Multilingual and Code-Switching Speech Recognition
2022 Eighth Frederick Jelinek Memorial Summer Workshop

Ahmed Ali, Shammur Chowdhury, Lucas Ondel, Matthew Wiesner
Ondrej Klejch, Kenton Murray, Jie Chi, Amir Hussein
Injy Hamed, Lea-Marie Lam-Yee-Mui, Barah Fazili, Brian Yan
Electra Wallington, Brian Lu, Debasmita Bhattacharya, Dorsa Zeinali
Oumnia Chellah, Danielle Cartagenes, Peter Bell, Nizar Habash
Preethi Jyothi, Sunayana Sitaram, Ravi Mamindlapalli, Shinji Watanabe
Jan Trmal, Najim Dehak, Sanjeev Khudanpur

August 22, 2023

Abstract

The development of automatic speech recognition (ASR) systems that can handle code-switched speech is challenging due to data scarcity, grammatical structure complexity, and domain mismatch. Several approaches have been proposed to address this challenge, including data augmentation techniques for synthesizing code-switched text, leveraging multilingual transformers for cross-lingual transfer, and generating code-switching translations using multilingual machine translation and natural language processing systems. Another proposed method uses a zero-shot learning methodology for code-switching ASR by augmenting monolingual data with artificially generated code-switching text. Further methods involve random lexical replacements and Equivalence Constraints while exploiting aligned translation pairs to generate random and grammatically valid code-switching content. Finally, code-switching ASR lacks robust and fair evaluation metrics as language pairs with different scripts can lead to cross-transcribe words that create a large number of homophones in hypothesis and reference transcription.

During the *six-week* summer school workshop, we focused on investigating novel techniques to build practical large vocabulary continuous speech recognition systems capable of dealing with monolingual and code-switching spoken utterances. We addressed the following four challenges in code-switching ASR: (*i*) Understanding why code-switching happens in human speech and how to measure such phenomenon in spoken and written content. (*ii*) Generating code-switching speech and text using data augmentation to train large models, (*iii*) Building ASR systems capable of dealing with monolingual and code-switching speech, and finally (*iv*) Evaluating code-switching ASR system.

Linguistic Aspects of Code-Switching: Before building a robust code-switching automatic speech recognition system, we should understand how to characterize code-switching in textual and spoken data to model such phenomenon efficiently. We study several metrics to evaluate code-switching; M-metric, I-metric, burstiness, and memory. We benchmark them across different corpora across a wide range of languages, named: Scottish-Gaelic - English, Mandarin - English, Spanish - English, Egyptian-Arabic - English and South African languages (isiXhosa, isiZulu, seSotho and seTswana) - English. We show stylistic differences across various langauge(s)-pairs. We highlight challenges in data processing for such data, as simple as what defines a *token* in code-switch text and how to define the boundary of language detection. Finally, we highlight the outcomes and limitations of these metrics. We identify potential causes of the code-switching variation in speech and text. We examine the influence a speakers' characteristics on code-switching behaviour: namely, age and gender. Finally, we do a more focused study on the characteristics of the speech or the domain: namely, *formality*. What role does formality play in code-switching? How to measure formality?

Synthetic Text Code-Switching Generation: Due to the challenges of building real code-switching data in the wild, we investigate data augmentation to create spoken and written code-switching corpus. We propose a zero-shot learning methodology by augmenting monolingual data with artificially generating Code-switching text. We perform lexical replacements using parallel corpora and alignments where code-switching points are either randomly chosen or learnt using a sequence-to-sequence model. Finally, We introduce a novel approach of forcing a multilingual machine translation system trained on monolingual data to generate CS translations.

Automatic Speech Recognition for Code-Switched Speech: Building robust speech recognition systems capable of dealing with multilingual and code-switching speech poses several challenges; the fact that code-switching is unwritten phenomenon that rarely occurs in most text and transcribed speech corpora makes it challenging in build speech recognition system using real data. We address here the following three research topics: (*i*) Introducing speech collage to generate multilingual and code-switching speech data, (*ii*) Building code-switching ASR system using monolingual data and (*iii*) Using self-supervised models for code-switching ASR.

Evaluating Code-Switching Speech Recognition Systems: To improve a speech recognition system, we should be able to evaluate the accuracy and report error rate using robust and fair evaluation metrics. The current word error rate evaluation (WER) method assumes a single reference transcription for each speech segment. However, in code-switching language pairs with different scripts, there is a tendency to cross-transcribe words, which creates a large number in both hypothesis and reference. Furthermore, we may expect partial (cross-script) transcription. Thanks to recent end-to-end ASR techniques such as word-pieces.

Team Members

Team Leader

Ahmed Ali	Qatar Computing Research Institute

Senior Researchers

Shammur Chowdhury	Qatar Computing Research Institute
Lucas Ondel	University Paris-Saclay, CNRS, LISN
Matthew Wiesner	Johns Hopkins University
Ondrej Klejch	University of Edinburgh
Kenton Murray	Johns Hopkins University

Graduate Students

Jie Chi	University of Edinburgh
Amir Hussein	KANARI AI and Johns Hopkins University
Injy Khairy Hamed	New York University Abu Dhabi
Lea-Marie Lam-Yee-Mui	Vocapia Research and University Paris-Saclay, CNRS, LISN
Barah Fazili	IIT Bombay
Brian Yan	Carnegie Mellon University
Electra Wallington	University of Edinburgh
Brian Lu	Johns Hopkins University
Debasmita Bhattacharya	Columbia University

Undergraduate Students

Dorsa Zeinali	Carnegie Mellon University
Oumnia Chellah	Stanford University
Danielle Cartagenes	University of Illinois Chicago

Senior Affiliates

Peter Bell	University of Edinburgh
Nizar Habash	New York University Abu Dhabi
Preethi Jyothi	IIT Bombay
Sunayana Sitaram	Microsoft Research India
Ravi Mamindlapalli	Uniphore
Shinji Watanabe	Carnegie Mellon University
Jan Trmal	Johns Hopkins University
Najim Dehak	Johns Hopkins University
Sanjeev Khudanpur	Johns Hopkins University

Acknowledgements
The work reported here was carried out during the 2022 Jelinek Memorial Summer Workshop on Speech and Language Technologies at Johns Hopkins University, which was supported with unrestricted gifts from Amazon, Microsoft, and Google. We would like to thank QCRI for the computing and infrastructure resources.

Contents

Part I

Linguistic Aspects of Code-Switching

This section offers insights into understanding code-switching. Given that code-switching is not a monolithic phenomenon across all languages, how do we formally characterize code-switching in speech and text? Is it the number of switches or language distribution? is the code-switching style different across domains from the same language? From the domain of the speech to speaker characteristics or speaker relationships, what triggers code-switching? Could it be a sociopolitical relationship, gender, age, formality, etc. In this section. we address the following research questions: (*i*) Characterizing Code-Switching and (*ii*) Predicting Code-Switching Behaviour.

Characterizing Code-Switching: Before building a robust code-switching automatic speech recognition system, we should understand how to characterize code-switching in textual and spoken data to model such phenomenon efficiently. We study several metrics to evaluate code-switching; M-metric, I-metric, burstiness, and memory. We benchmark them across different corpora across a wide range of languages, named: Scottish-Gaelic - English, Mandarin - English, Spanish - English, Egyptian-Arabic - English and South African languages (isiXhosa, isiZulu, seSotho and seTswana) - English. We show stylistic differences across various langauge(s)-pairs. We highlight challenges in data processing for such data, as simple as what defines a *token* in code-switch text and how to define the boundary of language detection. Finally, we highlight outcomes and limitations of these metrics.

Predicting Code-Switching Behaviour: We identify potential causes of the code-switching variation in speech and text. In this chapter, we examine the influence a speakers' characteristics on code-switching behaviour: namely, age and gender. Furthermore, we do more focused study on the characteristics of the speech or the domain: namely, formality. What role does formality play in code-switching? How to measure formality? Our results suggest that being a factor that can influence code-switching behaviour in spoken and written content across several languages.

We investigate the following research questions in the linguistic aspects of code-switching:

- How to evaluate code-switching in speech and text?
- What is the impact of age and gender on code-switching?
- How does formality impact code-switching?

Chapter 1

Characterizing Code-Switching

Electra Wallington, Debasmita Bhattacharya, Danielle Cartagenes, Peter Bell

1.1 Introduction and Motivation

Since our goal is to develop methods and build automatic speech recognition (ASR) systems that can handle code-switching, it is important we understand that code-switching is not a monolithic phenomenon. See the following examples:

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

(2) crystal systems के crystal structures को दिखाना उदाहरण के लिए cubic hexagonal और rhombohedral
`and showing the crystal structures **of different** crystal systems **for example** cubic, hexagonal **and** rhombohedral'

(3) **Chomi, hewethu i**worry **yam kengoku** it's the next meeting on friday and **kengoku** I'm going out **chomi nomfana wam**, how are things going to happen?
`Friend, (hewethu) my worry **now** its the next meeting on Friday and **now** Im going out **friend with my man,** How are things going to happen?'

Whilst all coming from what could be considered code-switching data-sets (specifically, transcripts of code-switched speech), what constitutes code-switching in each of these snippets is extremely different. In (1)[1], just a few English tokens comprise one segment in an otherwise monolingual Gaelic sequence. (2) sees far more English spans embedded within Hindi speech but, mirroring (1), there seems an asymmetric relationship between Hindi and English regarding language role: each of the English segments is a Noun Phrase (NP) providing a specific technical concept. In contrast, (3)'s code-switching sees English and Xhosa spans far more variable in length, there are far more switch points between the two languages, and, syntactically and semantically, the English also seems far less constrained (c.f. entire clauses containing function and content words).

Consider what would be required for an ASR system to recognize each of these examples. For cases like (1), we could perhaps exploit the code-switching's predictability and constrained nature and simply make use of monolingual ASR systems (running concurrently, or else switching between systems at predicted points). However, (3)-style code-switching may warrant more sophisticated methods; see the work discussed throughout Parts II and III. As such, not acknowledging this variability in code-switching 'style' could easily lead to domain mismatch issues: if we were to train and tune an ASR system to any one of the data-sets above, given this variation, we could not know whether such system would generalize.

This motivates the following research questions. First, how can we formally characterize the code-switching style of different data-sets? That is, which features or properties (number of switches; language distribution etc) should we observe and use to define how code-switching differs from data-set to data-set? Knowing how similar or different one data-set's code-switching is from another would greatly aid downstream ASR system development: we can choose appropriate training and test sets to minimize domain mismatch; we can anticipate lower word error rate (WER) in cases where different styles are unavoidable; we can select from a suite of potential methodologies one that is most suited to the code-switching style at hand.

Attempting to characterize this code-switching variation also brings to the fore a second research question: what causes this variation? Are there tangible variables which cause speech in different speech-settings (i.e. in different data-sets) to possess specific code-switching styles? We could consider for instance the speech's

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

domain; speaker characteristics like age or speaker-relationships; the languages in question (and the typological and socio-political relationship between them). In short, can we predict the style of code-switching we are to expect from a data-set given the data's meta-properties, or knowledge of where the data comes from? This would be particularly useful for when we are tackling code-switching in low-resource scenarios; that is, in situations where representative data is limited and we cannot infer code-switching style from the data itself.

Part I is structured according to these two research questions. For the remainder of Chapter 1, we elaborate on our attempts to characterize code-switching behaviour. In Chapter 2 we discuss our preliminary investigations into predicting code-switching behaviour and follow with an overall discussion of our future work plans and conclusions.

