairs. Our results robustly confirm a statistically significant connection



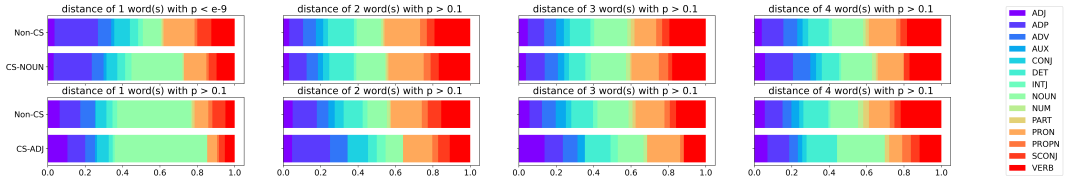
(a) POS of words positioned at 1-4 words ***before*** code-switching words tagged as ***NOUN*** (top) and ***ADJ*** (bottom) in ***SEAME***



(b) POS of words positioned at 1-4 words ***after*** code-switching words tagged as ***NOUN*** (top) and ***ADJ*** (bottom) in ***SEAME***



(c) POS of words positioned at 1-4 words ***before*** code-switching words tagged as ***NOUN*** (top) and ***ADJ*** (bottom) in ***BM***



(d) POS of words positioned at 1-4 words ***after*** code-switching words tagged as ***NOUN*** (top) and ***ADJ*** (bottom) in ***BM***

Figure 4.4: The visualization of the distribution of POS for words positioned at 1-4 words away from code-switching points, specifically those categorized as NOUN and ADJ in both corpora Scientific notation in the figure (e.g., `1e-10`) is equivalent to 10^{-10} as used in the main text.

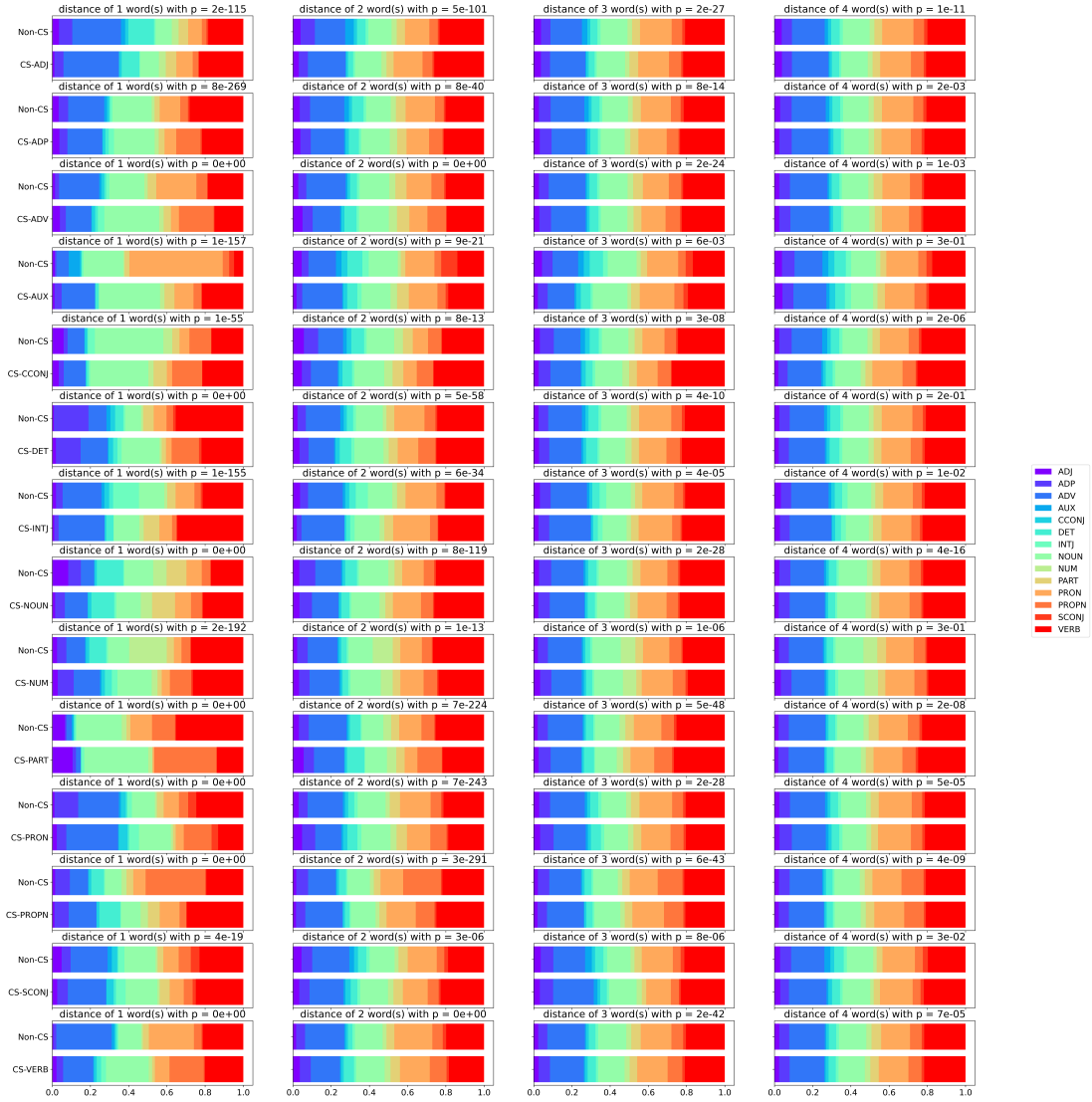


Figure 4.5: The visualization of the distribution of POS for words positioned at 1-4 words before code-switching points in SEAME.

between POS and code-switching, with Mandarin-English exhibiting a higher significance level. This suggests that a more diverse syntactic structure in a language pair leads to less flexibility in code-switching. By categorizing code-switched words and examining the distribution of neighbouring POS, we found, as expected, that the relationship is strongest in close proximity to code-switching instances and gradually diminishes as words move farther from these points. These insights can be instrumental in refining the design of models that account for code-switching behaviors.

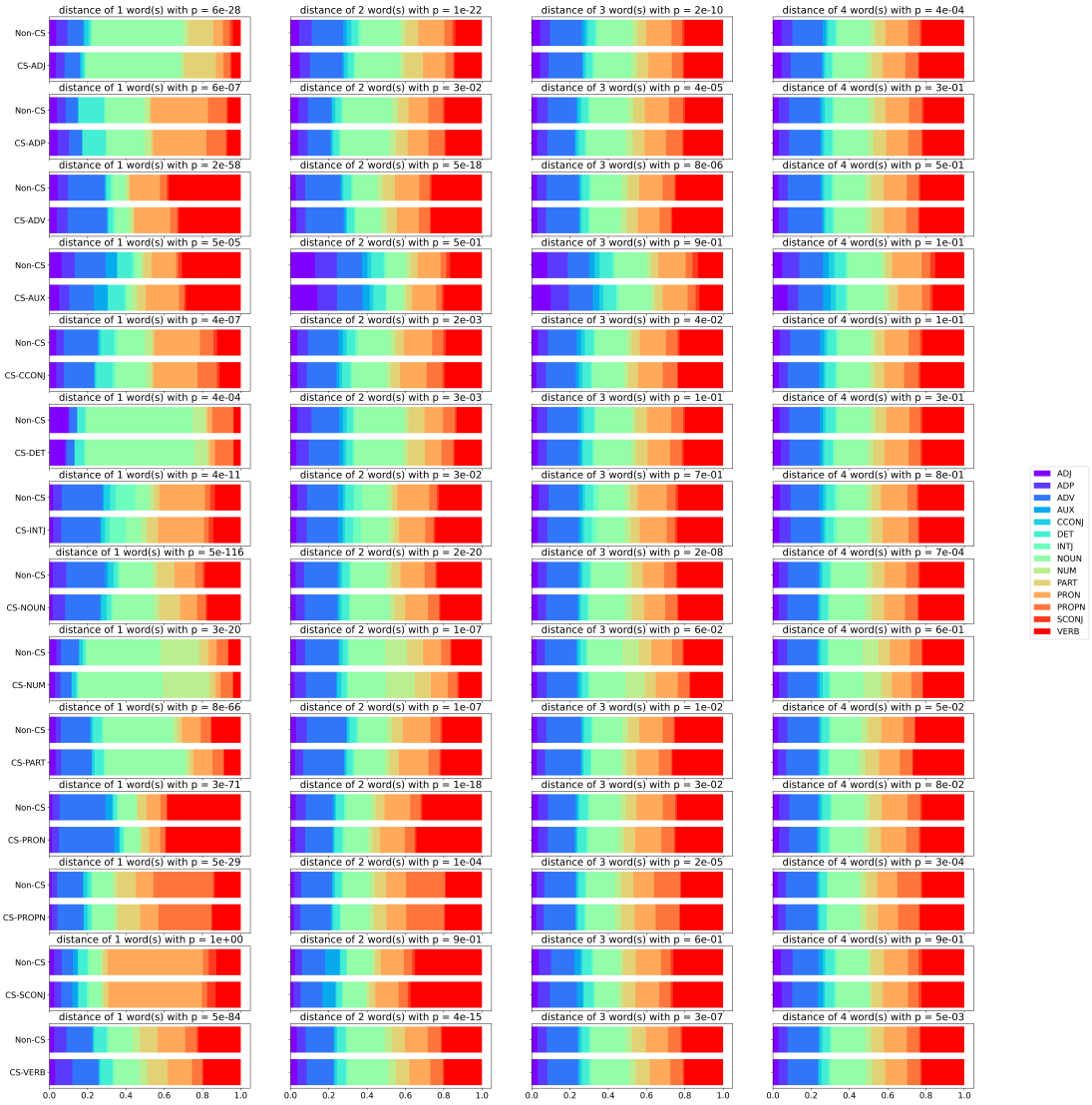


Figure 4.6: The visualization of the distribution of POS for words positioned at 1-4 words after code-switching points in SEAME.

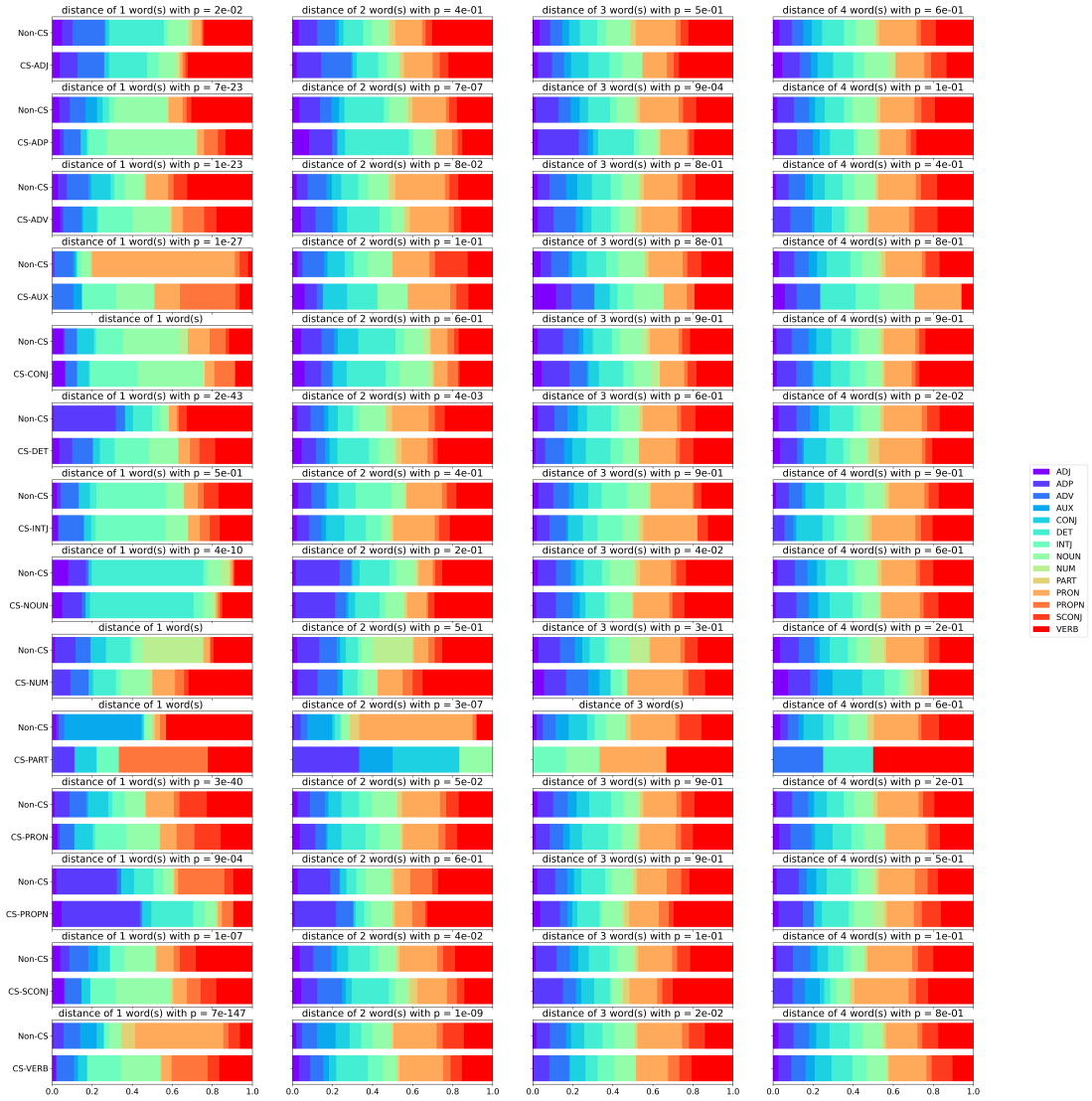


Figure 4.7: The visualization of the distribution of POS for words positioned at 1-4 words before code-switching points in BM.