1.2 Related Work

Several methods of defining a data-set's code-switching style have already been proposed in the literature. Similar to that here, motivation is often to define some 'code-switching typology', classify data-sets according to this typology, to then inform downstream system development ([1, 2, 3]). Here, we provide an overview of existing metrics developed for this purpose.

1.2.1 M-metric and I-metric

The amount of code-switching in the data seems an obvious first question to ask if wanting to compare and rank data-sets by their code-switching. [4]'s M-index - one of the earliest metrics to be proposed in this area – does this by quantifying a data-set's language ratio (considering how many tokens are used in each language):

$$\text{M-index} \equiv \frac{1 - \sum p_j^2}{(k-1) \cdot \sum p_j^2} \tag{1.1}$$

Where k is the total number of languages in the corpus, p_j is total number of words in language j over total number of words in the corpus, and j ranges over the languages present in the corpus. The more even this ratio, the closer the M-index is to 1, and thus the more code-switched the data-set.

As an alternative or supplementary method of quantifying amount of code-switching, [1, 5, 2] propose the I-index:

$$\text{I-index} \equiv \frac{1}{n-1} \sum_{1 \leq i < j < n} S(l_i, l_j), \tag{1.2}$$

As with M-index, we assume a corpus composed of tokens tagged by language: here, $\{l_i\}$ denotes a tag, with i ranging from 1 to n, the size of the corpus, and j ranging from $i+1$ to n. $S(l_i, l_j) = 1$, denoting a code-switch, if $l_i \neq l_j$; 0 otherwise.

M- and I-index's differences are best illustrated via example. Consider the following example:

(1) 'it's not the neighbor in the corner . **yo creo que es el de al lado** [2]

Seven English and eight Spanish tokens (when split by white-space, see Section 1.3.2) lead to a very high M-index whilst, with just one switch point, this sentence's I-index is low. Whilst both metrics quantify amount of code-switching in a data-set then, observe that very different conclusions can be drawn depending on which metric is chosen as proxy for 'amount'.

1.2.2 Burstiness and Memory

By considering language tags irrespective of their position or context, neither M- nor I-index consider how languages' distribution across different code-switching data-sets may vary. With a 'language span' being a sequence of tokens with the same language tag, Burstiness ([3]), bounded between -1 and 1, thus considers how much the language span distribution differs from the Poisson distribution shown here:

$$\text{Burstiness} \equiv \frac{(\sigma_\tau / m_\tau - 1)}{(\sigma_\tau / m_\tau + 1)} = \frac{(\sigma_\tau - m_\tau)}{(\sigma_\tau + m_\tau)} \tag{1.3}$$

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).

Finally, Memory ([3]), also bounded [-1,1], considers the correlation between lengths of spans in one language versus another:

[2] `http://siarad.org.uk/speakers.php?c=miami`

$$\text{Memory} \equiv \frac{1}{n_r - 1} \sum_{i=1}^{n_r - 1} \frac{(\tau_i - m_1)(\tau_{i+1} - m_2)}{\sigma_1 \sigma_2} \tag{1.4}$$

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 (c.f. example (1) where long Gaelic segments contrast with short English segments). Comparatively, values close to 1 suggest spans much more even in length (c.f. example (3)).

1.3 Methods

Our aim for the remainder of this chapter is to assess these metrics' utility for the task at hand: sufficiently characterizing the different code-switching styles one might come across in different data-sets. Specifically, we assess these metrics' descriptive and characterizing power in terms of the following:

- Are any of these metrics sufficient at characterizing differences in code-switching when used in isolation?
- If not, are they sufficient when combined?
- If not, what code-switching style differences do they fail to capture which would thus need incorporating into future metrics?

1.3.1 Data Collation

To answer these questions, we collated a range of data-sets exhibiting code-switching, including many which have not been widely used in the code-switching literature. To note: throughout this chapter's remainder we make the simplifying assumption that code-switching occurs between two languages only. Thus, each data-set discussed has an associated language pair; all of the data-set's tokens must come from one of these languages. Importantly, we ensured we compare corpora which differ both in domain - see Tables 1.1 and 1.2 - and in language pair - see Table 1.3 - to ensure comprehensive coverage of 'possible' code-switching styles.

Data-Set	Language Pair	Domain
Handwriting ref	Scottish-Gaelic - English	Traditional Narrative
TaD [3]	Scottish-Gaelic - English	Traditional Narrative
MG Alba [4]	Scottish-Gaelic - English	Media Broadcast
audiobooks ref	Scottish-Gaelic - English	Read Speech

Table 1.1: Four Scottish-Gaelic-English data-sets from different domains, which exhibit code-switching to varying degrees. Note both Handwriting and TaD corpora are comprised of traditional narrative recordings held by the School of Scottish Studies Archives, University of Edinburgh. The former's transcription was produced via digitization of manuscripts during the Scottish Gaelic Automatic Handwriting Recognition Project; the latter's was provided by Tobar An Dualchais. The MG Alba data-set consists of a compilation of short videos (snippets of radio and tv broadcasts) from Learn-Gaelic, an online language learning resource created by MG-Alba, the official Gaelic Media service.

Data-Set	Language Pair	Domain
SEAME [6]	Mandarin - English	Informal Conversation
Technical Lectures [5]	Mandarin - English	Technical Lecture

Table 1.2: Two Mandarin-English data-sets from different domains, which exhibit code-switching to varying degrees. The SEAME corpus is comprised of informal conversations/interviews between college students. The data-set we name Technical Lectures is self-collected, made up of a Youtube playlist of uploaded lectures/educational content about machine learning.

Data-Set	Language Pair	Domain
SEAME [6]	Mandarin - English	Informal Conversation
Bangor-Miami [6]	Spanish - English	Informal Conversation
ArzEn [7]	Egyptian-Arabic - English	Informal Conversation
Soap Opera [8]	isiXhosa - English	Scripted Informal Conversation (Soap Opera)
Soap Opera	isiZulu - English	Scripted Informal Conversation (Soap Opera)
Soap Opera	seSotho - English	Scripted Informal Conversation (Soap Opera)
Soap Opera	seTswana - English	Scripted Informal Conversation (Soap Opera)

Table 1.3: Code-switching data-sets which are all (somewhat) conversational in domain, varying in language. Like the SEAME corpus, the Bangor-Miami and ArzEn corpora consist of conversations (or informal interviews conducted) between bilingual speakers; the former Spanish - English speakers in Florida; the latter Egyptian Arabic - English speakers in Egypt. Note the slight discrepancy in that the speech in the Soap Opera data-sets, covering switches between several South African languages and English, is scripted (with this data coming from broadcast South African soap operas).

1.3.2 Data Preparation

We pre-processed all data by removing punctuation, non-alphabetic strings, and lower-casing. The metrics introduced in Section 1.2 all treat data-sets as 'bags of language-ids'; each data-set had to therefore be tokenized and each token had to be assigned a language tag. Considering tokenization first, we utilized a baseline approach and split by white-space for most of the data.

We found we could not extend this uniform approach to the Mandarin-English corpora in Table 1.2 however for, whilst not true of the SEAME data, the Technical Lectures transcripts omitted white-space completely (i.e. did not use it to mark word boundaries). This orthographic variation between the two Mandarin-English corpora can be understood if we consider that Written Chinese is a logograhic rather than alphabetic writing system. Rather than characters mapping to phonemes (or some other phonetic unit), each Written Chinese character roughly corresponds to (the semantic content of) a syllable. With Classical Chinese a largely monosyllabic language, each Written Chinese character traditionally represented its own unique word, hence making white-space redundant. Importantly, though polysyllabic words are more common in Modern Chinese (represented by sequences of logograms when writing), utilization of white-space during writing is still variable (when not used, a reader must infer word boundaries of polysyllabic words from the context).

Whilst easy to trace and understand the origins of this variation then, this Mandarin-English example nevertheless raises questions. Practically, how do we go about using Section 1.2's metrics to compare data-sets in situations such as this one where what constitutes a token is not clear-cut? Secondly, and more importantly, are there any theoretical consequences of utilizing metrics which rely on the notion of the token, when the concept of 'token' is inherently fuzzy in nature?

Taking a practical perspective first, Figure 1.1 illustrates the metric scores we obtained when we tokenized the SEAME data-set using the baseline white-space method (recall that the SEAME data did contain white-space) versus tokenizing at a character level (the only method by which we can tokenize the Technical Lectures corpus). Whilst not extreme in this particular case, that these two sets of scores do differ highlights these metrics' sensitivity to token definition. As such, for an experimentally valid comparison to be made between data-sets (of the same language), tokenization method should really be consistent. For this chapter's remainder, we thus assume tokenization at character level for both Mandarin-English data-sets.

While relevant here, we retain discussion of how the fuzzy nature of the 'token' interacts with comparisons between data-sets of **different** languages to Section 1.4.2. We believe a shortcoming of previous works utilizing these metrics ([1, 2, 3]) are their claims that these methods are 'language independent': we shall argue this is not the case – in part, because of the way differing linguistic typologies impact upon token distribution – and briefly consider how these metrics could be amended to account for this.

Setting aside for the moment these difficulties associated with the token, we turn to the task of assigning these tokens language tags. Some data-sets (those in Table 1.2 and the South-African - English data-sets in Table 1.3) came with language already annotated: that is, the transcripts were marked up with language switch-points. For the remaining data-sets listed above however, we had to apply some external, language identification (LID) method.

For instance, to language tag the Scottish-Gaelic – English data-sets of Table 1.1, we initially employed the Naïve Bayesian classifier compact language detector (CLD2)[7], which uses character n-grams as features. However, initial experimentation showed error rates to be high: somewhat expected as we are identifying the language of stand-alone tokens here rather than the language of a whole passage (within which there would be

[7] `https://CRAN.R-project.org/package=cld2`

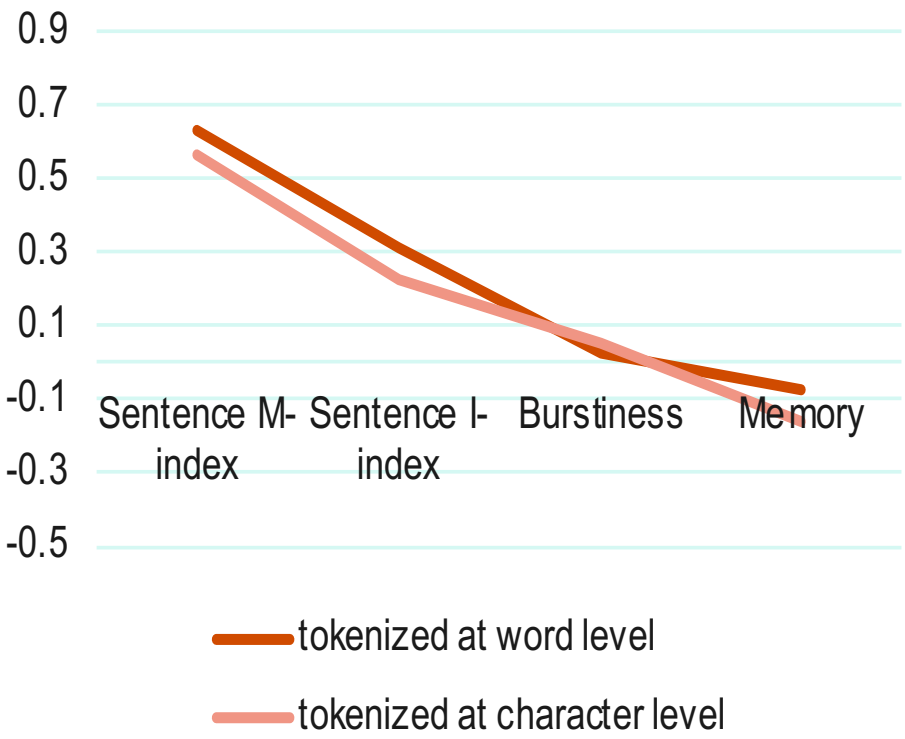


Figure 1.1: Code-switching metrics' sensitivity to word versus character level tokenization for the SEAME Mandarin-English data-set.

more opportunities for distinctive n-grams). We thus supplemented CLD2 with hand-crafted rules: for instance, if 'English' was predicted for a token comprising of three characters or fewer, we would check whether either adjacent token had also been tagged 'English'; if not, we would replace the predicted 'English' tag with that assigned to the previous token.

Still, depending on upstream, external LID will always result in at least some incorrectly tagged tokens. Given the metrics we are investigating only consider token language, rather than token identity – and are thus particularly sensitive to such error propagation – how best to integrate language identification is an important question to explore going forward. Also worth acknowledging here are the comments made in [9]: even for those data-sets where 'gold-standard' language tags are provided , we must be wary of annotation scheme variation between data-sets (e.g. can tokens be assigned labels other than the two main languages in question – 'unk', 'unkown-word' etc) leading to in-comparable metric scores (much like the consequences of inconsistent tokenization methods).

Also of relevance here is that a few of our code-switching data-sets exemplified 'intra-word' code-switches. Consider for instance 'ndibalance', 'uadvertise', 'ipost'. All three examples are drawn from the Xhosa-English Soap Opera data-set [8] and consist of an isiXhosa prefix prepended to an English verb or noun. As mentioned, since we tokenized by white-space, we could not language-tag these Xhosa prefixes and English stems distinctly; thus, during this work, we simply tagged such words as Xhosa instances.

Knowing how best to handle these intra-word cases is not trivial. Had we incorporated some extra morphological segmentation when tokenizing the isiXhosa-English Soap Opera data, we could have carried out language-tagging at a finer-grained level. But with this, we would be forced to return to the questions posed above: what should constitute a token in the first place; can we define a token in a language-independent fashion? Allowing for the possibility of an extra morphological analysis step prior to language identification would make the task of keeping the 'token' consistent across data-sets even more difficult: the quality, methods, and availability of morphological parsers vary extensively across languages (and c.f. also the opportunity for further error propagation).

Section 1.4.2 picks up this discussion of these metrics' potential pitfalls and complexities. Setting aside these questions for the time-being however, having obtained LID sequences for each data-set, we were able to calculate M-index, I-index, Burstiness and Memory per corpus as detailed in Section 1.2. In the following section, we compare data-sets according to these metrics and consider their adequacy as a comparative tool.

1.4 Results and Discussion

1.4.1 What is being captured?

We first (and primarily, given the following section's discussion) consider these metrics' utility for comparing data-sets exemplifying code-switching between the *same* two languages: in our work, this means considering Table 1.1's Scottish-Gaelic English data-sets and Table 1.2's Mandarin-English data-sets. Do these metrics facilitate an investigation into what happens to code-switching style when language remains constant but when the domain and/or speech-setting of a data-set changes? Does code-switching nevertheless remain consistent?

Consider the following scenario. We want to build a Scottish-Gaelic ASR system. We know some code-switching (into English) is likely to be present at test time. However, for us to better understand what type of code-switching ASR methods should be utilized, we would also like to know if there is any one particular style of code-switching inherent to Scottish-Gaelic - English speech; and if so, what properties such style might possess.

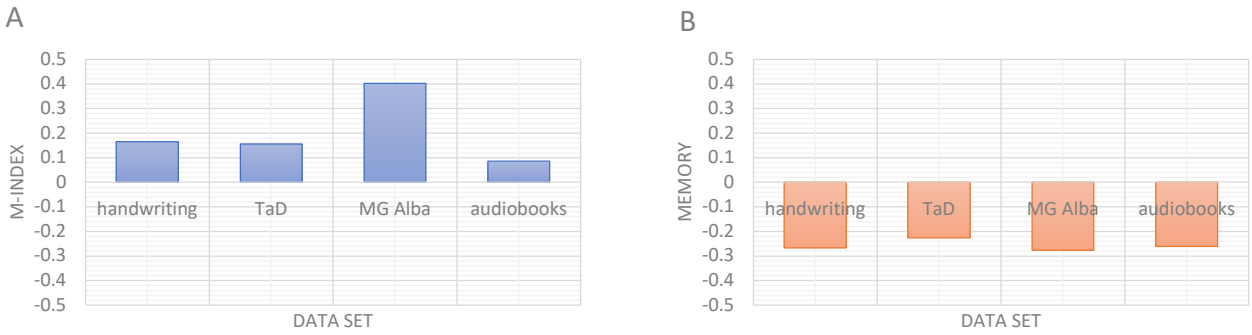


Figure 1.2: Measuring the M-index (A) and Memory (B) of different Scottish-Gaelic - English data-sets

Figure 1.2a compares the M-index of our four Scottish-Gaelic - English data-sets. Immediately evident is that, by this proxy at least, code-switching style is *not* necessarily consistent across corpora just because they contain the same languages. MG-Alba's far higher M-index suggests that this corpus of speech drawn from *media broadcasts* contains *significantly* more code-switching than the other three (which, recall, are all non-conversational in nature).

However, Figure 1.2b tells a different story. When using Memory as our code-switching richness measure, there is no stark difference between MG-Alba and the other corpora (in fact MG-Alba's Memory score is comparatively low). Hence, *this* metric points to a similar code-switching style across all four Scottish-Gaelic – English data-sets: one which is fairly constrained, regular and predictable (more akin to example (1) than (3)).

That these two metrics lead to opposing conclusions immediately highlights the dangers of using these metrics in isolation. Recall from Section 1.2 that Burstiness and Memory consider language distribution; M- and I-indexes comparatively measure quantity (and do so using very different definitions of quantity, see Example (1)). Code-switching variation is clearly extensive enough that we need (at least) these two dimensions to capture stylistic variation between data-sets.

Considering these M-index *and* Memory comparisons *together* then, we can return to the hypothetical scenario posited above – determining how best to build a Scottish Gaelic ASR system which can also cope with code-switching into English. We now have some evidence (albeit having only considered four corpora) for the notion that, in terms of language distribution, Scottish-Gaelic - English code-switching is fairly invariant across different domains and speech settings, regardless of absolute English quantity. This means, during the ASR development process, we can be more confident that we will not run into large domain-mismatch issues when selecting pairs of train/test data-sets; or run into problems if utilizing the same code-switching ASR method uniformly (in this case, one which is fairly simple, e.g. relying on monolingual systems). This comparison of Scottish Gaelic-English corpora has demonstrated that there is information to be gained from these existing metrics which can enrich the ASR development process, as long as we consider them in combination. We thus now apply a similar analysis to our Mandarin-English code-switching data-sets. This time, Figure 1.3, comparing the SEAME and Technical Lectures corpora, immediately considers all four metrics together rather than individually.

What is striking in Figure 1.3 is that, though some individual measures do not vary much (e.g. I-index), these two plots' overall 'shape' is markedly different; as with the Scottish-Gaelic - English case then, clear is that just looking at any metric individually is insufficient. Consider in particular the I-index : Memory ratio for the two data-sets. 1.3's Technical Lecture series' lower Memory score results in a steeper negative gradient on the graph's right hand side; SEAME's line is comparatively much flatter. This suggests that in

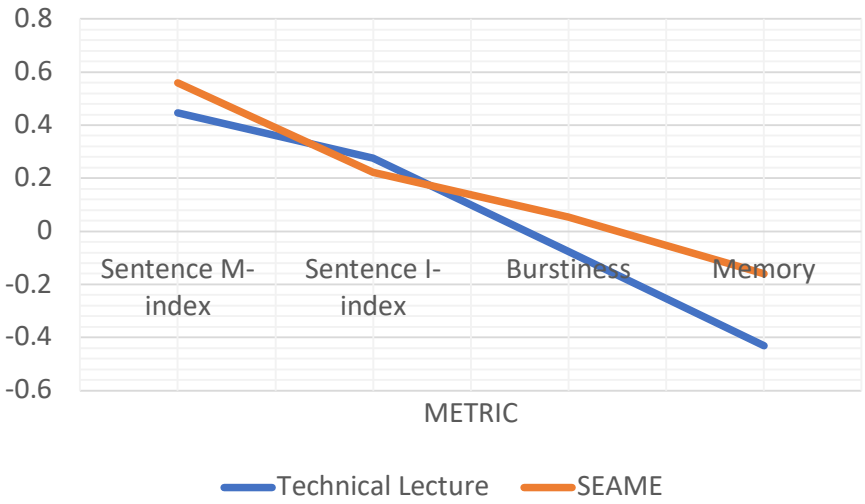


Figure 1.3: Extracting code-switching style metrics for two Mandarin-English data-sets: Technical Lectures (blue) and SEAME (orange)

the Technical Lectures data, English is being used for a more constrained or specific purpose (like supplying technical concepts), resulting in consistent cycles of long Mandarin- and short English spans, similarly to the Scottish-Gaelic - English data above. Comparatively SEAME's English may not be so constrained.

It is interesting that our observation for Scottish-Gaelic - English – that code-switching distribution seemed unaffected by domain – does not hold true for our Mandarin-English data. To explore in the future is whether there are any particular properties of these two language pairs (linguistic, socio-historic, cultural, geographic etc) which lead to this difference.

We can manually examine both Mandarin-English data-sets to confirm these stylistic distinctions (with example (2) from Technical Lectures; (3) from SEAME):

(2) "所以出了框架的差距就超 ε 所以今天呢你在做完Gradient Descent... " 所以你可以Simple Baseline 那有同就一必有什好呢"

(3) 我我喜欢人家to cheer me on 我觉得it would help to ... ... 然后然后有很多字right that we commonly use but they don't use

1.4.2 What is not being captured?

Language-Independence and Normalization

Whilst the previous section illustrated that these four metrics do successfully capture some basic code-switching style differences – and thus facilitate some preliminary code-switching variation analysis – there are also clear limitations. In Section 1.3.2, we began to question the reliance on the 'token' in these metrics' formulation. We explore this further here; in particular, what are the implications of this when using these metrics to compare data-sets of different languages? Consider the following example, taken in part from [10]:

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

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

(6) **English Translation** **and I like them** *M-Index* = 0.0

Because languages differ in their morpho-syntax and orthographic conventions, they also fundamentally differ in how tokens tend to be distributed across sentences and speech segments. Here, though both South African languages, Sotho uses a 'disjunctive' orthography; Zulu, a 'conjunctive' (Xhosa, sharing a language family with Zulu, is also conjunctive). The equivalent semantic and syntactic content is thus distributed over far more tokens in the former than the latter. And this has heavy implications on our code-switching metric computation. A Sesotho-English corpus will always have a higher Sesotho : English token ratio than a Xhosa-English corpus' Xhosa : English token ratio. Similarly, Sesotho spans will always be longer than Xhosa spans. And therefore, a Sesotho-English data-set will always receive a lower M-index score; a lower Memory score etc.