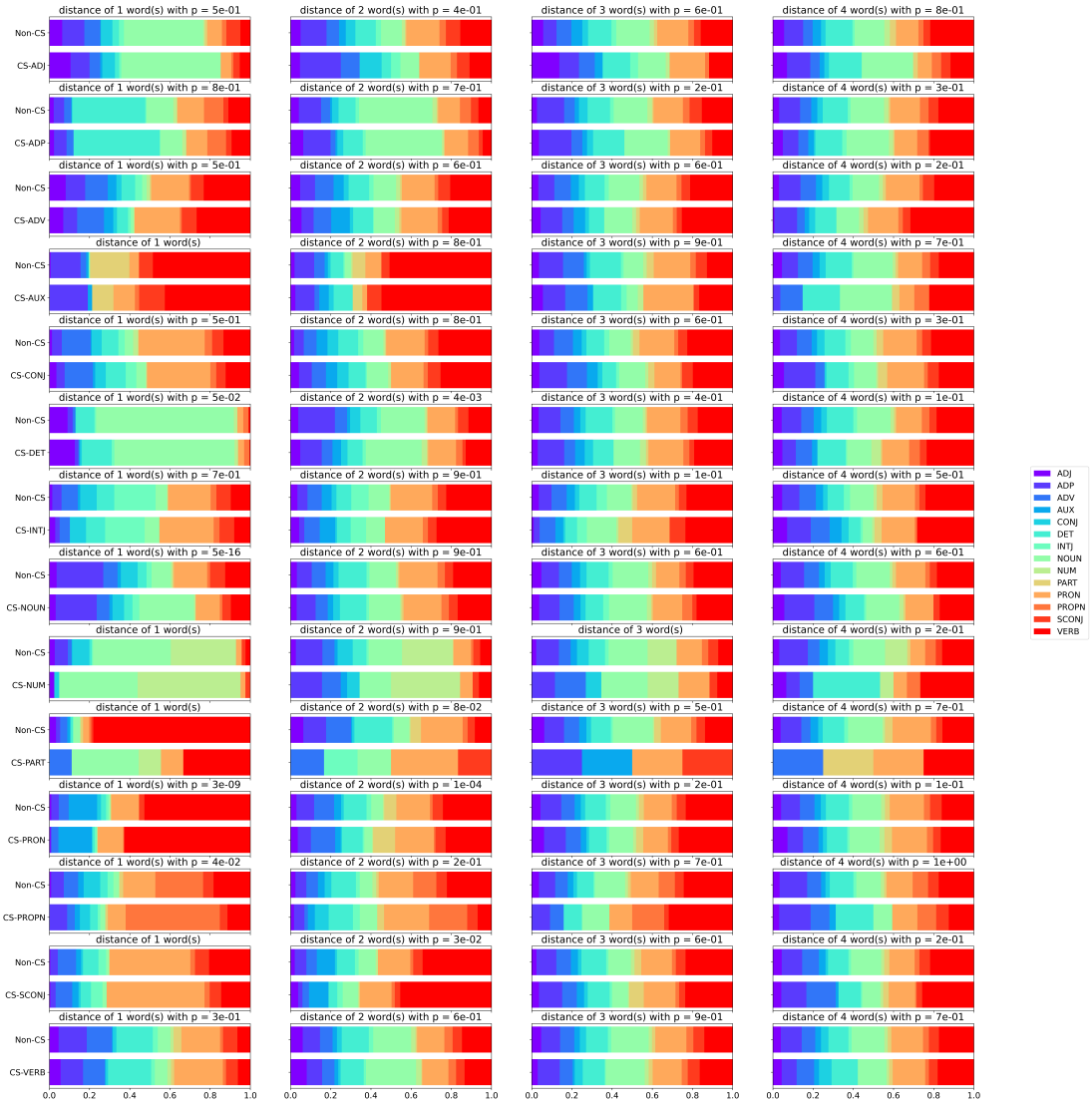


Figure 4.8: The visualization of the distribution of POS for words positioned at 1-4 words after code-switching points in BM.

Chapter 5

Linguistic theory based Text generation

Previous chapters have focused on understanding code-switching behaviors. In the upcoming chapters, we will explore text generation approaches from both linguistic theory-based and machine learning-based perspectives, with the goal of improving models using the generated texts. As previously mentioned, code-switching presents a significant challenge for ASR systems due to the relative scarcity of code-switching speech and text data, even when the individual languages are well-resourced. This chapter proposes addressing this challenge by applying Equivalence Constraint theory to generate more realistic code-switching text, which is crucial for effective language modeling in ASR. Focusing on English-Spanish code-switching, our findings indicate that incorporating Equivalence Constraint theory and POS labeling into text generation not only enhances the realism of the generated texts but also leads to a 2% improvement in ASR performance. This chapter is adapted from content originally presented in our paper published in the proceedings of COLING 2022 (Chi and Bell, 2022).

5.1 Introduction

With the rising popularity of voice assistants and translation applications, ASR systems have become increasingly integrated into daily life. Given the prevalence of bilingualism in many countries[1], and the common occurrence of code-switching in everyday conversations, there is growing interest in developing ASR systems that can handle such scenarios effectively. As discussed in Chapter 2, although different language

[1] https://www.uottawa.ca/clmc/55-bilingual-countries-world

pairs may exhibit varying extents or types of code-switching, these can generally be categorized into three types: inter-sentential, intra-sentential, and tag switching. Inter-sentential switching occurs at the sentence or clause boundary, intra-sentential switching happens within the sentence or clause, and tag switching involves inserting a tag phrase in a different language. In this chapter, our focus is on intra-sentential switching, which poses a greater challenge than the other types due to the increased acoustic variability of mixed languages within a sentence, as compared to across sentences (Li et al., 2019).

Developing a code-switched ASR system presents challenges from both linguistic and computational perspectives. On the linguistic side, code-switching is a complex phenomenon influenced by multiple factors, making its prediction challenging. As discussed in Chapter 2, extensive research has identified various motivations behind code-switching, including compensating for language proficiency gaps, expressing solidarity or emotions, discussing specific topics, and signaling social identity (Grosjean, 1982; Holmes and Wilson, 2017; Leung, 2006). On the computational side, ASR systems typically require large amounts of transcribed data, which is readily available only for a limited number of the world's approximately 7,000 languages. The data scarcity is even more pronounced for code-switching scenarios, particularly when they involve one or more low-resource languages (Austin and Sallabank, 2011).

Motivated by these challenges, this chapter proposes a novel code-switched ASR framework that leverages established linguistic theories. The effectiveness of this approach is demonstrated using Spanish-English conversational code-switching data from the Bangor-Miami (BM) corpus (Deuchar, 2011). Through this work, we show that incorporating phonological and syntactic information can significantly improve the performance of language modeling and ASR in code-switched environments.

5.2 Related work

There have been numerous efforts to address the challenge of modeling code-switched speech within conventional hybrid ASR systems. These systems integrate acoustic models with language models trained on extensive text datasets to predict word sequences accurately. Earlier in Section 2.5.2, we reviewed work focused on the acoustic and pronunciation aspects. In this section, we shift our attention to the research direction of text generation. Given the significantly larger availability of monolingual text compared to code-switched text, generating code-switched text through rule-based or

machine learning-based approaches applied to parallel monolingual texts has become an important research approach. This chapter emphasizes rule-based approaches, while the next chapter will delve into machine learning-based methods.

Rule-based approaches use explicit rules or strategies for generating code-switched text. These rules might be based on linguistic constraints, the frequency of words, POS, or other heuristics that guide the insertion or substitution of words between languages. For example, Shen et al. (2011) employed an English-Chinese dictionary to translate portions of Chinese sentences into English, taking into account the frequency of the English words to generate code-switched text. Similarly, Vu et al. (2012) implemented a method where portions of monolingual sentences were replaced with their translations only if the translated segment met a minimum threshold of occurrence. Additionally, the replacement was further constrained by the presence of trigger words or specific POS immediately preceding the segment. In the semi-supervised framework proposed by Gupta et al. (2020), code-switched texts are initially generated by replacing only noun phrases, such as named entities, with their counterparts from the target language (Piergallini et al., 2016).

Instead of relying solely on the frequency of words or parts of speech, another approach involves predicting code-switching points first and then determining which segments to switch. For instance, Solorio and Liu (2008a) treats each word boundary as a potential switching point and employs a classifier trained on a combination of word-level features, including the word itself, its lemma, POS, and its position relative to phrase constituents. During the text generation phase, this method begins by adding tokens from L1 until a code-switching point is predicted. Once this point is reached, the next tokens are drawn from the L2, and the process continues, switching back and forth based on the classifier's predictions. There is also research focused on code-switching prediction tasks that do not aim specifically at text generation (Ostapenko et al., 2022; Calvillo et al., 2020).

Instead of focusing solely on lexical properties, research has also delved into linguistic theories that emphasize the syntactic structure of code-switching for text generation, a focus that closely aligns with the work presented later in this chapter. Three prominent theories in this area are the Matrix Language Frame (MLF) theory, the Equivalence Constraint (EC) theory , and the Functional Head Constraint (FHC) theory. The MLF theory posits that the matrix language provides an abstract grammatical framework into which an embedded language is inserted (Myers-Scotton, 1997). This theoretical framework has been used to determine which segments of text can be effec-

tively replaced to generate code-switched text. For example, Lee et al. (2019) applied MLF theory to decide which segments could be replaced, thereby augmenting real code-switched text to achieve improved performance. On the other hand, the EC theory suggests that code-switching occurs at points where the grammatical constraints of both languages are simultaneously satisfied. This theory was implemented by Pratapa et al. (2018) using parse trees to generate code-switched sentences, ensuring that the switching points adhered to grammatical rules. Additionally, Winata et al. (2019) simplified the application of EC theory by allowing only non-crossing alignments to be switched. Lastly, the FHC theory argues that switching between a functional head and its complement is not permissible due to the strong syntactic relationship between these elements. Li and Fung (2014) incorporated it into a weighted finite state transducer (WFST) framework to restrict certain paths in the code-switching process, ensuring that syntactic rules were followed.

5.3 Methodology

Our proposed ASR framework is composed of two integral components: phone mapping for acoustic modeling and code-switched text generation for language modeling. The phone mapping component is designed to handle the acoustic variability that arises from the blending of two languages. On the other hand, the text generation component focuses on creating realistic code-switched text guided by EC theory, which is crucial for developing robust language models.

5.3.1 Phoneme mapping

We use the standard International Phonetic Alphabet (IPA) as the basis for our acoustic modelling units. As English and Spanish only have partly different inventories (Edwards, 1992; Goldstein, 2000), instead of treating them as completely two phone sets, we merge them according to their phonological features (Mortensen et al., 2016). The features are a set of global attributes as shown in Figure 5.1, which describes the characteristic of a speech sound, such as whether it is produced with nasal airflow or if the vocal folds vibrating during the production. After representing the feature with vectors, where each attribute can be negative or positive, we compute the hamming edit distance between each pair of phonemes. In this way, we map each Spanish-only phoneme to its nearest English equivalent based on phonological feature similarity, as

shown in Table 5.1. This approach helps by clustering phones from different languages when there is only limited speech data available.

	syl	son	cons	cont	delrel	lat	nas	strid	voi	sg	cg	ant	cor	distr	lab	hi	lo	back	round	tense	long
/p/	-	-	+	-	-	-	-	0	-	-	-	+	-	0	+	-	-	-	-	0	-
/p^h/	-	-	+	-	-	-	-	0	-	+	-	+	-	0	+	-	-	-	-	0	-
/p^j/	-	-	+	-	-	-	-	0	-	-	-	+	-	0	+	+	-	-	-	0	-
/p^{hj}/	-	-	+	-	-	-	-	0	-	+	-	+	-	0	+	+	-	-	-	0	-

Figure 5.1: Illustration of IPA segments and feature vectors. Adapted from (Mortensen et al., 2016)

Spanish Phoneme	Closest English Phoneme
/r/	/ɹ/
/x/	/h/
/β/	/v/
/ɣ/	/g/
/jˈ/	/j/
/ʎ/	/l/

Table 5.1: Mapping of Spanish-only phonemes to their closest English equivalents based on phonological similarity. /ɹ/ here approximates the English alveolar approximant, which is not available in the current fonts.

5.3.2 Code-switched text generation

5.3.2.1 Parallel text generation

We use the Google translate API[2] to generate parallel English and Spanish text. The API not only supports translation between languages but also often translates code-switched sentences into one target language while generally preserving segments in the original language. However, it may occasionally modify these segments, though human inspection typically confirms that the segments are mostly retained. We receive one translated sentence for each monolingual text in the corpus, while for each code-switched sentence, we obtain translations in both languages. As the translation quality varies across the sentences under manual inspection, we use Pseudo Fuzzy-match

[2]Accessed in August 2022

Score (PFS) shown in Equation 5.1 to filter any translation pairs whose PFS is less than 0.6 (He et al., 2010; Pratapa et al., 2018). s here is the monolingual source sentence, we forward translate s to target t, then reverse translate the target t into pseudo source s'.

$$PFS = \frac{EditDistance(s, s')}{max(|s|, |s'|)} \tag{5.1}$$

5.3.2.2 Constituency parse generation

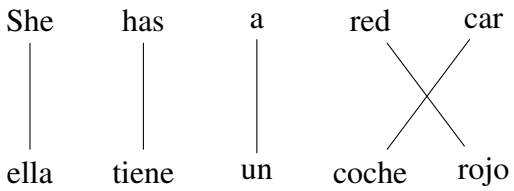
To generate word-level alignments between parallel sentences, we employ the *fast_align* toolkit, an unsupervised aligner that efficiently computes alignments by leveraging a simple but effective probabilistic model (Dyer et al., 2013). Following the methodology of Pratapa et al. (2018), we first generate parse trees for the English text using the Berkeley Neural Parser (Kitaev and Klein, 2018). This parser, which utilizes an encoder-decoder framework with an encoder incorporating factored self-attention, is adept at capturing complex syntactic structures. The alignments obtained from *fast_align* are then used to create corresponding parse trees for the Spanish sentences. This step facilitates syntactic comparisons and manipulations between the languages, allowing for a deeper understanding of structural similarities and differences. When alignments are one-to-one and the sentence lengths are equal, the tree can be directly projected, as illustrated in Figure 5.2. However, for more realistic cases with alignment mismatches, we use specific strategies to handle two major challenges:

1. Null alignments: If an English word does not align to any Spanish token (i.e., a null alignment), we insert a placeholder node (e.g., < >) into the Spanish side to preserve the structure of the tree and maintain alignment positionally. This allows the projection mechanism to remain well-formed even when content is dropped in translation, as shown for the unaligned pronoun *She* in Figure 5.3.

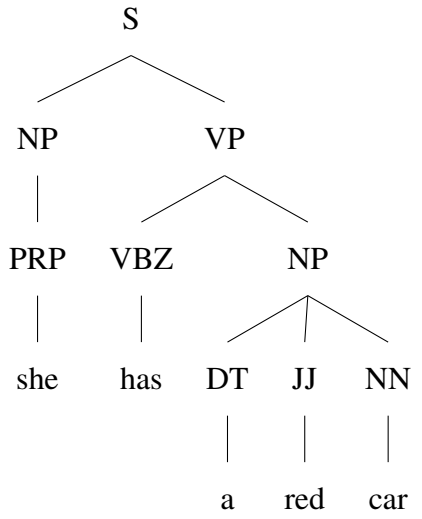
2. Many-to-one alignments: When multiple English words align to a single Spanish word or phrase (e.g., "will have" $\rightarrow$ "tendrá"), we identify the smallest subtree in the English parse that spans all aligned words. We then collapse this subtree into a single node corresponding to the target Spanish word. If the aligned English tokens belong to different branches of the tree (e.g., modal and verb), we flatten the relevant portion of the tree to form a single constituent. This ensures that syntactic integrity is preserved while allowing for projection under structural asymmetry.