In short, despite claims to the contrary ([2, 3]), we must be aware that these metrics are not strictly language independent but bake in language typological differences. If being employed to compare data-sets across different languages then and, specifically, if we want to use these measures to investigate whether other variables such as

domain, speaker characteristics etc predict certain code-switching styles (see Chapter 2), then such differences should really be factored out. Note that other metrics proposed more recently – c.f. [11]'s SyMCoM which considers POS tags – would also require such normalization or calibration if to be used in a language-independent fashion. We leave a more robust/in-depth investigation into approaches to metric calibration/normalization for future work.

Are these metrics sufficient?

As mentioned in Section 1.3, to fully evaluate these metrics' descriptive power, we must also consider if there are any code-switching properties these measures fail to capture. Compare the following Hindi-English (7) and Spanish-English (8) segment:

(7) **crystal systems crystal structures cubic hexagonal rhombohedral** [12]
'and showing the **crystal structures** of different **crystal systems** for example **cubic, hexagonal** and **rhombohedral**'

(8) **and then his other thing he noted when he was at the UOne siding and connector road** (.) que pusimos luces . **you know the new road we put in** . [8]
'**and then his other thing he noted that when he was at the UOne siding and connector road** that we put lights on **you know the new road we put in** . '

If basing comparison on the four metrics used thus far, the Hindi-English's code-switching would come out the richer of the two: the number of English and Hindi tokens is more even; there is more frequent switching; spans are more equal in length. However, what is not captured using these metrics is any sense of the languages' syntactic or semantic role. Referenced during the introduction, in the Hindi-English snippet, English solely provides NPs: it is being used to talk of concepts for which Hindi has no terminology. That this is not true of the Spanish-English's code-switching means that it is perhaps this snippet which should be considered the 'richer' or 'harder' code-switching here (at least regarding unpredictability and this effect on downstream modelling). In short, these metrics appear too coarse. The literature tells us we should not be agnostic to syntactic or semantic ([13, 14, 15]) or phonological features; there is thus a lot of scope to develop more sophisticated measures which *do* capture such additional stylistic variations. We aim to return to this in future work.

1.5 Future Work and Conclusions

In this chapter, we illustrated that code-switching is a variable phenomenon and proposed that, to avoid domain mismatch, we must: 1) understand how to characterize code-switching behaviour across different data-sets and 2) identify what variables can cause or predict such differences in behaviour. Regarding the former, we found existing metrics able to characterize some stylistic differences when combined, licensing their use as a preliminary tool for data-set selection and analysis during ASR development. However, the assumptions that these metrics make regarding language typology must be considered when using them to compare across languages. We also showed that there are some stylistic differences in data-sets' code-switching which these metrics do not encode; as such, there is scope to develop more sophisticated metrics which consider the languages' semantic, syntactic or phonological properties. We aim to tackle this as part of our future work in the area.

In Chapter 2, we turn our focus to the task of predicting code-switching behaviour. We build upon the findings of this chapter, using the metrics introduced here as a key methodological tool. We also end Chapter 2 with an overall discussion of Part I and a forward look to how our work could potentially begin to be utilized to inform decision-making during ASR system development.

[8] `http://siarad.org.uk/speakers.php?c=miami`

Chapter 2

Predicting Code-Switching Behaviour

Debasmita Bhattacharya, Danielle Cartagenes, Electra Wallington, Peter Bell

2.1 Introduction and Related Work

The previous chapter introduced four existing metrics designed to characterize the code-switching behaviour of different data-sets. We demonstrated that, when combined, M-index, I-index, Burstiness, and Memory can successfully capture some key stylistic differences between corpora (though there are also many differences which they are too coarse to pick up on). Using these metrics as our preliminary analytical tools, we turn now to the question: can we identify potential causes of, and thus predict, this code-switching variation?

The question of what kind of 'higher-level' variables may influence code-switching style has been the subject of much previous work. Links have been found between speaker characteristics and particular code-switching properties. [16] for instance found the main factors affecting the style of Spanish-English code-switching in New York to be the speaker's sex, the age the speaker acquired their second language (L2), which language was the speaker's dominant, and the speaker's workplace. [17] also showed age and method of L2 acquisition to be key predicting factors of Welsh-English code-switching.

We could additionally consider the effect of the languages themselves and the relationship between them, both from a socio-historical and linguistic point of view [1]. [18] for instance found a language's status (relative to the other) to affect how it is used and distributed during code-switching. [19] showed that the style of code-switching consisting of the same two languages could vary across different geographical regions due to distinct social contexts: c.f. the 'Spanglishes' of Miami, El Paso, Los Angeles, and New York. There also exists work exploring how the domain or setting of a particular speech situation influences its code-switching characteristics and, related to this, how the formality of a situation (which is typically dictated by the speech setting) influences its code-switching.

We focus here on a couple of these 'higher-level' variables and thus structure this chapter as follows. We begin with a brief examination of the influence a speakers' characteristics has on code-switching behaviour: namely, age and gender. The experiments we present in this section are not intended to be 'complete' (in the sense of being statistically significant or entirely robust); rather, they serve as a demonstration of the kind of baseline experiments one could run using the metrics introduced in the prior chapter. We then turn to examining the role the 'formality' of a data-set has on code-switching behaviour; here, we provide a much more in-depth investigation.

2.2 Age and Gender

We introduced above that there is considered a relationship between code-switching and age: [20] for instance find age to be a key predictor of code-switch style; [17] find a negative correlation between age and code-switching quantity, positing this as an indicator for code-switching becoming more common and socially accepted. The Bangor-Miami corpus – referenced in Chapter 1 – is comprised of a number of informal conversations between Spanish-English bilinguals living in Florida. Unlike many of the other data-sets established in the previous chapter, this corpus is marked-up for age; we thus use this data-set, along with the metrics discussed throughout Chapter 1, to investigate age's predictive power.

Figure 2.1 presents our analysis. We see that M- and I- index scores seem to slightly decline as age increases; tentatively then (considering the extremely shallow trend lines), the older one is, the less likely they are to code-switch (if these metrics are a proxy for 'amount' or 'richness'). Note that this trend also holds true for Memory and Burstiness metrics, omitted here for lack of space.

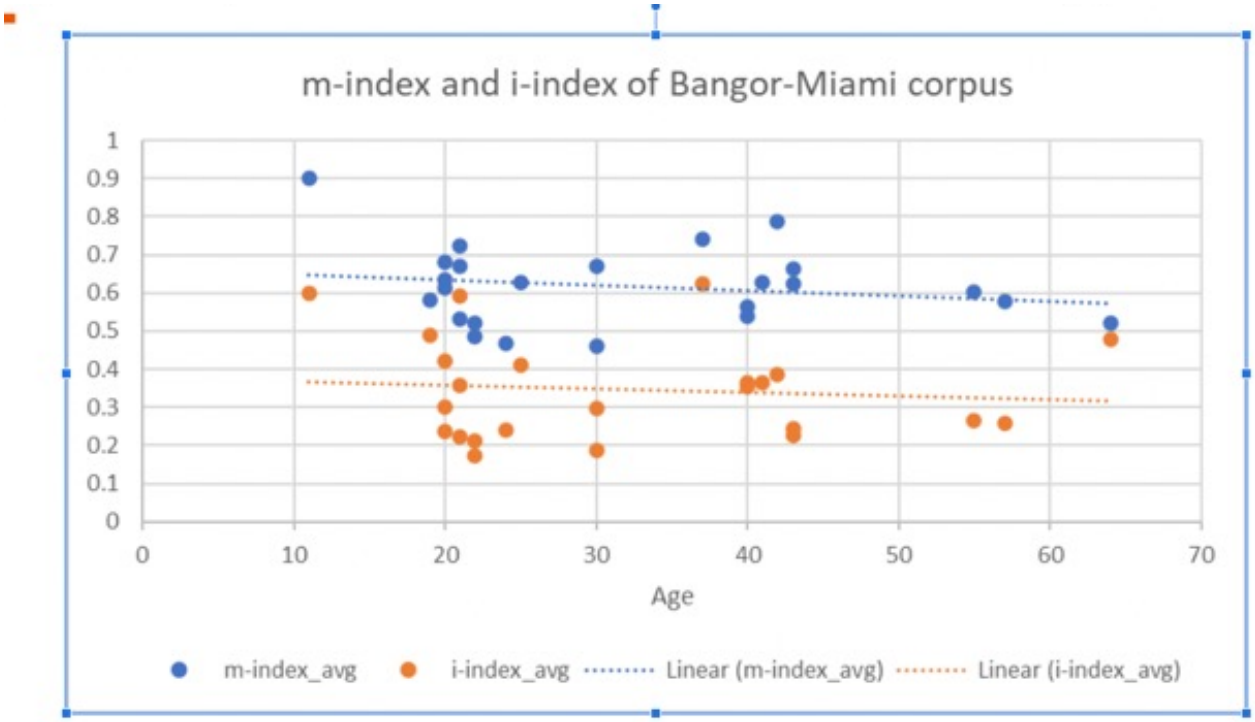


Figure 2.1: Investigating how code-switching metrics vary with age in the Bangor-Miami data-set

Gender has also been shown to affect code-switching behaviour; see [20] or [21]. Additionally to age, the Bangor-Miami corpus contains speaker gender information, as does the Mandarin-English SEAME corpus used throughout Chapter 1. We thus ran baseline experiments on both these data-sets.

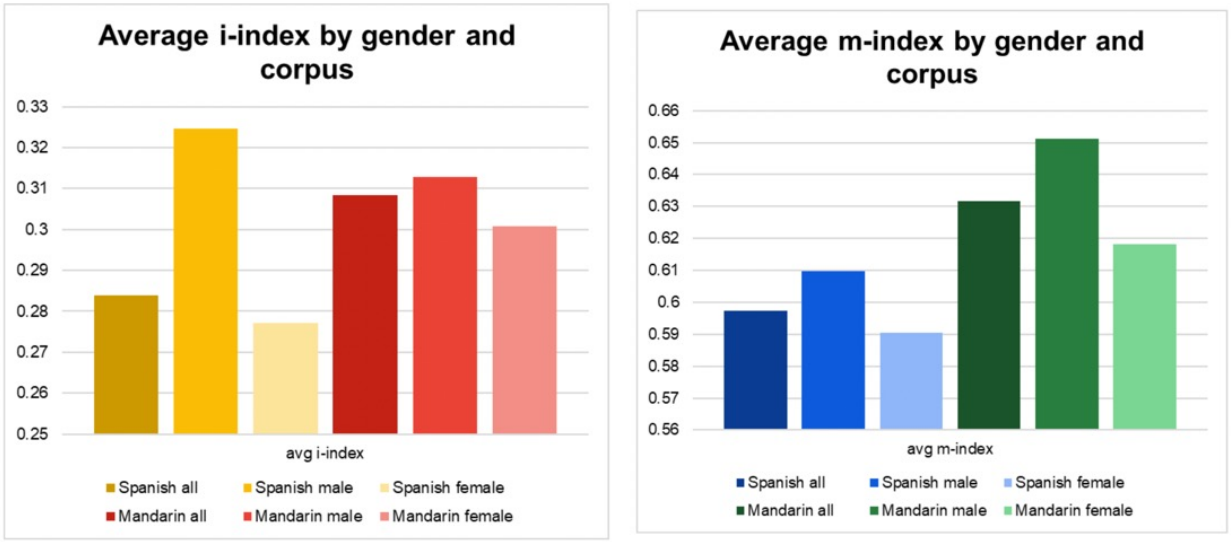


Figure 2.2: Investigating how code-switching metrics vary with gender in the Bangor-Miami and SEAME data-sets

Again utilizing M- and I-index, Figure 2.2 shows us that men seemingly code-switch more than women in both corpora. As with our exploration of age, we acknowledge this is incredibly tentative. We also cannot know if these conclusions would generalise, though it it is interesting that the same trend *was* found across both data-sets here. We thus hope to explore the effects of gender (and age) in more depth and across larger quantities of data during future work, such that we can obtain more valid and reliable results (recall this section's primary purpose was to illustrate the type of experiment and analysis that can be done given the data and metrics we now have available to us).

As part of this illustration, it is also necessary to point out here the relevance of Chapter 1's discussion; there are certain stylistic properties of code-switching behaviour that these metrics are too coarse to capture.

When running experiments akin to the above we must be aware of this; it could be the case that, were we to use alternative, more sophisticated metrics to define code-switching style, we would find larger, the effects of gender or age (or other higher-level factors) on code-switching style may be far more (or less) apparent.

2.3 Formality

We turn now to a more focused investigation: what role does formality play in code-switching (note that time constraints prevented us from applying such analysis to any more of the variables introduced in Section 2.1)? And are there universal patterns in formality in code-switched data sets involving different language pairs? In order to answer these questions, we conduct a study into text- and speech-based formality of three multilingual code-switched corpora. Our ultimate goal is to extrapolate our findings to predict the amount of code-switching in an unseen corpus or domain from its level of formality. Domain-level formality in particular would be useful for understanding and anticipating possible formality effects on code-switching in a new domain ahead of any experimentation, particularly for low-resourced languages or language pairs without much data, which is often the case in studies of code-switching.

There are a few reasons why formality is a relevant higher level factor to study in relation to code-switching. Firstly, formality is a characteristic of language production that is universal, so any work done on formality can, in theory, be extended to new language pairs, speech settings, and data sets. In addition, whenever speakers interact with one another, they adopt a specific register or level of formality. Thus, domains can be treated as having an inherent and stable level of formality to an extent, for instance, speech from broadcast news in any language tends to be quite formal whereas telephone conversations between friends tend to be informal.

In addition to this, when it comes to code-switching, we have some intuition about: the kinds of situations in which it is more or less acceptable to code-switch, how in different situations we are likely to code-switch in different ways, and how formality might inform the kind of code-switching we perform. In this way, a focus on formality draws indirectly on speech setting and speaker relationships, which are also possible factors that influence code-switching as a whole.

It is worth noting that formality itself is quite a big topic that has not been covered much in the computational literature, in relation to code-switching or otherwise, making it ideal for a preliminary investigation.

2.3.1 Previous Work

The key question at the heart of our investigation is how to define and measure formality of text or speech. Among the limited existing studies on formality in either medium, there has been slightly more work done on formality of text, in English. According to [22], formality is defined as "avoidance of ambiguity by minimising the context-dependence and fuzziness of expressions. This is achieved by explicit and precise description of the elements of the context needed to disambiguate the expression. A formal style is characterised by detachment, accuracy, rigidity, and heaviness; an informal style is more flexible, direct, implicit, and involved, but less informative". In other words, formal language avoids ambiguity by including the information about the context that would disambiguate the expression into the expression itself. Formal language explicitly states the necessary references, assumptions, and background knowledge which would have remained tacit in an informal expression of the same meaning ([22]). As a result, formal language is more context-independent and structurally complex than informal language. Note that we keep in line with Heylighen and Dewaele and treat formality as a relational attribute of language: an expression can be more or less formal compared to another expression, but no expression can be absolutely formal or absolutely informal.

An empirical measure of these features of formality, the F-score[1], is also provided by Heylighen and Dewaele in their 1999 paper and is the metric that we use throughout our investigation for text. The F-score metric is based on the distribution of part of speech categories within a corpus or conversation, as certain parts of speech lend themselves to more precise and unambiguous language, whereas others are needed to produce more implicit and involved language. The former class of parts of speech includes nouns, adjectives, prepositions, and articles, while the latter includes verbs, adverbs, pronouns, and interjections. This leads us to the following simple formula for calculating the F-score on a textual corpus:

F-score = (noun frequency + adjective freq. + preposition freq. + article freq. – pronoun freq. – verb freq. – adverb freq. – interjection freq. + 100)/2

The frequencies in the above formula refer to percentages of the number of words belonging to a particular part of speech category with respect to the total number of words in the corpus under investigation. The F-score can then vary between 0% and 100%, with more formal corpora getting higher F-score percentages. Note that it is not meaningful for a corpus to have an F-score or exactly 0% or 100%.

[1] F-score here is referring to the level of formality, a higher score means more formal.

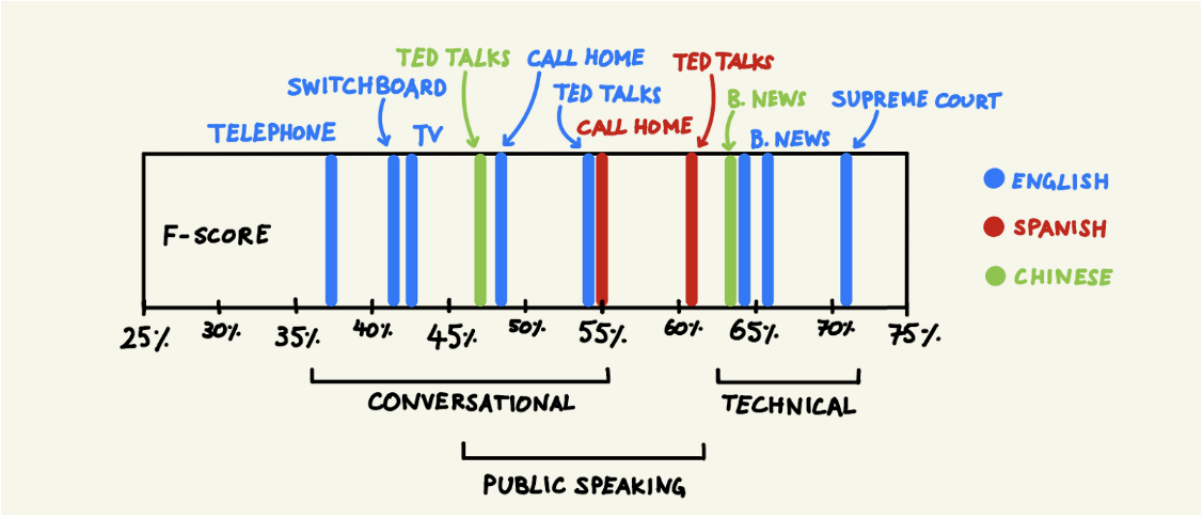


Figure 2.3: Calibrating F-score on monolingual corpora of known formality

Heylighen and Dewaele showed that their metric of formality is both applicable to and reliable in languages other than English, including French, Italian, and Dutch. We aim to extend this text-based metric to Mandarin, Spanish, and Scottish Gaelic in this work.

In addition to formality of text, there have also been a couple of previous studies on measuring the formality of speech. [23] investigated the phonetic profile of formal and informal Korean speech registers, and [24] examined acoustic properties of formality in conversational Japanese. These authors collectively found that acoustic features including speaking rate, pitch, and loudness were indicative of different levels of spoken formality. The common theme between these papers is the *variation* in the features considered, rather than their absolute values, where greater variation in any one feature is linked to less formal conversations. In this work, we follow up on two of these indicative features of speech formality: variation in speaking rate and variation in pitch throughout a conversation.

2.3.2 Corpora

As alluded to previously, our investigation involves explorations of corpora in Mandarin, Spanish, and Scottish Gaelic. The multilingual code-switched corpora we examine are the SEAME Chinese-English corpus, the Bangor Miami Spanish-English corpus, and the Clil Gaelic-English corpus respectively. These consist of interview-based and conversational data where speakers code-switch on occasion. We work most deeply on the first two of these corpora.

For the purposes of baselining, we also make use of a number of monolingual corpora. These include: The Switchboard Dialogue Act Corpus, English Broadcast News transcripts, English Television transcripts, an informal telephone conversation corpus provided by Peter Bell at the University of Edinburgh, proceedings of the US Supreme Court from the SigmaLaw Large Legal Text Corpus, CABank English CallHome Corpus, CABank Spanish CallHome Corpus, Multilingual TEDx talks in English and Spanish, The Multitarget TED Talks Task in English and Mandarin, and TDT4 Multilingual Broadcast News in English and Mandarin.

2.3.3 Investigating Formality in Text

Method

Throughout this work, we assume that a conversation or corpus has an established level of formality, which can be measured effectively on its monolingual segments. This established level of formality can then be used to analyse the formality of the code-switched segments of the corpus. We incorporated this assumption into our workflow for text-based formality. We first separated our data into monolingual and code-switched portions, then performed part of speech tagging on those subsets of the data. For Mandarin, Spanish, and English, we used an off the shelf library, spacy, to get part of speech tags for the SEAME and Bangor Miami corpora. For Scottish Gaelic in particular, we used a specialised tagger that was kindly made available to us by Will Lamb's Gaelic research group at the University of Edinburgh. Following tagging, we simply computed the F-score according to its formula.

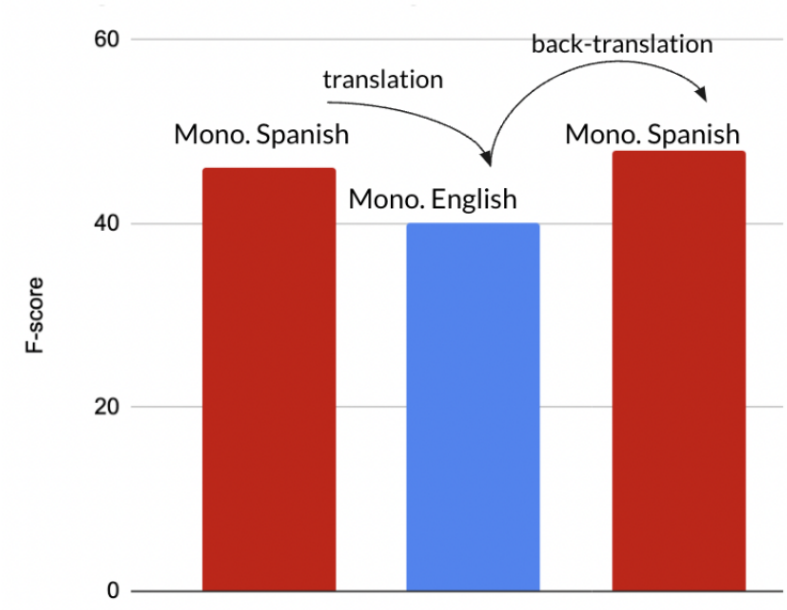


Figure 2.4: F-scores from translating monolingual portions of the Bangor Miami data from Spanish to English before back-translating to Spanish.

Experiments

We began with a number of baselining experiments, on the monolingual corpora previously mentioned, in order to calibrate the F-score metric on monolingual corpora of known formality.

As in Figure 2.3, we found that informal corpora, such as the Switchboard corpus, had F-scores of about 40% and that more formal corpora, such as broadcast news and US Supreme Court proceedings, had F-scores of about 70%. We also found that within the Switchboard corpus, random subsets of the corpus had consistently similar scores, and that within the Call Home corpus, both Spanish and English monolingual transcripts had similar scores, despite being in different languages. This was also true for Mandarin and English within a multilingual broadcast news corpus.