This alignment-aware projection approach enables syntactic transfer between English and Spanish while accounting for common translation divergences. Though such modifications may slightly distort the syntactic structure, their impact is limited, as

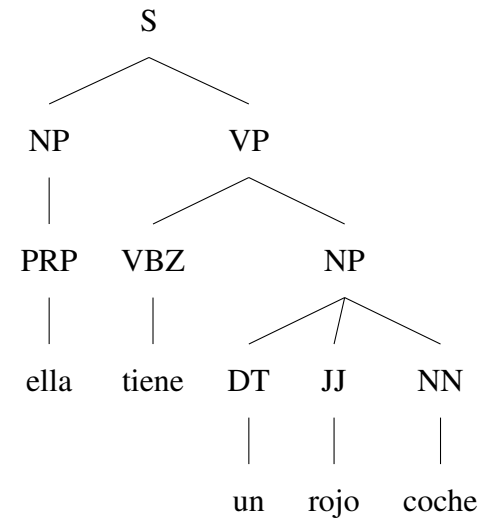
the close grammatical alignment between English and Spanish reduces the need for major alterations. While our implementation is designed for robustness in downstream processing, it does not resolve deeper structural mismatches (e.g., clause reordering).



(a) Word alignment between the English sentence *She has a red car* and its Spanish translation *ella tiene un rojo coche*.



(b) English parse tree

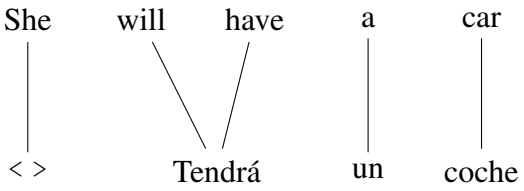


(c) Direct mapping of Spanish words onto the corresponding English parse tree

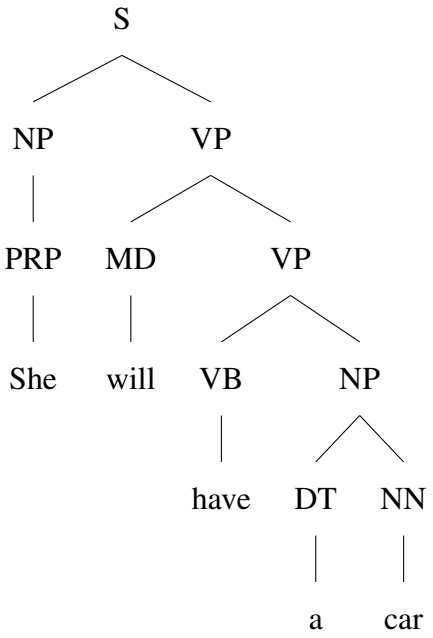
Figure 5.2: English parse tree and its equivalent Spanish parse tree.

5.3.2.3 Equivalence Constraint theory

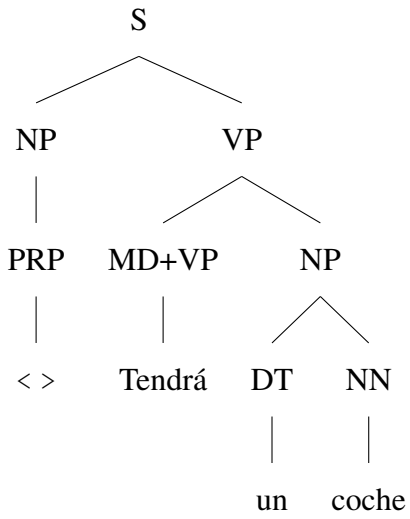
EC theory claims that, in general, *"Codes will switch at points where the surface structures of the languages map onto each other"* (Sankoff and Poplack, 1981). For example, in English and Spanish, code-switching cannot happen within possessive phrases or noun/adjective clauses because the grammatical structures are different and thus reject the transfer. Figure 5.4 illustrates this constraint using constituency parse trees. As noted previously, the order of the leaf nodes for *rojo coche* differs from the typical word order; *coche* cannot be directly substituted. Instead, to maintain structural congruence across the languages, both *rojo* and *coche* are collapsed into the same node. This ap-



(a) Word alignment between the English sentence *She will have a car* and its Spanish translation *Tendrá un coche*.



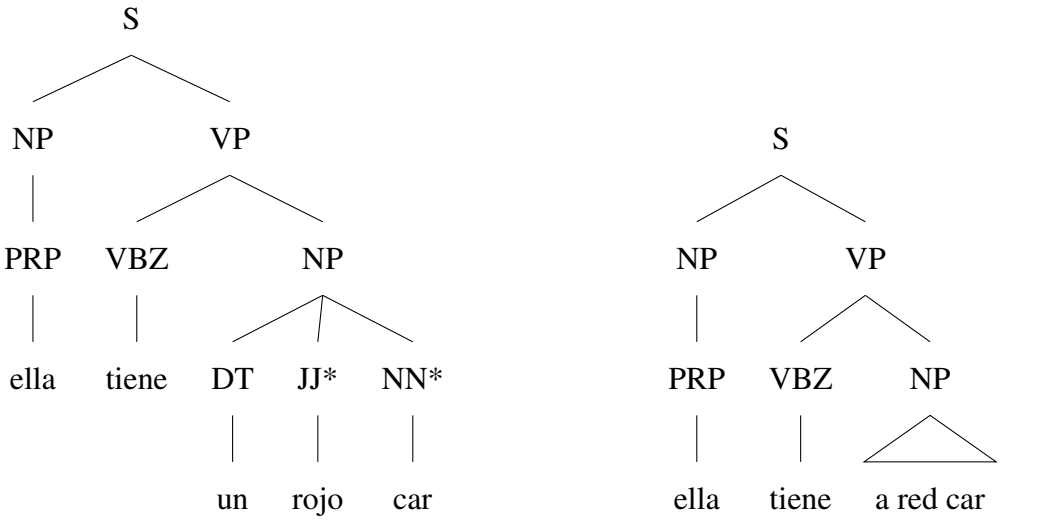
(b) English parse tree



(c) Spanish parse tree

Figure 5.3: English and Spanish parse trees with unequal lengths

proach ensures that the hierarchical structures of the parse trees in both languages are aligned. Subsequently, code-switched sentences can be generated by exhaustively using various combinations of constituents from the two languages, based on their tree structures.



(a) Incorrectly code-switching (b) EC theory allowed code-switching

Figure 5.4: Code-switched sentence generation with EC theory.

EC theory has been successfully applied to code-switched text data and proved crucial for a language modelling task (Pratapa et al., 2018), which our implementation inspired from. However, it should be noted that even when text data is taken from informal conversations on Twitter or other Internet platforms, they may still not follow the same patterns as speech (Sitaram et al., 2019).

We apply the Equivalence Constraint (EC) theory to generate code-switched sentences by combining syntactic segments permitted under the theory. To ensure the naturalness and linguistic plausibility of the generated outputs, we apply a two-stage process: (1) filtering using predefined linguistic constraints, and (2) ranking the remaining candidates using syntactic and contextual features. For each sentence pair, we retain at most the top 10 candidates based on this ranking.

Switch-points (SP): A switch-point is defined as a position in the sentence where the language of the word changes between adjacent tokens (Pratapa et al., 2018). We use SP Fraction (SPF), computed as the number of switch-points divided by the total number of word boundaries. This serves as a filtering threshold: candidates with SPF exceeding 0.1 are discarded.

CMI Following Gambäck and Das (2014), CMI is used to measure the degree of code-mixing in a sentence. For each utterance, we compute CMI as the total number of tokens not in the most frequent language, excluding named entities and punctuation. As with SPF, we use a filtering threshold of 0.3: sentences with CMI above this value are excluded.

POS-based ranking: After filtering, we rank the remaining candidates based on the syntactic context of the switched words. Since the code-switched sentence is derived from a parallel translation, and monolingual segments are retained, we can identify the POS tags of switched words in both source and target languages. We assign higher scores to sentences where proper nouns, nouns, determiners, or interjections precede the switch, and where the switched word itself is a noun or a subordinating conjunction. These preferences are informed by empirical findings in Soto et al. (2018) and our analysis in Chapter 4. To operationalize this preference, we define a simple scoring function based on POS tag occurrences at and around switch points. For each generated sentence, we identify all code-switched tokens and examine the POS tag of the word immediately before, at, and after each switch. We assign a score of +1 for each instance where the preceding POS tag is one of PROPN, NOUN, DET, or INTJ, and another +1 if the switched word itself is tagged as NOUN or SCONJ. The total POS-based score is then used as part of the overall ranking, where higher scores indicate syntactic contexts more likely to be associated with natural switching behavior.

Table 5.2: The statistics of the processed BM corpus, where the duration unit is hour.

	English		Spanish		CS	
	Number	Duration	Number	Duration	Number	Duration
Training	20813	10.9	7789	4.3	1879	1.6
Dev	3000	1.6	1250	0.6	250	0.2
Test	6000	3.1	2500	1.2	500	0.5

5.4 Experimental setup

5.4.1 Data

Although the BM corpus is publicly available, it lacks standardized preprocessing procedures. In our methodology, we initially classified all utterances into three distinct

categories: Spanish, English, and Code-switched. An utterance is considered code-switched if it contains exclusive words from both the English and Spanish lexicons. However, for words that appear in both lexicons, the categorization of the utterance depends on the linguistic context provided by the remainder of the sentence. After the cleaning process, we retained a total of 44,971 utterances. These were then divided into training, development, and test sets using a 7:1:2 ratio, respectively. To provide a clear overview of the dataset's composition, the statistics are presented in Table 5.2.

5.4.2 Training

5.4.2.1 Acoustic models

We utilized the Kaldi TDNN recipe[3] to develop our hybrid systems. The training data consisted of the 44,971 cleaned utterances described in Section 5.4.1, comprising Spanish, English, and code-switched speech. Initially, we extracted 40-dimensional MFCC features to train a GMM-HMM model. Prior to neural network training, we applied speed perturbation as a data augmentation technique. For the TDNN-HMM training, we used 40-dimensional high-resolution MFCCs and 100-dimensional i-vector features as inputs. The network architecture included seven TDNN hidden layers, each with 758 hidden units. We set the initial and final learning rates to 0.00015 and 0.000015 respectively, and trained the model for 4 epochs with a minibatch size of 128. The final phone set used for acoustic modeling included **44** distinct phones, constructed by merging the English and Spanish phoneme inventories as described in Section **??**.

5.4.2.2 Language models

We employed the SRILM toolkit to train n-gram models for comparative analysis. For each experiment, we trained a 3-gram model for decoding purposes and a 4-gram model for rescoring. The lexicon used was consistent across all experiments, utilizing the CMUDict for English and the Santiago Spanish Lexicon for Spanish. The lexicons comprise 206,500 English words and 91,121 Spanish words. Words not covered by these lexicons were treated as unknown (UNK).

[3] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5

Table 5.3: WER, in % and PPL on the test set. The top block shares the same language model which is trained only on the original transcript, and the bottom block shares the acoustic model with phoneme mapping.

	Test WER (%)				Test PPL		
	English	Spanish	CS	total	English	Spanish	CS
baseline	44.0	56.8	49.3	47.8	109.7	144.8	152.8
mapping base	43.6	56.4	49.1	47.5	109.7	144.8	152.8
translation	43.4	56.0	49.0	47.2	90.3	126.4	134.9
+ external	43.4	56.0	49.0	47.3	92.3	128.4	149.6
+ random	43.4	55.9	49.1	47.2	89.1	127.8	145.2
+ POS	43.3	55.6	48.7	47.0	88.7	126.0	130.1
+ EC	43.3	55.6	48.6	47.0	87.2	122.8	125.7
+ EC + POS	43.2	55.3	48.4	46.9	87.1	120.2	123.8

5.5 Results and discussion

Table 5.3 displays the word error rate (WER) and perplexity (PPL) for all experiments on the test set. Our *baseline* model utilizes the combined phoneme sets of English and Spanish, whereas the *mapping base* model maps the Spanish phoneme set to the English phoneme set. Despite using identical language models trained solely on the training set transcripts, phoneme mapping demonstrates a performance improvement, reducing the WER by 0.3% absolute. Consequently, we adopted phoneme mapping as the standard for our acoustic model. The variations in the experiments in the bottom block arise from the use of different synthetic texts for language modeling. The *translation* model indicates that the language model was trained not only on the transcripts but also on their translations. The + symbol indicates the techniques used for generating code-switched text added to the training corpus. To test the effectiveness of incorporating larger, but out-of-domain, text data, we interpolated the language model with models trained on external texts, specifically using TED talk subtitles for both English and Spanish (Tiedemann, 2012). This strategy did not yield performance improvements, demonstrating that merely increasing the volume of text data, if out-of-domain, is ineffective.

After acquiring word alignments from parallel sentences, we evaluated the efficacy of generating code-switched text by either randomly replacing aligned words or by ranking potential replacements based on the POS tags of current/adjacent words or EC

theory. Both the POS-based and EC-based approaches individually improved WER, and combining them yielded a small additional gain. However, the improvement was not fully additive, which may be due to overlap between the two methods: our EC implementation is based on constituency parses, which are themselves closely tied to POS information. As a result, the two strategies may reinforce similar syntactic patterns, limiting the marginal benefit when used together.

Our model with the best performance uses all of the linguistic information we discussed before, with approximately 2% improvement on both WER and PPL.

5.6 Conclusions

In this chapter, we have introduced a comprehensive framework tailored to the code-switched ASR task. This framework integrates phonological features through phoneme mapping, alongside leveraging POS and EC theory to generate more naturally code-switched text, and eventually achieves 2% improvement on PPL as well as WER. It is important to highlight that while our experiments utilized the BM corpus, the techniques employed are not restricted to specific languages when the related NLP tools (a constituency parser for at least one of the languages involved) are present. The absence of language-specific constraints in our framework not only demonstrates its robustness but also underscores its potential applicability in a broader multilingual context.

Chapter 6

Machine learning based Text generation

The preceding chapter outlined the challenges of training ASR systems amid dataset scarcity and introduced an approach for generating code-switched text based on linguistic theory. Building on this foundation, this chapter introduces a novel method whereby a multilingual Machine Translation system, originally trained on paired monolingual data, is adapted to generate code-switched text. By leveraging shared linguistic representations of Mandarin and English, this method demonstrates superior code-switched text generation capabilities, especially for syntactically diverse language pairs, compared to previous techniques. This chapter is adapted from content originally presented in our paper published in the proceedings of InterSpeech 2023 (Chi et al., 2023).

6.1 Introduction

Building on insights from the previous chapter, which highlighted the complexities of intra-sentential code-switching in ASR systems, this chapter introduces a novel data-driven approach. Despite challenges such as linguistic variability and data scarcity, we leverage a multilingual Machine Translation (MT) system trained on non-code-switched data to generate code-switched text. Our new method is distinct because it uses shared linguistic representations of Mandarin and English to mimic how bilingual people naturally code-switch. Rather than relying on rule-based methods that heavily analyze code-switched corpus data, our technique takes advantage of the inherent syntactic and semantic compatibilities between the languages from parallel data. We

pretrain a Transformer-based encoder-decoder model on parallel texts and then manipulate the decoder to switch languages a specified number of times during translation. This approach utilizes the shared representations induced by pretraining a multilingual translation model, effectively bridging linguistic gaps without direct code-switched input. Our experiments focus on intra-sentential code-switching and show improvements over two standard methods when tested on the SEAME corpus (Lyu et al., 2010).