We also did baseline experiments on multilingual TED Talks transcripts, and found some interesting observations, which come up again in our experiments on the code-switched corpora in the following section. With the Spanish-English TED Talks, the talks were originally given in Spanish and later manually translated into English. In this case, while the F-score in either language is similar to the other, Spanish consistently has a higher score, which may have something to do with the syntax of the language compared to English. We observed a similar effect but in the opposite direction with the Mandarin-English TED Talks.

Overall, from our baseline experiments, we found that the F-score metric seems to work reasonably well across the different languages we are investigating and across different levels of formality. We estimated that the lower bound for informal scores is around 30% and the higher bound for formal scores is around 80%.

To complete our calibration of the F-score metric, we computed F-score on our chosen code-switching corpora [2]. Specifically, we wanted to see whether the F-score would be consistent before and after automatic translation of the monolingual segments of the data. We found some interesting patterns in what happens when we start with monolingual Spanish, translate to English, and then back to Spanish, and vice versa (see Figures 2.4 and 2.5). As was the case with our baselining experiments, identical English data has a lower F-score than its corresponding Spanish data, and back-translation to Spanish actually increases the F-score compared to the original value. This effect is visible in the opposite direction as well, with back-translation to English also increasing the F-score compared to the original value [3].

There are a couple of different effects at play here. One explanation for these observations is that the F-score has some inherent language-dependence due to the differences in the natural distributions of parts of speech in different languages with differing syntax. Spanish might use more articles to express the same meaning as in English, resulting in a higher F-score in Spanish than English, for instance. Another effect at play, however, does seem to be that of machine translation alone, where automatic back-translation seems to alter syntax to a significant enough level that then indirectly affects formality measurement via F-score.

[2]Note that in this section of the paper, we report results only for the Bangor-Miami Spanish-English corpus.

[3]Note that we observed a similar effect on the F-score when looking at the SEAME Mandarin-English data, where identical Mandarin data had lower scores compared to corresponding English data.

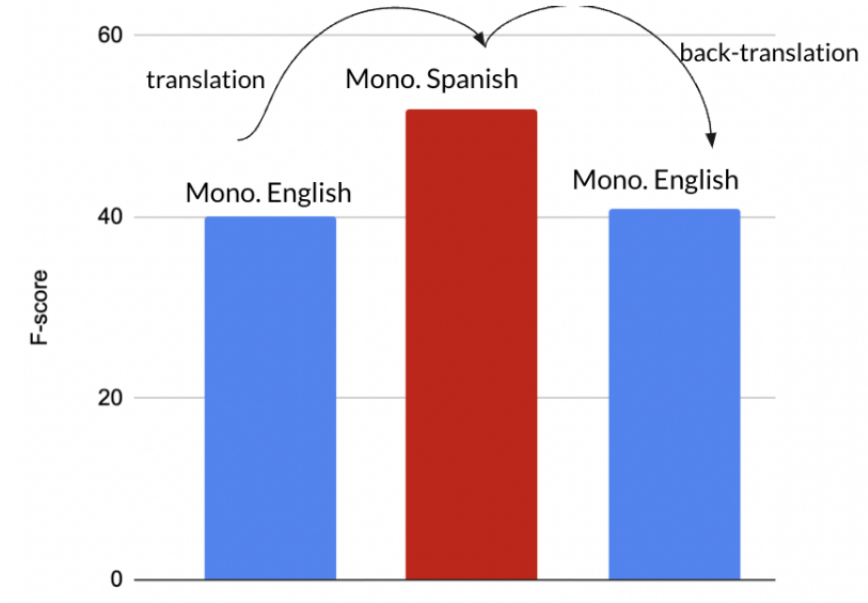


Figure 2.5: F-scores from translating monolingual portions of the Bangor Miami data from English to Spanish before back-translating to English.

Overall, in addition to seeing that F-score works reasonably well, we get a sense of how it may depend on certain language-specific properties. Furthermore, F-score should be normalized per language.

Having established that F-score works on monolingual data, we next tested and found that it also works on code-switched data across all three of our corpora, as all of the resulting F-scores in Figure 2.6 are on the informal side, matching our prior knowledge and expectations.

Upon translation of the code-switched data into fully Spanish sentences and fully English sentences (see Figure 2.7), we observed similar effects on the F-score as we did with monolingual data. Note that these translation results generalise to code-switched data in the SEAME and Clil corpora.

Next, we take a closer look at the distribution of part of speech categories across the code-switched sentences in the Bangor Miami corpus, since these are what feed directly into the formality scores. Interestingly, as in Figure 2.8, we get different proportions of the parts of speech when we translate the code-switched sentences into monolingual Spanish versus English, with more of the parts of speech that are associated with formality in Spanish. The reverse is true when we look at the English portions of the code-switched sentences compared to the Spanish portions of the same, which gives us some insight into which parts of speech are favoured for code-switching into either language.

Further interesting translation results are visible when considering the monolingual sentences, as we can see the effect that automatic translation has on the distribution of parts of speech, and thus on the formality metric. In Figure 2.9, translations to English seem to increase verbs and interjections, while translations to Spanish seem to increase nouns and articles and decrease pronouns. This has an effect on the formality scores and explains those results, since verbs, interjections, and pronouns are more informal parts of speech, while nouns and articles are more formal parts of speech.

After looking at the formality metric at the corpus level, we went one level of abstraction lower and considered the conversation level. In Figure 2.10, the fact that there is no correlation at the conversation level between the formality of English and Spanish is interesting, in the sense that it seems we can't think of a conversation as having an inherent formality level that is independent of the language being spoken. However, we thought that this could simply be due to statistical noise dominating the measure at the level of individual conversations, and a follow up ANOVA analysis at the sentence level showed that this was indeed the case, providing support for our initial assumption about fixed formality. We also found that F-score does not work well on shorter sections of text or at a very granular level.

Another clearly evident pattern in Figure 2.10 is that the variance in formality levels between conversations is quite markedly higher for Spanish than English. We are not sure exactly why this happens in this corpus, but it could be that the Spanish speech is reflecting the true formality level of the conversation, with English being employed by speakers for a limited range of topics, with less variation in formality. Further investigation involving topic analysis of code-switching and how it relates to formality is a good direction for future work.

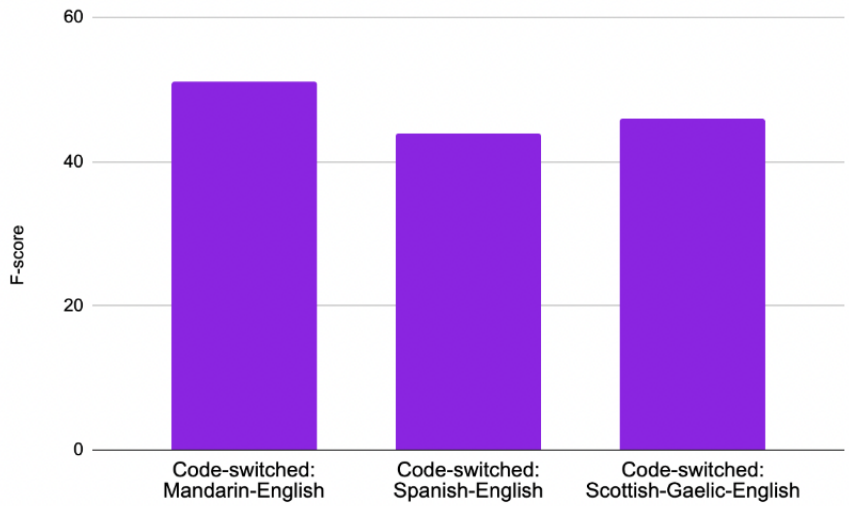


Figure 2.6: F-scores of SEAME, Bangor Miami, and Clil code-switching corpora

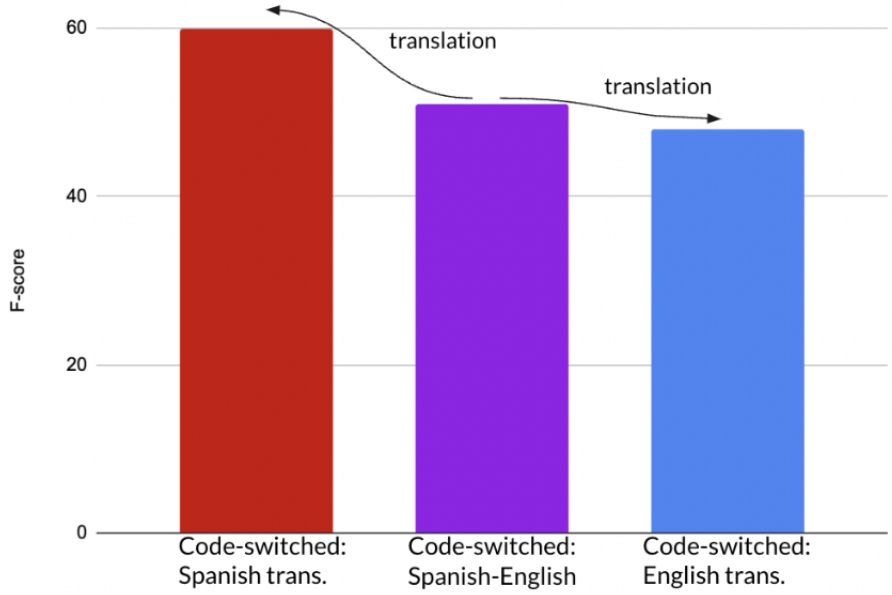


Figure 2.7: F-scores from translating code-switched Bangor Miami data into monolingual data

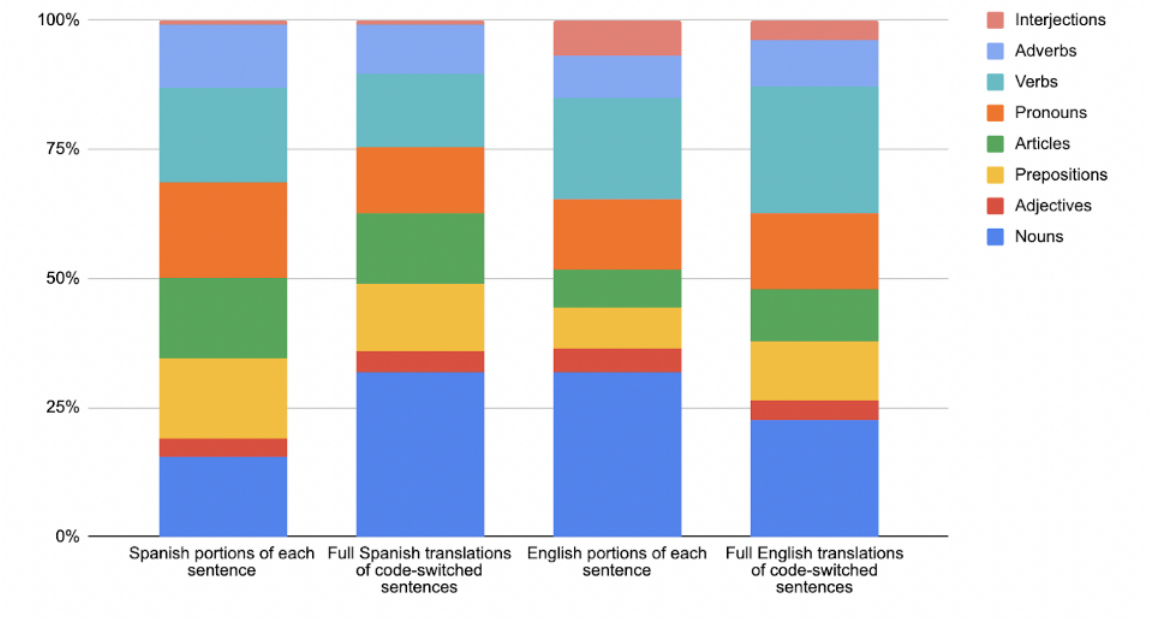


Figure 2.8: Part of speech distributions across code-switched sentences in the Bangor Miami corpus

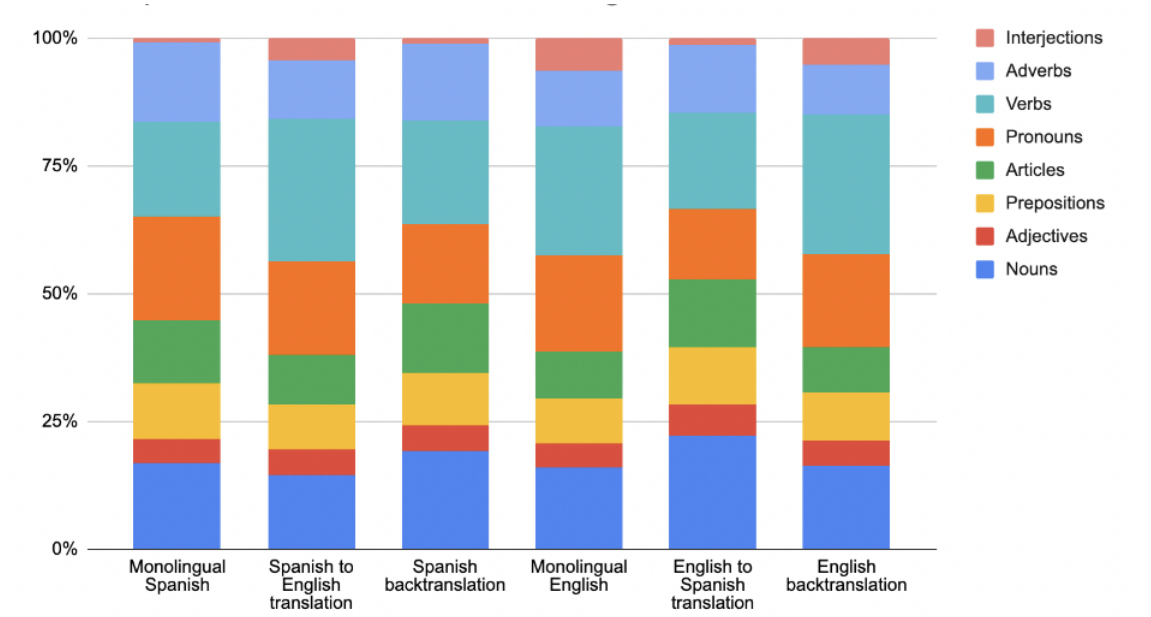


Figure 2.9: Part of speech distributions across monolingual sentences and their translations in the Bangor Miami corpus

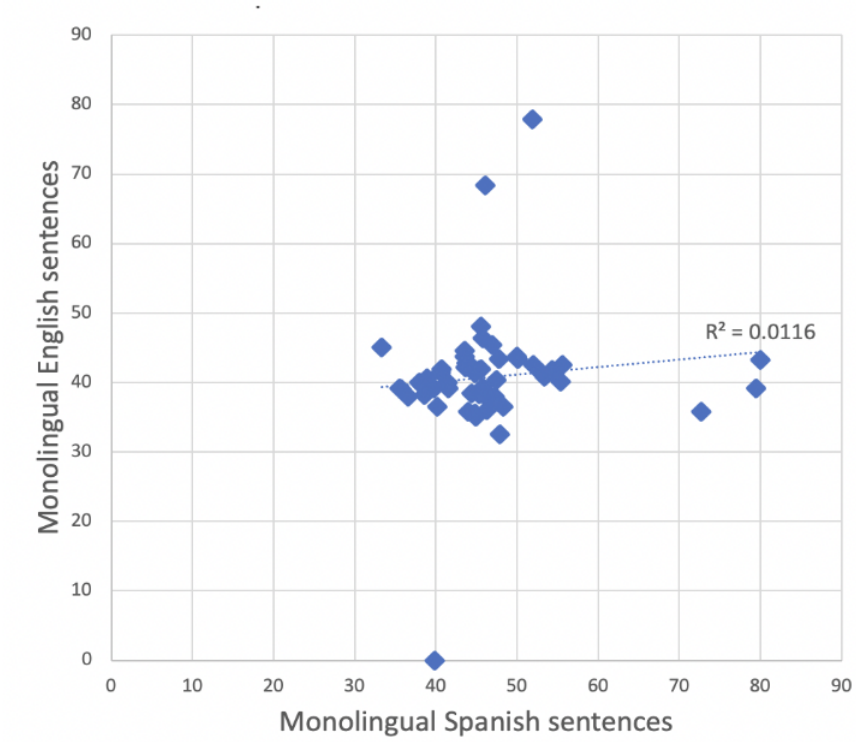


Figure 2.10: Comparing F-scores between monolingual English versus monolingual Spanish sentences by conversation in the Bangor Miami corpus

2.3.4 Investigating Formality in Speech

Method

Having carried out the text-based side of our investigation, we pivot back to the acoustic side, starting with the experimental setup. For speech-based formality, our workflow involved computing speaking rate and pitch at the utterance level in each conversation in the corpus, separated by speaker, and normalising to a 0-1 scale, before computing the variance in each feature across the corpus.

Experiments

As with the text-based metric, we began the speech-based experiments by performing calibration on monolingual corpora of known formality. Our baselining experiments confirmed that greater variance in speaking rate and pitch do indeed correspond to more informal speech corpora in general (see Figure 2.11). One slight exception in Figure 2.11) is the Call Home corpus in English, whose variance is lower than expected.

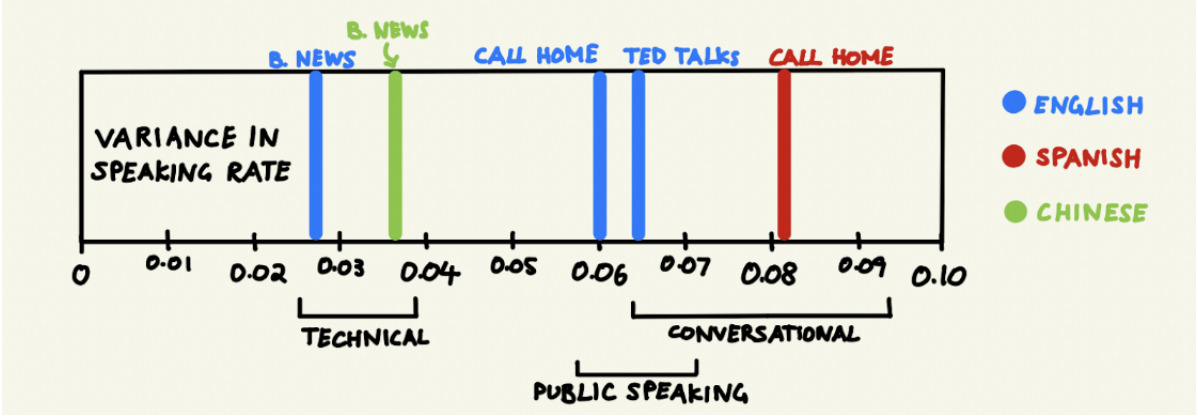


Figure 2.11: Calibrating variance of acoustic features on monolingual corpora of known formality

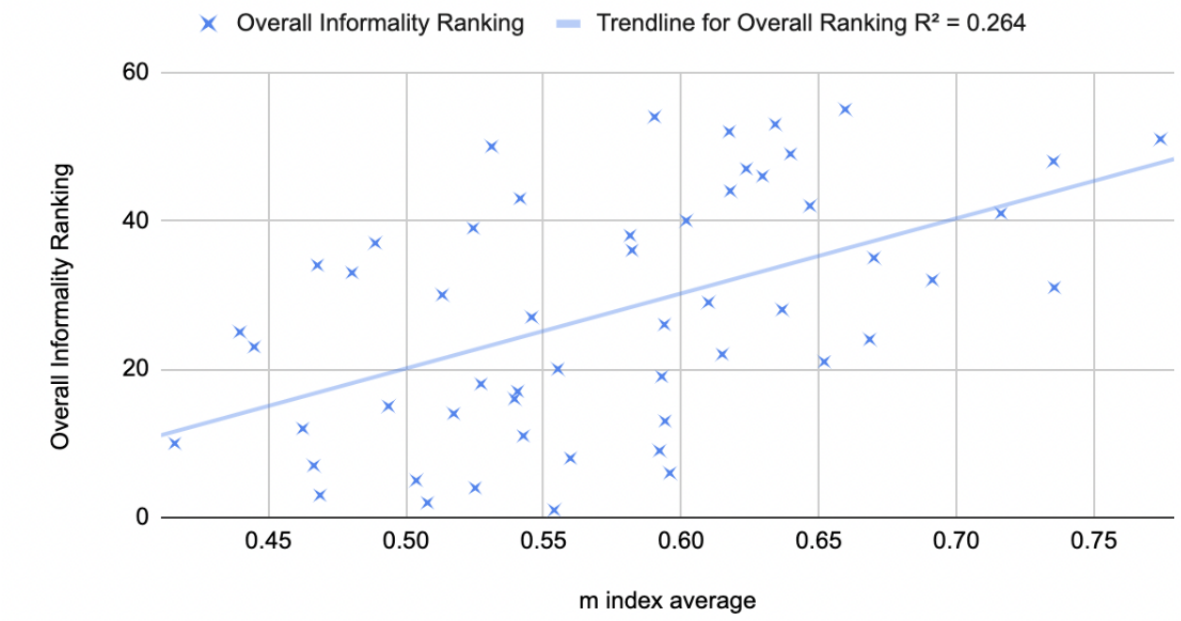


Figure 2.12: Overall informality ranking versus average m-index value of conversations in the Bangor Miami corpus

After calculating variance in speaking rate and pitch across our code-switching corpora, we aligned these results with the text-based F-scores, and combined the scores to create an informality ranking[4], where a greater ranking score corresponds to more informality. We chose to combine these because we think the text-based F-score encodes information that is complementary to and independent from speech-based acoustic scores, and we hypothesised that combining the two modalities would help provide a more complete characterisation of formality, which could more reliably be used to investigate relationships with code-switching for later prediction.

As a starting point for our ultimate goal of predicting propensity to code-switch within a corpus based on its formality, we investigated whether there was any correlation between the overall formality of a corpus and the amount of code-switching present, according to known metrics for measuring amount of code-switching. We particularly focused on the m-index metric. This is a token counting metric where a lower bound of 0 indicates pure monolingual speech, an upper bound of 1 indicates perfectly balanced bilingual speech, and any intermediate value indicates code-switching. Note that the following results generalise fairly well across other code-switching metrics, such as i-index, burstiness, and memory, as well.

As in Figure 2.12, we found a positive correlation between how informal the Bangor Miami corpus is and how much code-switching is present in the same corpus. Although the correlation is quite weak, it is statistically significant [p-value = 2.048e-12]. Given its statistical significance, the weak correlation might be a result of the ranking system we used, and so we plan to retry this analysis with the raw scores for text- and speech-based formality. We also think that it could be fruitful to try multivariate logistic regression here, with features other than formality, as it could be that formality does influence the amount of code-switching, but only when interacting with other features, such as topic or speech setting. While we are not absolutely sure to what extent formality and code-switching are dependent on each other, we do at a very basic level see what we expect, which is promising. We would like to note that we have found this correlation on only the Bangor Miami corpus so far, and that there can be noise even at the corpus level, so one direction for future work that we are pursuing is to do similar correlation analysis across more corpora, both Spanish-English corpora, and other language pairs, across other domains from which we can integrate other features. Overall, this finding acts as a good starting point for further refinement of formality metrics, and provides some insight into which methods of measuring code-switching align well with measuring formality.