6.2 Related work

There have been significant efforts to construct synthetic data to augment the relatively small existing datasets of code-switched text. Given the abundance of parallel monolingual texts (or can be created by MT), a popular research direction has involved aligning these sentences and mixing them under the guidance of linguistic theories of code-switching, as discussed in Chapter 5. These rule-based methods typically involve extracting and concatenating monolingual fragments from parallel texts. In contrast, another line of work has focused on directly generating synthetic code-switched sentences using language models or conditional language models that have been pretrained on a limited corpus of code-switched data. This approach seeks to create more naturally integrated code-switched sentences by learning from the nuances of existing code-switched interactions. The following subsections will review several recent frameworks that have been developed along these lines.

Recurrent Neural Networks (RNNs) are specifically designed to process sequential data. By maintaining a hidden state that captures information from previously encountered tokens, RNNs effectively model temporal dependencies. This capability enables them to handle variable-length input and to predict future elements based on historical context. A straightforward application of RNNs in language modeling involves training on real code-switched text to generate additional synthetic data. For instance, Yılmaz et al. (2018) trained an LSTM-based language model on transcriptions from the FAME! Corpus and used it to generate synthetic code-switched text, which was then incorporated into the language model training data to improve ASR performance. Although the LSTM architecture itself is standard, the work demonstrates the practical utility of recurrent models for code-switched text generation in low-resource settings. Adel et al. (2013a) enhanced the standard RNN model by integrating POS tagging into the input layer and adapting the output layer to factorize based on language, calculating the probability of each word contingent on language selection. This model was further

refined by incorporating a factor language model that treats each word as a sequence of features, employing linear interpolation with n-gram probabilities to enhance prediction accuracy (Adel et al., 2013b). LSTMs have seen additional improvements for code-switched language processing through multitask learning frameworks that simultaneously engage in POS tagging (Winata et al., 2018) and language identification classification (Chandu et al., 2018). Further innovating in this space, Garg et al. (2018) introduced a Dual LSTM architecture. In this model, each LSTM cell is dedicated to processing input tokens from one of two languages, producing unnormalized output distributions for each. These outputs are then combined and normalized through a softmax function to generate a cohesive output distribution across the entire bilingual vocabulary.

Generative Adversarial Networks (GANs) are used in unsupervised machine learning, implemented by a system of two neural networks contesting with each other in a zero-sum game framework (Goodfellow et al., 2014). This setup involves a generator that creates data instances and a discriminator that evaluates them, with the generator striving to produce data indistinguishable from reality while the discriminator aims to detect differences between real and generated data. The adversarial process drives both networks to improve their methods continuously, enhancing the quality of the generated data over time. Recent research has extended GANs to the code-switching task, exploring their potential to generate realistic bilingual text that switches languages seamlessly. (Chang et al., 2019) used the generator to transform a monolingual sentence into a code-switched sentence and the discriminator to predict if the sentence is real or generated. With a similar idea, Gao et al. (2019) employed BERT models in the generator, which first was finetuned on code-switched text to learn to generate code-switched words based on the contexts.

Variational AutoEncoders (VAEs) encode data into a compressed latent space and decode it back to the original space, effectively capturing complex data distributions (Kingma and Welling, 2022). Samanta et al. (2019) trained VAE models on parallel monolingual text before fine-tuning on code-switched text to facilitate bilingual text generation. With the advent of large language models, there has also been very recent work using prompting to generate code-switched sentences (Yong et al., 2023; Terblanche et al., 2024).

6.3 Methodology

6.3.1 Parallel Text Pretraining

Word embeddings play an important role in modern Speech and NLP models due to their ability to capture semantic relationships between words in a dense, continuous vector space. These embeddings represent words or phrases as vectors in a high-dimensional space, where semantically similar words are usually positioned closer together. A significant area of interest within the field is understanding how the embedding spaces of different languages relate to one another. Specifically, researchers have been investigating whether the embedding spaces of different languages exhibit a similar structural organization, and if so, to what extent these spaces can be aligned or mapped onto each other. Early research in cross-lingual word embeddings demonstrated that non-contextual word embeddings—where each word is represented by a single fixed vector regardless of context—can be aligned across languages using linear transformations. For instance, Mikolov et al. (2013)) introduced methods to align word embeddings of different languages by finding linear mappings that transform embeddings from one language into the embedding space of another language. This approach showed that even without context, embeddings could be effectively aligned. With the advent of contextual word embeddings, the approach to alignment has evolved. Research showed that even for separate monolingual BERT models, alignment of contextual embeddings across languages could be achieved using more sophisticated techniques, such as Centered Kernel Alignment (CKA) (Conneau et al., 2020b). Moreover, multilingual BERT models, which are trained on multiple languages simultaneously, have demonstrated promising results in aligning contextual embeddings across languages (Muller et al., 2021).

Would emergent alignments on word embeddings (contextual or not), learned simply from text prediction tasks on multiple languages, be enough to support code-switched generation among those languages? We study this question by pretraining a many-to-many MT system on monolingual inputs and outputs, and then forcing the decoder to translate monolingual inputs into code-switched outputs. Specifically, we train a Transformer encoder-decoder model (Vaswani et al., 2017) to translate between any pair of languages in $\{$Mandarin, English$\}$. We then translate monolingual sentences from either Mandarin or English to sentences that are forced to code-switch to varying degrees, using a constrained decoding technique known as grid beam search (Hokamp and Liu, 2017), which we describe in detail below. Finally, we evaluate

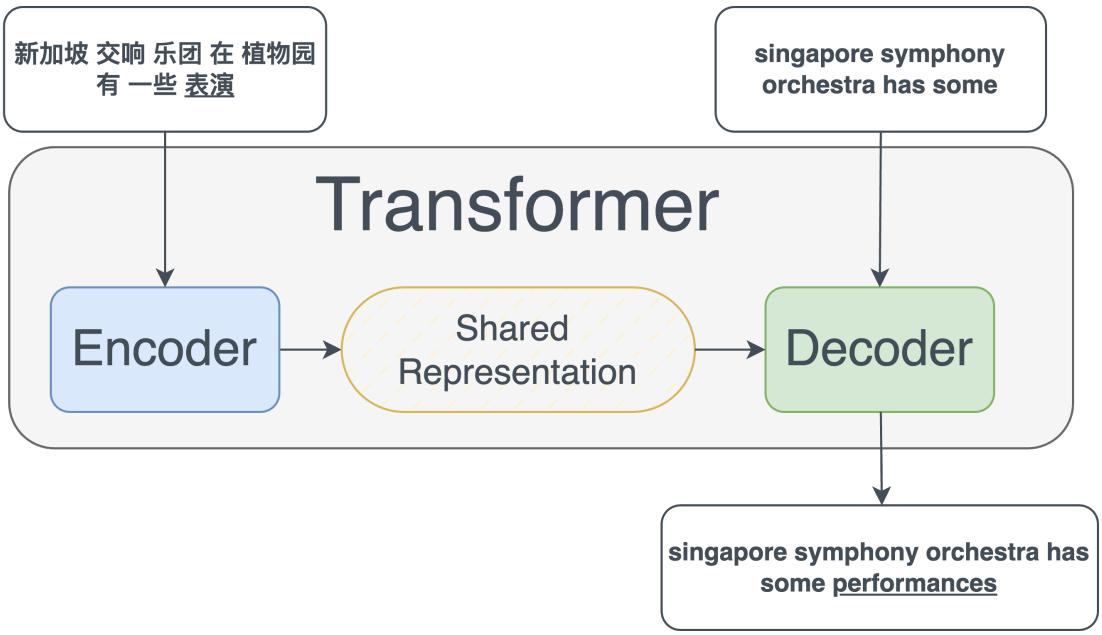


Figure 6.1: Transformer architecture, where for each parallel sentence pair (S_{zh}, S_{en}), we have four training examples $(S_{zh}, S_{zh}), (S_{en}, S_{en}), (S_{zh}, S_{en}), (S_{en}, S_{zh})$. Here we use (S_{zh}, S_{en}) for illustration

the utility of our synthetic code-switching sentences by using them to train a n-gram language model to use in a downstream ASR system similar to Chapter 5.

6.3.2 Translation Model

We use a Transformer encoder-decoder architecture (Vaswani et al., 2017) (Figure 6.1), with a vocabulary that is the disjoint union of the vocabularies of the two languages of interest. For simplicity, strings that exist in both vocabularies, such as numbers, are given two separate embeddings, one for each language. This also facilitates the softmax modification described later in this section. These two vocabularies are harvested from the two sides of the parallel training corpus. They are constructed using task-specific tokenization: for Mandarin, we apply JieBa[1] tokenization during training of the translation model, while English is tokenized based on whitespace and punctuation. Note that this results in separate subword or word units for each language, and the vocabularies remain disjoint by design. We create 4 training examples for every pair of parallel sentences (x, y): each training example takes one of x or y as encoder input and one of x or y as decoder output. This is the same scheme used to train unified

[1] https://github.com/fxsjy/jieba.git

Algorithm 1 Pseudo-code for Grid Beam Search, note that t and c indices are 0-based

procedure CONSTRAINEDSEARCH($model$, $input$, $constraints$, $maxLen$, $numC$, k)
  $\mathbf{startHyp} \Leftarrow$ model.getStartHyp($input, constraints$)
  $\mathbf{Grid} \Leftarrow$ initGrid($maxLen$, $numC$, k) ▷ initialize beams in grid
  $\mathbf{Grid}[0][0] = \mathbf{startHyp}$
  for $t = 1,\ \ t{+}{+},\ \ t < maxLen$ **do**
    for $c = max(0, (numC + t) - maxLen),\ \ c{+}{+},\ \ c \leq min(t, numC)$ **do**
      $n, s, g = \varnothing$
      for each $hyp \in \mathbf{Grid}[t-1][c]$ **do**
        if $hyp.isOpen()$ **then**
          $g \Leftarrow g \bigcup$ model.generate($hyp, input, constraints$) ▷ generate new open hyps
        end if
      end for
      if $c > 0$ **then**
        for each $hyp \in \mathbf{Grid}[t-1][c-1]$ **do**
          if $hyp.isOpen()$ **then**
            $n \Leftarrow n \bigcup$ model.start($hyp, input, constraints$) ▷ start new constrained hyps
          else
            $s \Leftarrow s \bigcup$ model.continue($hyp, input, constraints$) ▷ continue unfinished
          end if
        end for
      end if
      $\mathbf{Grid}[t][c] =$ k-argmax$_{h \in n \bigcup s \bigcup g}$ model.score(h) ▷ k-best scoring hypotheses stay on the beam
    end for
  end for
  $topLevelHyps \Leftarrow \mathbf{Grid}[:][numC]$ ▷ get hyps in top-level beams
  $finishedHyps \Leftarrow hasEOS(topLevelHyps)$ ▷ finished hyps have generated the EOS token
  $bestHyp \Leftarrow$ argmax$_{h \in finishedHyps}$ model.score(h)
  return $bestHyp$
end procedure

Figure 6.2: Pseudo-code for Grid Beam Search, adapted from (Hokamp and Liu, 2017)

MT systems that translate between many language pairs using the same parameters (Johnson et al., 2017), and indeed our method could be extended beyond 2 languages.

In a preliminary study, we found that it worked best to train the model with a modified version of softmax that normalizes over only words of the desired output language, as opposed to normalizing over the entire union vocabulary. The latter formulation penalizes assigning high logits to any word in the other language. That hinders the natural emergence of aligned word embedding spaces between the two languages, since it pushes away uniformly the embeddings of the other language.

6.3.3 Grid Beam Search

Following the method outlined by (Hokamp and Liu, 2017), we constrain the decoding process by the number of code-switching points. Unlike a standard beam search, this

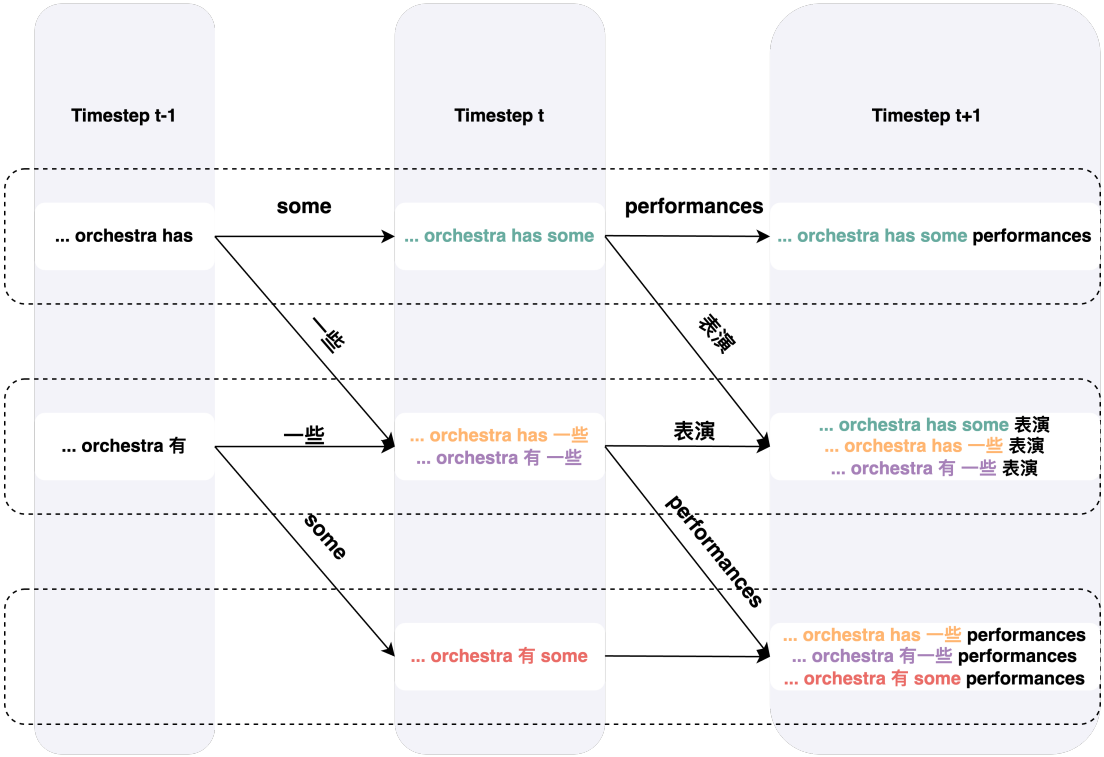


Figure 6.3: Decoding process with grid beam search, where shaded boxes denote three subsets with 0, 1 and 2 code-switching points. Each box represents the top k hypotheses (not all hypotheses are shown in the figure) at each timestep in each subset. Colored text and edges show the expansion for each beam.

approach partitions the set of prefix sequences in the beam into subsets based on a specific feature, as illustrated in Figure 6.2, which in our case is the number of code-switching points. Each subset is independently pruned to retain only the top-k prefixes. When a prefix is extended during the decoding process, this extension may introduce a new switching point, altering the feature value of the prefix and causing it to shift to a different subset. Ultimately, the algorithm concludes by selecting the top elements from each subset, yielding the best outputs categorized by 0, 1, 2, and more code-switching points. This method is visually exemplified in a simplified manner in Figure 6.3

6.3.4 Other Approaches

To compare to prior methods for generating code-switching text (Pratapa et al., 2018; Winata et al., 2019), we also implement models based on EC Theory and on Pointer-Generator Networks (PGN). Both of these models depend on parallel data. The PGN

additionally requires code-switching training data, so it serves as a supervised baseline against which to compare our unsupervised method.

Equivalence Constraint Theory claims that code-switching can only happen at boundaries where both languages have the same surface structure. Following the pipeline in Chapter 5, we first use *fast_align* to obtain the word alignment between parallel sentences and then generate the parse tree for English text with the Berkeley neural parser (Kitaev and Klein, 2018). In contrast to the EC baseline in (Winata et al., 2019), where they used a simplified linear version of EC that determines the acceptability of a substitution solely by checking whether there are crossing alignments, here we follow the previous chapter and use the alignment together with the constituency parses to determine if a substitution is acceptable. We first produce parses for the English part of the parallel texts, then individually replace them with parses for Mandarin based on the word alignments. Then we will re-order the child nodes in the subtrees so that the order of the words is the same as in Mandarin sentence. If one word is aligned to multiple words from the other language, then we can treat the multiple words as a single multi-word node. By ensuring both languages share the same hierarchical structure, we effectively generate code-switched sentences that combine constituents from both languages.

Pointer-Generator Networks integrate the conventional sequence-to-sequence model with an attention mechanism, and require supervised training. The distinctive feature of these networks is the pointer mechanism, which at each step in the output sequence, dynamically decides whether to generate a word from the model's vocabulary or copy directly from the input sequence. This dual functionality allows the model to flexibly handle both novel and repetitive information in text generation tasks. This decision is controlled by a contextually dependent switch variable p_{gen}, which is calculated at each decoder step. When generating the output vocabulary distribution for each timestep, the model integrates the probability distribution from the language model and the attention distribution as

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i \tag{6.1}$$

where $P_{\text{vocab}}(w)$ is the probability of the word w from the model's vocabulary, and a_i are the attention weights for each input word w_i that matches the word w. The input is the *concatenation* of the parallel sentences x and y and the output is the desired code-switched sentence, as illustrated in Figure 6.4. We re-implement the model introduced

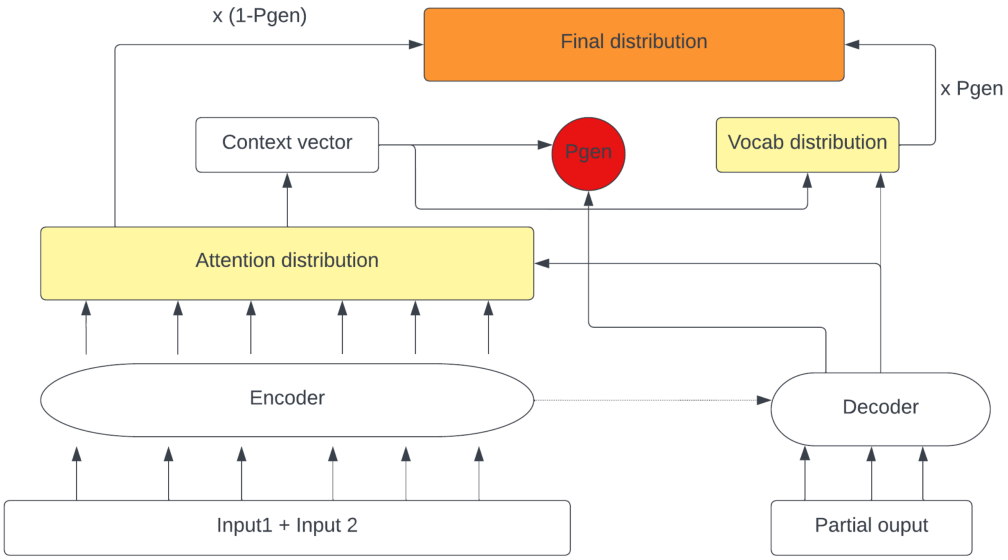


Figure 6.4: The framework of pointer-generator networks.

by (Winata et al., 2019) with a modification: we omitt POS tags as additional input features to ensure a fair comparison with our methods and to reduce dependency on NLP tools, which are often unavailable for many languages.

6.4 Experimental setup

6.4.1 ASR Framework

We initiate our process by pretraining an acoustic model on a combination of TED-LIUM 3 (Hernandez et al., 2018), an extensive English speech corpus derived from TED Talks, and AISHELL-1 (Bu et al., 2017), a comprehensive Mandarin speech corpus with over 170 hours of labeled data. This combined training serves as our pretrained acoustic model, capturing a rich variety of phonetic and prosodic features across both languages. Subsequently, we fine-tune this model on the *monolingual* utterances from SEAME, a bilingual Mandarin-English speech corpus. This step adapts the model to the specific characteristics of the SEAME dataset, thereby enhancing its ability to handle code-switched speech more effectively. As our acoustic model is not trained on any of the code-switching utterances, it is typical of those used in existing multilingual ASR systems. Below, we combine this hybrid acoustic model with

language models trained on code-switching text, experimenting with different ways of obtaining such text. We evaluate the resulting ASR systems on the held-out portion of SEAME (11,852 utterances),[2] which contains both monolingual (5,384) and code-switched (6,468) spoken utterances.

6.4.2 Real code-switching Text

SEAME contains around 50K spoken utterances for training that are labeled as code-switching. We use their transcripts to train a LM on real data. Our goal in the next section is to synthesize ersatz data that works almost as well.

6.4.3 Parallel Non-CS Text

For each code-switching utterance z from SEAME, we also ask Google Translate to translate it to both English (x) and Mandarin (y), in each case treating the source sentence z as if it were in the other language.[3] This yields 50K parallel utterances (x, y, z).

Methods	**Training Pairs (input, output)**	**Generation**
CT	(x,y), (y,x), (x,x), (y,y)	Setting 1, 2, 3 switching points: 1 sentence each on x and y, totaling 6 sentences.
ECT	/	All legal sentences permitted by the theory, around 12 per pair
CST	(x,y), (y,x), (x,x), (y,y), (x,z), (y,z)	Two sentences each on x and y, totaling 4 sentences
PGN	(x+y, z)	Freely 3 sentences

Table 6.1: Overview of different methods, their training data, and generation outcomes. Here, x and y denote the parallel monolingual sentences in Mandarin and English respectively, and z denotes a code-switched sentence derived from the pair (x, y). For bidirectional MT models, x may be Mandarin and y English, or vice versa, depending on the direction of the pair.

[2]Both the `dev_man` and `dev_sge` subsets are from `https://github.com/zengzp0912/SEAME-dev-set.git`.

[3]Google Translate may not be optimized to deal with code-switched inputs like z. As a result, its supposedly monolingual translations sometimes contain some code-switched content from z.

6.4.4 Synthetic code-switching Data

For a controlled comparison, we take care to have all of our methods generate synthetic datasets of the same size. For our proposed *unsupervised* Constrained Translation (**CT**) approach, we train a unified MT model (Section 6.3.2) on the 50K (x, y) pairs and then use grid beam search (Section 6.3.3) to decode 3 code-switching translations of each x and each y. Specifically, for each $c \in 1, 2, 3$, the final beam holds up to 5 prefixes with exactly c switching points, and we return the top 1 of those. That yields 6 sentences per pair, which we then randomly subsample to 3.

As our unsupervised baseline, we run **ECT** (Section 6.3.4) on the (x, y) pairs. Hybridizing each pair in all legal ways yields about 12 code-switching utterances on average, which we then subsample to 3.

To make use of SEAME *supervised* z data, we start with the unified MT model above, and fine-tune it to translate each of x and y to each of x, y, and z. That is, each SEAME utterance now yields 6 training examples instead of 4. This can be seen as multi-task regularization. Our actual goal is to learn to translate $x \mapsto z$ and $y \mapsto z$, but fine-tuning on those pairs alone would lead to low diversity in the beam search, which generates duplicate n-grams and prevents us from training a KN-discounted n-gram LM. Thus, we also include the other 4 pairs when fine-tuning. We refer to this model as **CST** (code-switched translation) and use it to retranslate each x and each y to 2 new code-switching utterances (totaling 4), which we then subsample to 3.

As our supervised baseline, we train **PGN** (Section 6.3.4) to translate from the concatenated input xy to the code-switched z, and use it to retranslate each xy pair to 3 new code-switching utterances.

Table 6.1 provides an overview of the training and generation processes described above. All generated sentences longer than three are subsampled to three to ensure uniform text volume for language modeling. Note that we used only the code-switching portion of SEAME, not the monolingual portion, to generate our synthetic code-switching utterances. The main reason is that we wanted to control as much as possible when comparing the performances of different data augmentation methods with the Real code-switching data. Utterance length has been shown to have a significant influence on the probability of code-switching (Calvillo et al., 2020). The average length of monolingual and code-switched sentences differ significantly in SEAME, which may produce qualitatively different kinds of synthetic code-switching text compared to those generated from the translations of the code-switching portion. This ensured a

(unrealistically good) match with the topics and lengths of the held-out code-switching utterances.

6.4.5 Model Architectures and Training

6.4.5.1 Translation Models

We use 8-layer, 12-head Transformer encoder and decoders with dimension size 768 for our **CT** and **CST** systems. For the **PGN** baseline, we implemented our own following (Winata et al., 2019) using a one-layer Bi-LSTM encoder and a one-layer LSTM decoder with hidden dimension 256.[4]

We tokenize all Mandarin parts of the data using JieBa[5] for pretraining our translation models, while we use character tokenization for our implementation of pointer-generator networks and the ASR language models. For English parts, we always tokenize at whitespace and punctuation.

6.4.5.2 Language Models

For each dataset described in Section 6.4.2, 6.4.3, and 6.4.4, we use the SRILM toolkit to train a trigram model with Kneser-Ney smoothing (Ney et al., 1994). For each *generated* dataset in Section 6.4.3, and 6.4.4, we additionally train a trigram model by *combining* it with real code-switched data (Section 6.4.2) via LM interpolation. This leads to improvements in WER and PPL discussed in Section 6.5. Interpolation weights are optimized on the monolingual SEAME data, disjoint from both SEAME code-switched training and test set.

6.4.5.3 Acoustic Model

Considering that both Mandarin and English are resource-rich languages, unlike the previous focus on Spanish-English code-switching, we include a pretraining step in this chapter, leveraging the abundant monolingual datasets available for each language. We use the Kaldi toolkit to train a hybrid acoustic model. AISHELL and TED-LIUM datasets are combined to train a standard speaker adaptive GMM-HMM model at first, then we use it to produce alignments to train a CNN-TDNN model with lattice-free maximum mutual information (LF-MMI) criterion, which consists of 6 CNN layers

[4]We also explored adding more parameters in a preliminary study but did not observe significant improvements in downstream ASR.

[5]`https://github.com/fxsjy/jieba.git`

and 12 TDNN layers. We pretrain the acoustic model on AISHELL and TED-LIUM and then fine-tune it on SEAME monolingual. Although we have never used code-switching speech during acoustic training, by pretrainging on additional two monolingual speech corpora, the obtained acoustic model has already achieved a competitive result compared with models trained on entrain SEAME corpus in (Winata et al., 2018; Zeng et al., 2019).

The lexicon is obtained by combining the pronunciations of English words from CMU dictionary and Mandarin characters from AISHELL dictionary. We use different phoneme units for each language. Pronunciations for OOV words in the training data are generated by Phonetisaurus (Novak et al., 2016). In the end, we have 180K entries in the lexicon and any uncovered words are treated as UNK.[6] Compared with training two monolingual models using the same architecture, the performance of the obtained bilingual ASR model only drops by 0.5 absolute WER on the same SEAME dev set.

6.5 Results and Discussion

6.5.1 ASR Results

Table 6.2 presents WER of our ASR system on SEAME test set. The ASR system uses the pretrained acoustic model with trigram LMs trained on synthetic code-switching text. We break down the test set further by whether there is any code-switching contained. The unsupervised (**CT**) and supervised (**CST**) versions of our proposed model respectively achieves better overall WER than the **ECT** and **PGN** baselines, regardless of LM interpolation with **RealCS**. Among unsupervised methods, **CT** consistently gets lower WER than **ECT** on both code-switching and monolingual utterances. Among supervised methods, **CST**'s superior performance compared to **PGN** on the monolingual subset and worse performance on the code-switching subset could be attributed to the multi-task regularization as well as its lack of the dual-language input and a copying mechanism which can make learning the alignment between the two languages easier.

[6]There are 117 OOV out of 151146 total word tokens on test data.

Table 6.2: ASR WER and LM perplexity evaluations on SEAME dev sets. In each row, the overall best system is bolded and best systems within categories are underlined. When combining with RealCS, an optimal weight is selected following Section 6.4.5.2

			Supervised		Unsupervised		
		RealCS (50K)	code-switchingT (150K)	PGN(150K)	CT(150K)	ECT(150K)	Non code-switching(100K)
Individual Datasets	WER_{all}	**31.32**	31.44	31.80	33.52	34.68	33.97
	WER_{cs}	**29.82**	30.58	30.50	32.80	33.73	33.44
	WER_{mono}	34.94	**33.78**	35.11	35.53	37.03	35.45
	PPL	**123.43**	128.86	136.60	161.21	183.77	153.57
Combined with RealCS	WER_{all}	/	**30.80**	30.90	30.92	31.06	31.18
	WER_{cs}	/	29.73	**29.48**	29.62	29.69	30.09
	WER_{mono}	/	**33.63**	34.38	34.21	34.46	34.07
	PPL	/	119.42	134.08	133.15	130.70	129.87

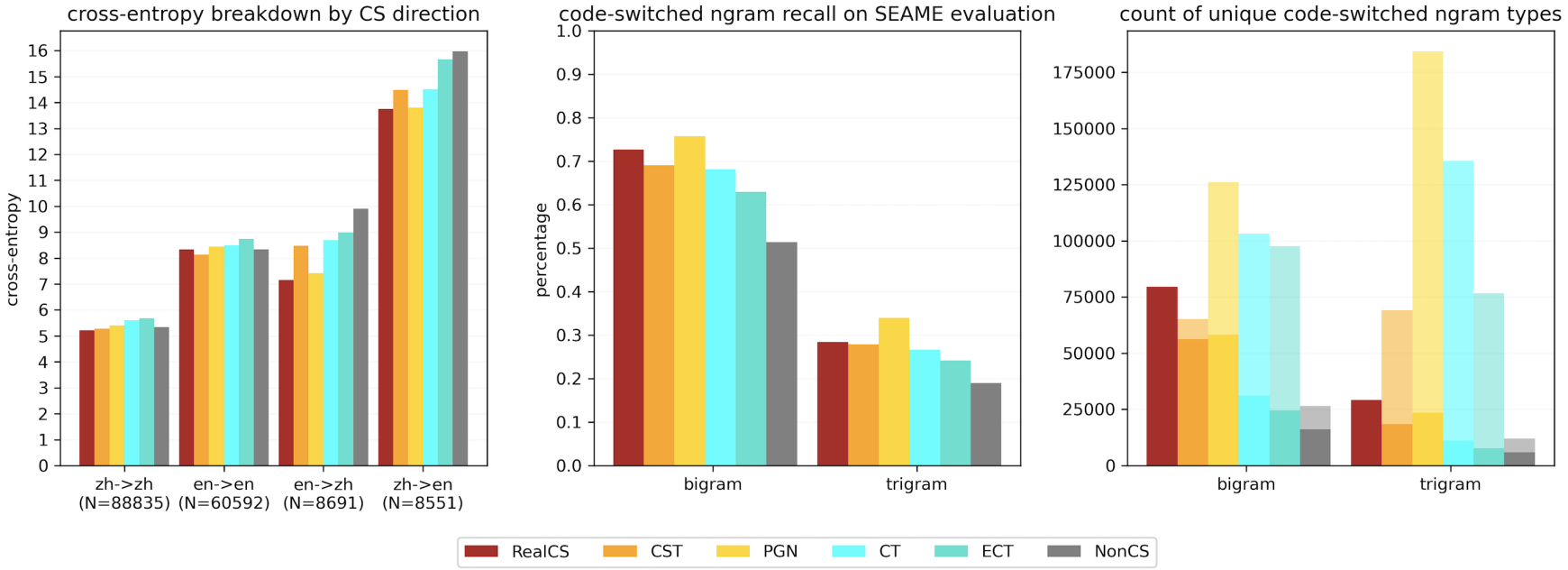


Figure 6.5: Cross-entropy breakdown and n-gram coverage. Left: Cross-entropy breakdown by the transition of language IDs. Middle: *Token*-level code-switched bigram and trigram recall on the SEAME evaluation set. Right: *Type*-level code-switched bigram and trigram counts. Darker bars count the number of shared n-gram types between a particular dataset and the SEAME training data.

6.5.2 Language Modeling Results

We also evaluate the various LMs on the text transcripts of SEAME test set, ignoring the audio. The PPL results are in Table 6.2, with a cross-entropy breakdown in Figure 6.5 (Left). As the plot shows, both **ZH** and **EN** tokens cause higher surprisal when the previous token is in the other language, but models that use more real code-switching data are less surprised.

Figure 6.5 (Mid) shows the percentage of the code-switching bi/trigrams contained in the test set that appear in the synthetic texts. It should be noted that a truly non code-switching corpus would contain no code-switching bigrams and trigrams. However they exist in our non code-switching dataset because our non code-switching dataset is generated by (Google) translating code-switched sentences into monolingual ones, and Google Translate sometimes fails to produce a purely monolingual mandarin output, especially for interjection words such as *lor*, *ah* and *er*. Figure 6.5 (Right) shows the total counts of unique code-switching bi/trigrams in the synthetic text (lighter bars) as well as the number of those bi/trigrams that also appear in the real code-switching training data (darker bars).[7] Unsurprisingly, **RealCS** and supervised methods generate data with much better *n*-gram coverage. Under unsupervised training, **CT** can generate more diverse language transitions compared with **ECT**, as **ECT** is constrained to output segments that are in the original sentence pair. Under supervised settings, **PGN** generates more diverse *n*-grams and has better coverage of the test set.

6.5.3 Qualitative Properties of Synthetic code-switching Text

The code-switched sentences generated by our **CT** model are not always perfect translations of the input, but are they reasonable code-switching text? Most of the code-switching consists of lexical substitutions as follows.

(1) you go to take 营 销 loh you are the 最 好 的
you go to take **marketing** loh you are the **best**

We find that the resulting sentences are mostly understandable, but errors occur such as follows, where the *train* is treated as a noun instead of a verb in the generation.

(2) it is fun to 火 车 with them

[7] Recall that the training data of the various synthetic generations methods were derived from SEAME code-switching training set.

it is fun to **train** with them

Sometimes, they might not code-switch at the typical locations a bilingual speaker would (which can be subjective), and some sentences might code-switch *too often* because the **CT** approach requires them to do so.

6.5.4 Discussion

We only experimented with two languages in this work, but the framework could be generalized to generate text that code-switches among any number of languages. Although only hybrid ASR systems have been used in this paper, which generally perform better when limited data is available (Zeng et al., 2019), it is interesting to investigate if the same finding still holds in an End-to-End framework.

Compared with **ECT**, whose performance is constrained by the effectiveness of the tools for natural language processing, **CT** is fully data-driven, relying on learning shared representations of the language pair from translation pretraining. Hence, it is less sensitive to the formality of the data. SEAME contains mostly conversational speech and transcriptions, which is harder to parse and align, but on more formal domains where linguistic knowledge may help, **ECT** may become more competitive since it has a better inductive bias. **CT** is able to freely generate code-switching text, including content (such as paraphrases) not contained in either of the parallel sentences, which could improve diversity. On the other hand, the popular **ECT** approach is constrained not to do this, which prevents it from generating wildly incorrect outputs.

6.6 Conclusions

We presented a simple yet effective idea: to leverage the emergence of shared representations in pretrained encoder-decoder models to generate synthetic code-switched data without using any prior knowledge about code-switching. Although, as we would expect, the data it generates does not outperform methods that use real code-switching as supervision, it performs slightly better than other unsupervised methods such as ECT, and without needing a parser or specialized knowledge about code-switching. Showing the possibility of a fully data-driven, learning approach to unsupervised code-switching generation opens up opportunities for more research in the design of the model architectures and training objectives. While we explored a simple instantiation with

Transformer encoder-decoders and just the translation objective, more specialized architectures could lead to better representation sharing and in turn better code-switching generation.

Chapter 7

Conclusion

7.1 Summary

This thesis contributes to both the understanding and modeling of intra-sentential code-switching phenomena through both theoretical exploration and practical applications. Chapter 3 introduced a new framework for evaluating the effectiveness of code-switching metrics across various domains and languages by assessing how well their outputs align with linguistic findings. The framework is able to identify that code-switching can exhibit different styles depending on the speech setting. We proposed a normalized version of existing popular metrics to account for orthographic token differences. We also introduced a new metric that, instead of relying on language tags, uses machine translation systems to measure the lexical properties relative to the participating language pairs. This framework aims to enable future research to benchmark different metrics more effectively, while the new metric offers a novel approach to measuring code-switching with respect to the languages involved and will allow researchers can capture a broader range of code-switching properties.

Building on previous research on exploring the relationship between POS and code-switching occurrence, Chapter 4 further incorporated word positions into the analysis, providing robust evidence that this relationship is most pronounced in close proximity to switching points. By studying language pairs with different syntactic structures, we also demonstrate that more syntactically diverse structures correlate with reduced flexibility in code-switching.

Furthermore, Chapters 5 and 6 explored two effective methodologies for generating code-switched text. Chapter 5 utilizes ET theory to constrain segment replacement, while Chapter 6 employs machine translation systems with parallel data. Both

approaches have shown significant promise in downstream applications, such as language modeling and ASR tasks. These methods not only enhance the generation of realistic code-switched data but also contribute to the improvement of computational linguistic models and their practical applications.

Taken as a whole, the experiments in this thesis collectively highlight the complexity and challenges of studying code-switching, from foundational research on how to quantitatively analyze this behavior across different languages, to practical efforts aimed at improving multilingual systems in code-switching contexts. We aim to provide valuable insights into the nature of code-switching and offer practical solutions for technological applications. However, like all studies in this field, the contributions are constrained by the limited number of language pairs and the amount of data analyzed, a limitation that will be discussed further in the next section.

7.2 Limitations

The most significant limitation arises from data scarcity. Although code-switching is a widespread phenomenon that is increasing due to globalization, capturing and archiving it remains challenging because it tends to occur more frequently in informal contexts. Furthermore, most existing corpora are relatively small, typically comprising only tens of hours of data, compared to hundreds or even thousands of hours available for monolingual speech. Within these small datasets, actual instances of code-switching often account for less than 50% of the content. The challenge is further compounded by the fact that one of the languages involved in code-switching is often a low-resource language, making data collection even more difficult. As a result, theoretical studies are typically constrained to specific language pairs and datasets, which limits the scope of their findings. This limitation also applies to the experiments in this thesis—despite efforts to explore different languages, the generalizability of the proposed claims remains uncertain.

Another limitation relates to the sociolinguistic factors influencing code-switching. In Chapter 2, we reviewed work that establishes that code-switching occurs in different contexts and for various reasons. The experiments in Chapter 3 used different *speech settings*, in this case technical lectures versus conversational speech - as ground truth for analysis. However, the concept of *conversational speech* remains lacking a precise definition. Classifying different settings in a continuous manner is challenging due to the multitude of social factors involved. This ambiguity poses a potential problem

when researchers attempt to apply the proposed framework to other language pairs, as it may restrict the analysis to extreme datasets, such as those that are either highly informal or highly formal. This limitation also affects any knowledge adaptation effort, where the extent of domain mismatch is uncertain and difficult to predict. Similarly, the investigation into the relationship between POS and code-switching assumes a relatively static view of language usage, where in fact it can be influenced by rapidly changing social and cognitive factors.

Our research also has a heavy reliance on existing computational tools. For instance, LID systems are employed at the early stages of corpus collection, while we use machine translation systems in Chapters 3, 5, and 6. Additionally, a POS tagger is utilized in Chapter 4, and a neural constituency parser in Chapter 5. The effectiveness of each of these tools is dependent on the quality and quantity of data they were trained on, which ties back to the issue of data scarcity. Most NLP models to date have been developed using (multiple) monolingual data, without specifically targeting code-switching scenarios. As a result, errors introduced at each stage can propagate through the analysis and are difficult, if not impossible, to correct. This creates a chain effect, where the final results are inevitably influenced by the performance limitations of the tools used. Although there are ongoing efforts to develop benchmarks for code-switched NLP tasks, there is not yet a SOTA model that can perform equally well for all languages. This inconsistency makes it challenging to compare findings across different experiments, as the availability and performance of these tools vary significantly.

Finally, this thesis lacks human experiments, which is particularly relevant for Chapters 5 and 6, where two approaches for code-switched text generation are proposed. While we have conducted quantitative analyses using synthetic text for downstream tasks and performed limited inspections to identify lexical replacements, these methods do not fully capture the naturalness or complexity of actual code-switching. The extent to which our approaches accurately reflect real-world code-switching remains uncertain, as we have not included empirical evaluations involving human subjects to assess the effectiveness and authenticity of the generated text.

7.3 Future research directions

The most immediate direction for future work would involve applying newer SOTA model architectures to the same set of experiments conducted in this thesis. In the ASR

downstream tasks, the current hybrid systems could be replaced with more advanced end-to-end models such as wav2vec 2.0 (Baevski et al., 2020) or Whisper[1], both of which have demonstrated strong performance in multilingual and code-switched speech recognition. These models can also be combined with external n-gram language models or fine-tuned for improved integration with syntactic constraints. For code-switched text generation, multilingual pretrained language models such as mBART (**?**) or instruction-tuned variants of T5 or mT5 could be used to improve fluency and syntactic well-formedness. These models offer a stronger baseline for evaluating structural constraints, such as those based on part-of-speech tags or syntactic projections. While we expect these stronger models to improve quantitative performance—e.g., lower WER or perplexity—the core conclusions of the thesis regarding structural constraints, syntactic context, and the evaluation of code-switching metrics are likely to remain valid. Finally, expanding the scope to include more language pairs, particularly involving low-resource languages, would further test the generalizability of the proposed frameworks and deepen our understanding of cross-linguistic code-switching behavior.

Given the findings and limitations identified earlier, one research direction would be to collect code-switching corpora that are specifically grounded in linguistic research. As demonstrated in Chapter 3, understanding code-switching behaviors, even within a single language pair, is limited when the data is drawn exclusively from a uniform speech setting. For instance, if all the data is collected from technical lectures, the resulting analysis can only capture the style of code-switching unique to that specific context. To address this, a more diverse and contextually rich approach to data collection is needed. One potential method is to categorize participants based on their relationships and familiarity, such as grouping strangers together, friends, parents talking with their children, or teachers interacting with students. Additionally, controlling the topics of conversation from scientific discussions to everyday activities could provide a more comprehensive range of code-switching styles. However, ethical considerations must be addressed when collecting data, especially when involving vulnerable groups or capturing sensitive interactions, to ensure that the research respects participants' privacy and cultural contexts. This approach would allow for a finer-grained analysis of how different social dynamics and communicative purposes influence code-switching behavior. Simply labeling the data as *conversational speech* may not be sufficient for in-depth theoretical study, as it overlooks the variations that

[1] https://openai.com/research/whisper

different speech settings and relationships can introduce.

Regarding the study of the T-index, future work could expand the analysis beyond the lexical properties of individual code-switched words to code-switched segments or n-grams. For instance, if the phrase *Black Hole* is identified as a code-switched segment, the two words together may function as a technical term, whereas analyzing *black* in isolation would not reveal the same property. This extension would allow researchers to capture more complex patterns of code-switching, where meaning and usage are tied to multi-word expressions rather than single words.

In regard to the relationship between POS and code-switching, future research could extend the analysis to consider larger syntactic structures and discourse-level phenomena. For example, exploring how code-switching interacts with sentence boundaries could reveal new insights into the planning and cognitive processes behind language alternation. It is possible that individuals plan to code-switch in advance, meaning that the trigger or influencing factors for a switch could be more distant within the sentence or discourse than previously considered. This broader approach would allow for a deeper examination of how code-switching is integrated into the overall structure of discourse, and how far-reaching the influences of syntax and discourse planning are on the occurrence of code-switching.

Given the promising results that large language models have demonstrated in multilingual and zero-shot tasks, future research could extend the exploration of code-switched text generation to include these advanced models. Even with the existing machine translation framework discussed in Chapter 6, incorporating pragmatic and sociolinguistic knowledge through multitask learning could yield better results.

Finally, incorporating human feedback represents another crucial research direction. Chapters 5 and 6 highlight the absence of benchmarks for synthetic code-switched text, and while downstream tasks have shown performance improvements, this does not necessarily reflect the naturalness of the generated text. Enhancements in performance may result from an increased frequency of plausible code-switched segments rather than genuine improvements in linguistic quality. Furthermore, traditional evaluation metrics such as WER or CER may introduce inaccuracies due to transliteration and spelling variations, potentially leading to false errors (Hamed et al., 2022). To address these challenges, developing a benchmark with human-evaluated code-switched text could offer more reliable assessments. Ideally, such a benchmark would include human-ranked examples derived from the same monolingual language pairs, providing a more accurate measure of text naturalness and coherence. Alternatively, creating a

corpus that records all possible switching points based on human consensus could also facilitate a thorough evaluation of generated patterns.

Moreover, longitudinal studies tracking code-switching usage over time within individuals or communities could offer insights into how bilingual language practices evolve and impact code-switching patterns. Such studies could reveal how different age groups perceive code-switching, some may find it disruptive, while others may view it as a normal part of communication. Leveraging findings from sociolinguistics will be essential in understanding how, when, and why code-switching occurs, ensuring that any systems developed are both theoretically sound and practically relevant.

Bibliography

Abdul-Zahra, S. (2010). Code-switching in language: An applied study. *Journal Of College Of Education For Women*, 21(1):283–296.

Abidi, K. and Smaïli, K. (2022). Cesar: A new metric to measure the level of code-switching in corpora - application to maghrebian dialects. In Arai, K., editor, *Intelligent Systems and Applications*, pages 793–803, Cham. Springer International Publishing.

Adel, H., Vu, N. T., Kraus, F., Schlippe, T., Li, H., and Schultz, T. (2013a). Recurrent neural network language modeling for code switching conversational speech. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8411–8415.

Adel, H., Vu, N. T., and Schultz, T. (2013b). Combination of recurrent neural networks and factored language models for code-switching language modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 206–211.

Agarwal, P., Sharma, A., Grover, J., Sikka, M., Rudra, K., and Choudhury, M. (2017). I may talk in english but gaali toh hindi mein hi denge : A study of english-hindi code-switching and swearing pattern on social networks. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 554–557.

Aguilar, G., Kar, S., and Solorio, T. (2020). Lince: A centralized benchmark for linguistic code-switching evaluation. *ArXiv*, abs/2005.04322.

Aikhenvald, A. (2007). *Grammars in Contact A Cross-Linguistic Perspective*, pages 1–66.

AlGhamdi, F., Molina, G., Diab, M., Solorio, T., Hawwari, A., Soto, V., and Hirschberg, J. (2016). Part of speech tagging for code switched data. In Diab, M., Fung, P., Ghoneim, M., Hirschberg, J., and Solorio, T., editors, *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107, Austin, Texas. Association for Computational Linguistics.

Ali, A., Chowdhury, S. A., Hussein, A., and Hifny, Y. (2021). Arabic code-switching speech recognition using monolingual data. *CoRR*, abs/2107.01573.

Alvanoudi, A. (2018). Language contact, borrowing and code switching: A case study of australian greek. *Journal of Greek Linguistics*, 18(1):3 – 44.

Attia, M., Samih, Y., Elkahky, A., Mubarak, H., Abdelali, A., and Darwish, K. (2019). POS tagging for improving code-switching identification in Arabic. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 18–29, Florence, Italy. Association for Computational Linguistics.

Austin, P. K. and Sallabank, J. (2011). *The Cambridge handbook of endangered languages*. Cambridge University Press.

Azuma, S. (1997). *Shakai gengogaku nyumon [An introduction to sociolinguistics]*. Kenkyusha, Tokyo.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477.

Barman, U., Das, A., Wagner, J., and Foster, J. (2014). Code mixing: A challenge for language identification in the language of social media.

Barnett, R., Codó, E., Eppler, E., Forcadell, M., Gardner-Chloros, P., Van Hout, R., Moyer, M., Torras, M. C., Turell, M. T., Sebba, M., et al. (2000). The lides coding manual: A document for preparing and analyzing language interaction data version 1.1–july 1999. *International Journal of Bilingualism*, 4(2):131–271.

Beatty-Martínez, A. L., Navarro-Torres, C. A., and Dussias, P. E. (2020). Codeswitching: A bilingual toolkit for opportunistic speech planning. *Frontiers in Psychology*, 11:532799.

Belazi, H. M., Rubin, E., and Toribio, A. J. (1994). Code switching and x-bar theory: the fuctional head constraint. *Linguistic Inquiry*, 25:221–238.

Berk-Seligson, S. (1986). Linguistic constraints on intrasentential code-switching: A study of spanish/hebrew bilingualism. *Language in Society*, 15(3):313–348.

Bhat, G., Choudhury, M., and Bali, K. (2016). Grammatical constraints on intra-sentential code-switching: From theories to working models. *CoRR*, abs/1612.04538.

Bhattacharya, D., Chi, J., Hirschberg, J., and Bell, P. (2023). Capturing formality in speech across domains and languages. In *Interspeech 2023*, pages 1030–1034.

Bhuvanagiri, K. and Kopparapu, S. (2010). An approach to mixed language automatic speech recognition. *Oriental COCOSDA, Kathmandu, Nepal*.

Blom, J.-P. and Gumperz, J. J. (1972). Social meaning in linguistic structure: Code-switching in norway. In Gumperz, J. J. and Hymes, D., editors, *Directions in Sociolinguistics*. Holt, Rinehart and Winston, New York.

Boztepe, E. (2003). Issues in code-switching: Competing theories and models.

Broersma, M. (2009). Triggered codeswitching between cognate languages. *Bilingualism: Language and Cognition*, 12(4):447–462.

Broersma, M., Carter, D., Donnelly, K., and Konopka, A. (2020). Triggered codeswitching: Lexical processing and conversational dynamics. *Bilingualism: Language and Cognition*, 23(2):295–308.

Broersma, M. and De Bot, K. (2006). Triggered codeswitching: A corpus-based evaluation of the original triggering hypothesis and a new alternative. *Bilingualism: Language and Cognition*, 9(1):1–13.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bu, H., Du, J., Na, X., Wu, B., and Zheng, H. (2017). AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline. In *Oriental COCOSDA 2017*. Submitted.

Bultena, S., Dijkstra, T., and van Hell, J. (2014). Language switch costs in sentence comprehension depend on language dominance: Evidence from self-paced reading. *Bilingualism: Language and Cognition*, 18:1–17.

Calvillo, J., Fang, L., Cole, J., and Reitter, D. (2020). Surprisal predicts code-switching in Chinese-English bilingual text. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4029–4039, Online. Association for Computational Linguistics.

Chandu, K., Manzini, T., Singh, S., and Black, A. W. (2018). Language informed modeling of code-switched text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 92–97.

Chang, C.-T., Chuang, S.-P., and Lee, H.-Y. (2019). Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. In *Proc. Interspeech 2019*, pages 554–558.

Chang, J. C. and Lin, C. (2014). Recurrent-neural-network for language detection on twitter code-switching corpus. *CoRR*, abs/1412.4314.

Chi, J. and Bell, P. (2022). Improving code-switched ASR with linguistic information. In *Proceedings of COLING*, pages 7171–7176.

Chi, J. and Bell, P. (2024). Analyzing the role of part-of-speech in code-switching: A corpus-based study. In Graham, Y. and Purver, M., editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1712–1721, St. Julian's, Malta. Association for Computational Linguistics.

Chi, J., Lu, B., Eisner, J., Bell, P., Jyothi, P., and Ali, A. M. (2023). Unsupervised Code-switched Text Generation from Parallel Text. In *Proc. INTERSPEECH 2023*, pages 1419–1423.

Chi, J., Wallington, E., and Bell, P. (2024). Characterizing code-switching: Applying Linguistic Principles for Metric Assessment and Development. In *Proc. INTERSPEECH 2024*.

Chung, H. H. (2006). Code switching as a communicative strategy: A case study of korean–english bilinguals. *Bilingual research journal*, 30(2):293–307.

Clyne, M. (1972). *Perspectives on Language Contact: Based on a Study of German in Australia*. Hawthorn Press.

Clyne, M. (2003). *Dynamics of Language Contact: English and Immigrant Languages*. Cambridge Approaches to Language Contact. Cambridge University Press.

Clyne, M. G. (1967). Transference and triggering; observations on the language assimilation of postwar german-speaking migrants in australia. 2010.

Clyne, M. G. (1980). Triggering and language processing. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 34(4):400.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020a). Unsupervised cross-lingual representation learning at scale. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Conneau, A., Wu, S., Li, H., Zettlemoyer, L., and Stoyanov, V. (2020b). Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034.

Cook, V. (2016). *Second Language Learning and Language Teaching: Fifth Edition*. Taylor and Francis, London, 5 edition.

Das, A. and Gambäck, B. (2014). Identifying languages at the word level in code-mixed Indian social media text. In Sharma, D. M., Sangal, R., and Pawar, J. D., editors, *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

Deuchar, M. (2011). The miami corpus: Documentation file.

Deuchar, M., Davies, P., Herring, J. R., Couto, M. C. P., and Carter, D. (2014). *5. Building Bilingual Corpora*, pages 93–110. Multilingual Matters, Bristol, Blue Ridge Summit.

Deuchar, M., Donnelly, K., and Piercy, C. (2016). 'mae pobl monolingual yn minority': Factors favouring the production of code switching by welsh–english bilingual speakers. In *Sociolinguistics in Wales*, pages 209–239. Springer.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Diwan, A., Vaideeswaran, R., Shah, S., Singh, A., Raghavan, S., Khare, S., Unni, V., Vyas, S., Rajpuria, A., Yarra, C., Mittal, A., Ghosh, P. K., Jyothi, P., Bali, K., Seshadri, V., Sitaram, S., Bharadwaj, S., Nanavati, J., Nanavati, R., and Sankaranarayanan, K. (2021). MUCS 2021: Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages. In *Proc. Interspeech 2021*, pages 2446–2450.

Donnelly, K. and Deuchar, M. (2011). The bangor autoglosser: A multilingual tagger for conversational text.

Du Bois, J., Schuetze-Coburn, S., Cumming, S., and Paolino, D. (1983). *Outline of discourse transcription*, pages 45–89.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.

Edwards, H. T. (1992). *Applied Phonetics: The Sounds of American English*. Singular Publishing Group.

Gambäck, B. and Das, A. (2016). Comparing the level of code-switching in corpora. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855, Portorož, Slovenia. European Language Resources Association (ELRA).

Gambäck, B. and Das, A. (2014). On Measuring the Complexity of Code-Mixing.

Gao, Y., Feng, J., Liu, Y., Hou, L., Pan, X., and Ma, Y. (2019). Code-Switching Sentence Generation by Bert and Generative Adversarial Networks. In *Proc. Interspeech 2019*, pages 3525–3529.

Gardner-Chloros, P. (2008). *Bilingual Speech Data: Criteria for Classification*, chapter 4, pages 53–72. John Wiley & Sons, Ltd.

Gardner-Chloros, P. (2009). *Code-switching*. Cambridge University Press.

Garg, S., Parekh, T., and Jyothi, P. (2018). Code-switched language models using dual RNNs and same-source pretraining. In Riloff, E., Chiang, D., Hockenmaier, J.,

and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3078–3083, Brussels, Belgium. Association for Computational Linguistics.

Gautam, D., Kodali, P., Gupta, K., Goel, A., Shrivastava, M., and Kumaraguru, P. (2021). CoMeT: Towards code-mixed translation using parallel monolingual sentences. In Solorio, T., Chen, S., Black, A. W., Diab, M., Sitaram, S., Soto, V., Yilmaz, E., and Srinivasan, A., editors, *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55, Online. Association for Computational Linguistics.

Gella, S., Bali, K., and Choudhury, M. (2014). "ye word kis lang ka hai bhai?" testing the limits of word level language identification. In Sharma, D. M., Sangal, R., and Pawar, J. D., editors, *Proceedings of the 11th International Conference on Natural Language Processing*, pages 368–377, Goa, India. NLP Association of India.

Ghosh, S., Ghosh, S., and Das, D. (2017). Complexity metric for code-mixed social media text. *CoRR*, abs/1707.01183.

Goldstein, B. (2000). *Resource guide on cultural and linguistic diversity*. Singular resource guide series. Singular Pub. Group, San Diego, Calif.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Grosjean, F. (1982). *Life with two languages: An introduction to bilingualism*. Harvard University Press.

Gumperz, J. (1982). *Discourse Strategies*. Studies in Interactional Sociolinguistics. Cambridge University Press.

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

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

Guzmán, G. A., Ricard, J., Serigos, J., Bullock, B., and Toribio, A. J. (2017). Moving code-switching research toward more empirically grounded methods. In *CDH@ TLT*, pages 1–9.

Guzman, G. A., Serigos, J., Bullock, B., and Toribio, A. J. (2016). Simple tools for exploring variation in code-switching for linguists. In *Proceedings of the second workshop on computational approaches to code switching*, pages 12–20.

Guzmán, G., Ricard, J., Serigos, J., Bullock, B. E., and Toribio, A. J. (2017). Metrics for Modeling Code-Switching Across Corpora. In *Proc. Interspeech 2017*, pages 67–71.

Hamed, I., Hussein, A., Chellah, O., Chowdhury, S., Mubarak, H., Sitaram, S., Habash, N., and Ali, A. (2022). Benchmarking evaluation metrics for code-switching automatic speech recognition.

Haspelmath, M. (2009). *II. Lexical borrowing: Concepts and issues*, pages 35–54. De Gruyter Mouton, Berlin, New York.

He, Y., Ma, Y., Way, A., and Genabith, J. (2010). Integrating n-best smt outputs into a tm system. *He, Yifan and Ma, Yanjun and Way, Andy and van Genabith, Josef (2010) Integrating N-best SMT outputs into a TM system. In: COLING 2010 - 23rd International Conference on Computational Linguistics, 23-27 August 2010, Beijing, China.*

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., and Estève, Y. (2018). TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer*, pages 198–208. Springer International Publishing.

Hoffmann, C. (2014). *Introduction to bilingualism*. Routledge.

Hokamp, C. and Liu, Q. (2017). Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546.

Holmes, J. (2001). *An Introduction to Sociolinguistics*. Insights Into Human Geography. Longman.

Holmes, J. and Wilson, N. (2017). *An Introduction to Sociolinguistics*. Routledge, London, 5 edition.

Hymes, D. (1972). Models of the interaction of language and social life. *Directions in sociolinguistics: The ethnography of communication/Holt, Rinehart & Winston*.

Hymes, D. H. (1971). Pidginization and creolization of languages : proceedings of a conference held at the university of the west indies, mona, jamaica, april, 1968 / edited by dell hymes. Cambridge. University Press.

Isurin, L., Bot, K., and Winford, D. (2009). Multidisciplinary approaches to code switching. *Multidisciplinary Approaches to Code Switching*, pages 1–384.

Jaech, A., Mulcaire, G., Hathi, S., Ostendorf, M., and Smith, N. A. (2016). Hierarchical character-word models for language identification.

Jennifer A. Vu, A. L. B. and Howes, C. (2010). Early cases of code-switching in mexican-heritage children: Linguistic and sociopragmatic considerations. *Bilingual Research Journal*, 33(2):200–219.

Jisa, H. (2000). Language mixing in the weak language: Evidence from two children. *Journal of Pragmatics*, 32(9):1363–1386. Codeswitching.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Khanuja, S., Dandapat, S., Sitaram, S., and Choudhury, M. (2020). A new dataset for natural language inference from code-mixed conversations. In Solorio, T., Choudhury, M., Bali, K., Sitaram, S., Das, A., and Diab, M., editors, *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 9–16, Marseille, France. European Language Resources Association.

King, B. and Abney, S. (2013). Labeling the languages of words in mixed-language documents using weakly supervised methods. In Vanderwende, L., Daumé III, H., and Kirchhoff, K., editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119, Atlanta, Georgia. Association for Computational Linguistics.

Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.

Kitaev, N. and Klein, D. (2018). Constituency parsing with a self-attentive encoder. *CoRR*, abs/1805.01052.

Kodali, P., Goel, A., Choudhury, M., Shrivastava, M., and Kumaraguru, P. (2022). Symcom-syntactic measure of code mixing a study of english-hindi code-mixing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 472–480.

Kohler, J. (1998). Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 1, pages 417–420 vol.1.

Kootstra, G. J., Dijkstra, T., and van Hell, J. G. (2020). Interactive alignment and lexical triggering of code-switching in bilingual dialogue. *Frontiers in Psychology*, 11.

Kootstra, G. J., van Hell, J. G., and Dijkstra, T. (2012). Priming of code-switches in sentences: The role of lexical repetition, cognates, and language proficiency – corrigendum. *Bilingualism: Language and Cognition*, 16:476 – 476.

Kunchukuttan, A., Mehta, P., and Bhattacharyya, P. (2018). The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Labov, W. (1972). Some principles of linguistic methodology. *Language in society*, 1(1):97–120.

Lee, G. and Li, H. (2020). Modeling code-switch languages using bilingual parallel corpus. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 860–870, Online. Association for Computational Linguistics.

Lee, G., Yue, X., and Li, H. (2019). Linguistically Motivated Parallel Data Augmentation for Code-Switch Language Modeling. In *Proc. Interspeech 2019*, pages 3730–3734.

Leung, C. (2006). Codeswitching in print advertisements in hong kong and sweden.

Li, K., Li, J., Ye, G., Zhao, R., and Gong, Y. (2019). Towards Code-switching ASR for End-to-end CTC Models. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6076–6080. ISSN: 2379-190X.

Li, Y. and Fung, P. (2012). Code-switch language model with inversion constraints for mixed language speech recognition. In Kay, M. and Boitet, C., editors, *Proceedings of COLING 2012*, pages 1671–1680, Mumbai, India. The COLING 2012 Organizing Committee.

Li, Y. and Fung, P. (2014). Language modeling with functional head constraint for code switching speech recognition. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 907–916, Doha, Qatar. Association for Computational Linguistics.

Lipski, J. M. (1985). *Linguistic aspects of Spanish-English language switching / by John M. Lipski.* Special studies ; no. 25. Center for Latin American Studies, Arizona State University, Tempe, Ariz.

Liu, H. (2019). Attitudes toward different types of chinese-english code-switching. *Sage Open*, 9(2):2158244019853920.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Luo, N., Jiang, D., Zhao, S., Gong, C., Zou, W., and Li, X. (2018). Towards end-to-end code-switching speech recognition. *CoRR*, abs/1810.13091.

Lyu, D.-C., Lyu, R.-y., Chiang, Y.-c., and Hsu, C.-n. (2006). Speech recognition on code-switching among the chinese dialects. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I.

Lyu, D.-C., Tan, T.-P., Chng, E. S., and Li, H. (2010). SEAME: a Mandarin-English code-switching speech corpus in south-east asia. In *Proc. Interspeech 2010*, pages 1986–1989.

Lyudovyk, T. and Pylypenko, V. (2014). Code-switching speech recognition for closely related languages. In *Spoken Language Technologies for Under-Resourced Languages*.

Madhumani, G. R., Shah, S., Abraham, B., Joshi, V., and Sitaram, S. (2020). Learning not to discriminate: Task agnostic learning for improving monolingual and code-switched speech recognition.

McKay, S. L. S. L. (2002 - 2002). *Teaching English as an international language : rethinking goals and approaches / Sandra Lee McKay.* Oxford handbooks for language teachers. Oxford University Press, Oxford.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting Similarities among Languages for Machine Translation. *arXiv e-prints*.

Milroy, L. and Muysken, P. (1995). *Introduction: code-switching and bilingualism research*, page 1–14. Cambridge University Press.

Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., and Levin, L. (2016). PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee.

Muller, B., Elazar, Y., Sagot, B., and Seddah, D. (2021). First align, then predict: Understanding the cross-lingual ability of multilingual BERT. *arXiv e-prints*.

Munteanu, D. S. and Marcu, D. (2007). Isi chinese-english automatically extracted parallel text. *Linguistic Data Consortium, Philadelphia*.

Muysken, P. (2011). *Code-switching*, page 301–314. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Myers-Scotton, C. (1993). *Social Motivations For Codeswitching: Evidence from Africa*. Oxford University Press.

Myers-Scotton, C. (1997). *Duelling Languages: Grammatical Structure in Codeswitching*. Clarendon Press.

Myers-Scotton, C. (2006). *Multiple Voices: An Introduction to Bilingualism*. Wiley.

Neveu, A., McDonald, M., and Kaushanskaya, M. (2022). Testing the triggering hypothesis: Effect of cognate status on code-switching and disfluencies. *Languages*, 7(4).

Ney, H., Essen, U., and Kneser, R. (1994). On structuring probabilistic dependences in stochastic language modelling. *Comput. Speech Lang.*, 8:1–38.

Nguyen, D. and Doğruöz, A. S. (2013). Word level language identification in online multilingual communication. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu,

K., and Bethard, S., editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 857–862, Seattle, Washington, USA. Association for Computational Linguistics.

Novak, J., Minematsu, N., and Hirose, K. (2016). Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938.

Ostapenko, A., Wintner, S., Fricke, M., and Tsvetkov, Y. (2022). Speaker information can guide models to better inductive biases: A case study on predicting code-switching. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3853–3867, Dublin, Ireland. Association for Computational Linguistics.

Pattichis, R., LaCasse, D., Trawick, S., and Cacoullos, R. (2023). Code-switching metrics using intonation units. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16840–16849.

Piergallini, M., Shirvani, R. A., Gautam, G. S., and Chouikha, M. F. (2016). Word-level language identification and predicting codeswitching points in swahili-english language data. In *CodeSwitch@EMNLP*.

Poplack, S. (1980). Sometimes i'll start a sentence in spanish y termino en espanol: toward a typology of code-switching1.

Poplack, S. (1988). Contrasting patterns of code-switching in two communities. *Codeswitching: Anthropological and sociolinguistic perspectives*, 48:215–244.

Poplack, S. (2013). "sometimes i'll start a sentence in spanish y termino en español": Toward a typology of code-switching. *Linguistics*, 51(s1):11–14.

Poplack, S. and Dion, N. (2012). Myths and facts about loanword development. *Language Variation and Change*, 24(3):279–315.

Post, R. E. (2015). *The impact of social factors on the use of Arabic-French code-switching in speech and IM in Morocco*. PhD thesis.

Pratapa, A., Bhat, G., Choudhury, M., Sitaram, S., Dandapat, S., and Bali, K. (2018). Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of ACL*, pages 1543–1553.

Redlinger, W. E. and Park, T.-Z. (1980). Language mixing in young bilinguals. *Journal of child language*, 7(2):337–352.

Reitmaier, T., Wallington, E., Kalarikalayil Raju, D., Klejch, O., Pearson, J., Jones, M., Bell, P., and Robinson, S. (2022). Opportunities and challenges of automatic speech recognition systems for low-resource language speakers. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Roche, J. (1993). Language selection and switching in strasbourg. penelope gardner-chloros. oxford: Clarendon press, 1991. pp. 218. *American Journal of Germanic Linguistics and Literatures*, 5(2):207–211.

Romaine, S. (1989). *Bilingualism*. Language in Society, No.13. Basil Blackwell.

Romaine, S. (2000). *Language in society: An introduction to sociolinguistics*. OUP Oxford.

Sahib, H., Hanafiah, W., Aswad, M., Yassi, A. H., and Mashhadi, F. (2021). Syntactic configuration of code-switching between indonesian and english: Another perspective on code-switching phenomena. *Education Research International*, 2021(1):3402485.

Samanta, B., Reddy, S., Jagirdar, H., Ganguly, N., and Chakrabarti, S. (2019). A Deep Generative Model for Code-Switched Text.

Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., and Solorio, T. (2016). Multilingual code-switching identification via LSTM recurrent neural networks. In Diab, M., Fung, P., Ghoneim, M., Hirschberg, J., and Solorio, T., editors, *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas. Association for Computational Linguistics.

Sankoff, D. and Poplack, S. (1981). A formal grammar for code-switching. *Paper in Linguistics*, 14(1):3–45.

Seki, H., Watanabe, S., Hori, T., Roux, J. L., and Hershey, J. R. (2018). An end-to-end language-tracking speech recognizer for mixed-language speech. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4919–4923.

Shah, S., Abraham, B., M, G. R., Sitaram, S., and Joshi, V. (2020). Learning to recognize code-switched speech without forgetting monolingual speech recognition.

Shan, C., Weng, C., Wang, G., Su, D., Luo, M., Yu, D., and Xie, L. (2019). Investigating end-to-end speech recognition for mandarin-english code-switching. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6056–6060.

Shen, H.-P., Wu, C.-H., Yang, Y.-T., and Hsu, C.-S. (2011). Cecos: A chinese-english code-switching speech database. In *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, pages 120–123.

Shoko, Y. O. et al. (2003). Metaphorical code-switching revisited. *Ferris Jogakuin University Bulletin of the Faculty of Letters*, 38:A137–A153.

Simensen, A. (1998). *Teaching a Foreign Language: Principles and Procedures*. Fagbokforlaget.

Sitaram, S., Chandu, K. R., Rallabandi, S. K., and Black, A. W. (2019). A survey of code-switched speech and language processing. *CoRR*, abs/1904.00784.

Sivasankaran, S., Srivastava, B. M. L., Sitaram, S., Bali, K., and Choudhury, M. (2018). Phone merging for code-switched speech recognition. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 11–19, Melbourne, Australia. Association for Computational Linguistics.

Soares, A. P., Oliveira, H., Ferreira, M., Comesaña, M., Macedo, A. F., Ferré, P., Acuña-Fariña, C., Hernández-Cabrera, J., and Fraga, I. (2019). Lexico-syntactic interactions during the processing of temporally ambiguous l2 relative clauses: An eye-tracking study with intermediate and advanced portuguese-english bilinguals. *PLOS ONE*, 14(5):1–27.

Solorio, T. and Liu, Y. (2008a). Learning to predict code-switching points. In *Conference on Empirical Methods in Natural Language Processing*.

Solorio, T. and Liu, Y. (2008b). Part-of-speech tagging for english-spanish code-switched text. pages 1051–1060.

Soto, V., Cestero, N., and Hirschberg, J. (2018). The Role of Cognate Words, POS Tags and Entrainment in Code-Switching. In *Interspeech 2018*, pages 1938–1942. ISCA.

Soto, V. and Hirschberg, J. (2018). Joint part-of-speech and language ID tagging for code-switched data. In Aguilar, G., AlGhamdi, F., Soto, V., Solorio, T., Diab, M., and Hirschberg, J., editors, *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Srivastava, V. and Singh, M. (2021). Challenges and limitations with the metrics measuring the complexity of code-mixed text. *CoRR*, abs/2106.10123.

Stefanich, S., Cabrelli, J., Hilderman, D., and Archibald, J. (2019). The morphophonology of intraword codeswitching: Representation and processing. *Frontiers in Communication*, 4.

Sulminski, A. M. (2022). *Examining Patterns of Code-Switching in Preschool-Age Spanish-English Bilingual Children in Formal and Informal Contexts*. Bowling Green State University.

Sybrine Bultena, T. D. and van Hell, J. G. (2015). Switch cost modulations in bilingual sentence processing: evidence from shadowing. *Language, Cognition and Neuroscience*, 30(5):586–605.

Taljard, E. and Bosch, S. E. (2006). A comparison of approaches to word class tagging: Disjunctively vs. conjunctively written bantu languages. *Nordic journal of African studies*, 15(4).

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401.

Terblanche, M., Olaleye, K., and Marivate, V. (2024). Prompting towards alleviating code-switched data scarcity in under-resourced languages with gpt as a pivot. *ArXiv*, abs/2404.17216.

Thomason, S. G. (2001). *Language contact / Sarah G. Thomason.* Edinburgh University Press, Edinburgh.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In Chair), N. C. C., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ting, S.-H. and Yeo, K.-L. (2020). Code-switching functions in facebook wallposts. 20:7–18.

Van Assche, E., Duyck, W., and Hartsuiker, R. (2012). Bilingual word recognition in a sentence context. *Frontiers in Psychology*, 3.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vu, N. T., Lyu, D.-C., Weiner, J., Telaar, D., Schlippe, T., Blaicher, F., Chng, E.-S., Schultz, T., and Li, H. (2012). A first speech recognition system for mandarin-english code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892.

Weiner, J., Vu, N. T., Telaar, D., Metze, F., Schultz, T., Lyu, D.-C., Chng, E.-S., and Li, H. (2012). Integration of language identification into a recognition system for spoken conversations containing code-switches. In *Spoken Language Technologies for Under-Resourced Languages*.

Winata, G. I., Madotto, A., Wu, C.-S., and Fung, P. (2018). Code-switching language modeling using syntax-aware multi-task learning. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 62–67.

Winata, G. I., Madotto, A., Wu, C.-S., and Fung, P. (2019). Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280.

Winford, D. (2003). *An Introduction to Contact Linguistics*.

Wintner, S., Shehadi, S., Zeira, Y., Osmelak, D., and Nov, Y. (2023). Shared Lexical Items as Triggers of Code Switching. *Transactions of the Association for Computational Linguistics*, 11:1471–1484.

Wu, C., Chiu, Y., Shia, C., and Lin, C. (2006). Automatic segmentation and identification of mixed-language speech using delta-bic and lsa-based gmms. *IEEE Transactions on Speech and Audio Processing*, 14(1):266–275.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yong, Z. X., Zhang, R., Forde, J., Wang, S., Subramonian, A., Lovenia, H., Cahyawijaya, S., Winata, G., Sutawika, L., Cruz, J. C. B., Tan, Y. L., Phan, L., Phan, L., Garcia, R., Solorio, T., and Aji, A. F. (2023). Prompting multilingual large language models to generate code-mixed texts: The case of south East Asian languages. In Winata, G., Kar, S., Zhukova, M., Solorio, T., Diab, M., Sitaram, S., Choudhury, M., and Bali, K., editors, *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 43–63, Singapore. Association for Computational Linguistics.

Yow, W. Q., Tan, J. S. H., and Flynn, S. (2017). Code-switching as a marker of linguistic competence in bilingual children. *Bilingualism: Language and Cognition*, 21:1075–1090.

Yu, H., Hu, Y., Qian, Y., Jin, M., Liu, L., Liu, S., Shi, Y., Qian, Y., Lin, E., and Zeng, M. (2023). Code-switching text generation and injection in mandarin-english asr. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yusuf, Y. Q., Fata, I. A., and Chyntia, C. (2020). Types of indonesian-english code-switching employed in a novel. *Kasetsart Journal of Social Sciences*, 41(1):208–213.

Yılmaz, E., van den Heuvel, H., and van Leeuwen, D. (2018). Acoustic and Textual Data Augmentation for Improved ASR of Code-Switching Speech. In *Proc. Interspeech 2018*, pages 1933–1937.

Zeng, Z., Khassanov, Y., Pham, V. T., Xu, H., Chng, E. S., and Li, H. (2019). On the End-to-End Solution to Mandarin-English Code-Switching Speech Recognition. In *Proc. Interspeech 2019*, pages 2165–2169.

Zhang, R., Cahyawijaya, S., Cruz, J. C. B., Winata, G., and Aji, A. F. (2023). Multilingual large language models are not (yet) code-switchers. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

Zirker, K. A. H. (2007). *Intrasentential vs. intersentential code switching in early and late bilinguals*. Brigham Young University.