2.3.5 Formality Takeaways

Overall, the key takeaways from this chapter are:

[4]Note that we omit conversations where speaking rate variance was either 0 or 0.5 due to there being only one or two utterances, and where conversations did not have pitch measurements. The maria18 conversation is not present in the combined ranking. Also note that we found starkly different orderings of formality between text scores and acoustic scores, i.e. acoustic features don't align with text based features. Only 3 out of 56 conversations have similar ranking positions between text order and acoustic order.

- Text-based F-score works fairly well on monolingual and code-switched data; however, it is not perfect due to slight language dependency.
- Speech-based variance score works in general too, but the weak correlation we found between overall (in)formality and code-switching suggests that there is room to further refine speech-based formality as well.
- Formality is best measured on large corpora, as F-score in particular is most reliable and stable at the corpus level.
- Slightly separate from our core findings are the automatic translation effects on syntax that we observed, which are worth keeping in mind when working with multilingual data in general.

2.4 Overall Conclusions

To summarize this chapter, we tentatively suggest age and gender can influence code-switching style, though we acknowledge that we do not investigate enough data-sets, nor are the differences significant enough, for this to be reliable. We also consider formality and here too find *some* evidence for it being a factor that can influence code-switching behaviour. However, we note the limitations of current formality metrics – including language dependency – thus paralleling Chapter 1's discussion of the pitfalls of existing code-switching metrics.

We would like to continue to explore the relationship between age, gender, formality and code-switching behaviour, as well as begin to consider the effects of other higher-level variables like those mentioned in Section 2.1. How does topic interact with formality and how does this affect code-switching style? What about the languages themselves? We hypothesize that the socio-political relationship between the languages, and their historical context, could be a particularly fruitful avenue to explore.

When conducting these types of experiments, we must also ensure we keep Chapter 1's discussion in mind. The current 'code-switching style' metrics we are utilizing as our analytic tools may be too coarse to pick up on certain stylistic differences. Our future workflow must thus be iterative. In response to Chapter 1's findings, we aim to propose and test out alternative characterizing metrics. In turn, we can repeat (and run novel) experiments akin to those discussed in *this* chapter, utilizing these updated metrics.

Finally, to end Part I, we hope that in the near future we will be able to utilize and integrate some of these findings into the actual building of ASR systems, thus making these insights and their utility more tangible.

Part II

Synthetic Code-Switching Text Generation

Building robust language models for code-switched tasks poses several challenges, largely due to the lack of available resources of code-switched text. In this section, we address two major properties inherent to code-switched data and research directions they support: (*i*) the lack of code-switching textual data and (*ii*) the diversity in code-switching across different languages and domains.

Getting access to large amounts of textual code-switch data: This is a challenging problem as textual code-switched data comes from real-world conversations, social media discussion, or other sources where privacy is a concern. Compared to monolingual or bilingual datasets, code-switched datasets are relatively scarce. While some code-switched corpora exist, they may be small in size, domain-specific, or focused on particular language pairs or communities. Accessing a diverse range of code-switched data that covers multiple languages and contexts can be a challenge. There is a concern in bias and representation since code-switched data availability can vary across languages and communities. Certain language pairs or communities may be underrepresented in the available datasets, leading to potential biases in the models trained on such data.

Structurally complex; large diversity in how code-switching manifests: Code-switching can occur at different linguistic levels, ranging from individual words or phrases to entire sentences or conversations. Moreover, the switching patterns can vary widely. Some individuals may predominantly insert single words from one language into another, while others may switch at the sentence level or mix languages within a single sentence. Modeling and understanding these diverse switching patterns require flexible language models capable of handling various code-switching phenomena. Sentence structure in code-switched sentences is complex due to the combination of multiple languages. Each language may have its own syntactic rules, word orders, and grammatical features. As a result, code-switched text may exhibit non-standard or hybrid sentence structures, making it challenging for language models to parse and generate accurate code-switched sentences. Code-switching is influenced by sociolinguistic factors, such as social context, formality, speaker identity, and language proficiency [25]. The choice to code-switch can vary depending on the situation, relationship between speakers, or even power dynamics. Capturing these sociolinguistic nuances and incorporating them into language models is crucial for generating code-switched text that reflects the appropriate social and cultural context.

Both these aspects of code-switching highlight a paradox specific to code-switching in that there are scarce resources for code-switching, but its inherent diversity demands more data. Techniques to synthetically generate code-switched text could help alleviate this challenge.

We aim to synthetically generate code-switched text using the following three methods:

- Using linguistic theories to guide text generation;
- Creating parallel text with little real code-switched text;
- Deploying pre-trained models on monolingual text

Chapter 3

Linguistic Theory Based Code-Switching Text Generation and its Practicality in Speech Recognition

Amir Hussein, Shammur Absar Chowdhury, Ahmed Ali, Sanjeev Khudanpur

Abstract

The pervasiveness of intra-utterance code-switching (CS) in spoken content requires that speech recognition (ASR) systems handle mixed language. Designing a CS-ASR system has many challenges, mainly due to data scarcity, grammatical structure complexity, and domain mismatch. The most common method for addressing CS is to train an ASR system with the available transcribed CS speech, along with monolingual data. In this work, we propose a zero-shot learning methodology for CS-ASR by augmenting the monolingual data with artificially generating CS text. We based our approach on random lexical replacements and Equivalence Constraint (EC) while exploiting aligned translation pairs to generate random and grammatically valid CS content. Our empirical results show a 65.5% relative reduction in language model perplexity, and 7.7% in ASR WER on two ecologically valid CS test sets. The human evaluation of the generated text using EC suggests that more than 80% is of adequate quality.

3.1 Introduction

Code-switching (CS) is a prevalent phenomenon in multi-cultural and multi-lingual societies due to the advent of globalization and as a remnant of colonialism. CS, wherein speakers alter between two or more languages during spoken discourse, is now receiving the attention of automatic speech recognition (ASR) researchers, making them address and model mixed-language input to ASR systems. Efforts have been made to design CS-ASR for a variety of language pairs, including Mandarin-English [26], Hindi-English [27], French-Arabic [28], Arabic-English [29, 30] and English-French-dialectal Arabic [31]. Some studies also discuss the complexities of, and the need for, CS ASR between dialects of a language [32]. Despite the aforementioned efforts, CS ASR still faces challenges due to the scarcity of transcribed resources, with skewed coverage of languages, dialects and domain mismatch.

We propose to augment the monolingual text with artificially generated CS data to reduce the effect of the domain mismatch and the data scarcity in ASR. In our experiments, we choose the Arabic language which is a morphologically complex language with more than 20 mutually-incomprehensible dialects, with modern standard Arabic (MSA) being the only standardized form [33]. The proposed methods should be generalized to other languages, e.g. Chinese dialects and Mandarin, subject to the availability of the corresponding monolingual resources.

There have been attempts to explain the grammatical constraints on CS based on Embedded-Matrix theory [34], Equivalence Constraint [35] and Functional Head Constraint [36]. In [37], authors proposed creation of grammatically valid artificial CS data based on the Equivalence Constraint Theory (EC), since it explains a range of interesting CS patterns beyond lexical substitution and is suitable for computational modeling. On the other hand, [38] proposed to model CS with a sequence-to-sequence recurrent neural network (RNN) with attention, which learns when to switch between and copy words from parallel sentences. In [39] researchers proposed generating multi-lingual CS data using mBert alignments with random replacements and then fine-tuning mBert on that data. The proposed approach showed significant improvements in five classification NLP

tasks. Although the aforementioned approaches for CS text augmentation provided substantial improvements in downstream NLP tasks, it is not clear if they will help in speech recognition.

In this work, we investigate the effectiveness of different CS text generation approaches to improve ASR performance in a zero-shot learning scenario. To mitigate the bias to a specific domain during the evaluation, we collect multi-dialectal Arabic-English CS test sets from different domains (Sports, Education, and Interviews) and different dialects (Levantine, Egyptian, Gulf, and Moroccan). In this study, we consider linguistic-based (Equivalence constraint) and lexicon-based (Random) CS generation approaches to answer the following two research questions in the context of ASR performance: 1) Is the knowledge of the number of switching points important? 2) Does linguistically motivated CS generation provides improvement over random lexical replacements? In addition, we propose a morphologically enhanced pipeline to generate realistic dialectal Arabic-English CS text. First, we build parallel data by translating original Arabic content into English. Later, we generate the CS content by mixing the pair of parallel sentences guided by the data alignments and different sampling techniques. Our method can be applied without the need for any CS speech data.

The Arabic language is a morphologically complex language with a high degree of affixation[1] and derivation – it is very challenging to obtain accurate alignments with the corresponding English translation. Thus, we propose to segment the Arabic text into morphemes. The segmentation allows aligning single morphemes with their English translations [40]. We evaluate the well-formed and acceptable generated sentences through subjective evaluation. We compare the mean opinion score (MOS) of the generated text data with ecological transcribed CS speech data, showing majority of the generated utterance are acceptable according to human judgements. The key contributions of this chapter are:

- Create the largest parallel multi-dialectal conversational Arabic-English text corpus.
- Develop a novel pipeline for generating Arabic-English CS text.
- Analyze human evaluation of the generated Arabic-English CS sentences.
- Evaluate the efficacy of the generated dialectal Arabic-English CS data in language modeling and ASR.

3.2 Generating Code-Switched Language

We begin by describing the proposed approach for generating synthetic Arabic-English CS based on EC theory.

3.2.1 Equivalence Constraint Theory

In EC theory, both languages $S1$ and $S2$ are defined by context-free grammars $G1$ and $G2$. Every non-terminal category $c1$ in $G1$ has a corresponding non-terminal category $c2$ in $G2$, and every terminal word $w1$ in $G1$ has a corresponding terminal word $w2$ in $G2$. These assumptions imply that intra-sentential code-mixing can only occur at places where the surface structures of two languages map onto each other, hence implicitly following the grammatical rules of both languages. In this work, we build our approach on top of the EC implementation in the GCM toolkit[2].

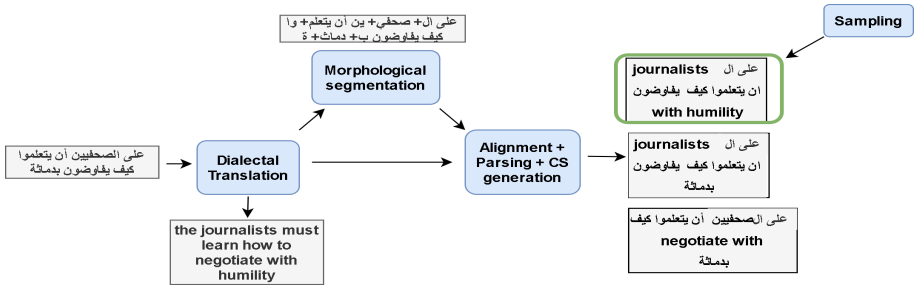


Figure 3.1: Illustration of the proposed Arabic-English CS generation pipeline.

3.2.2 Code-switching Text Generation

The input to the generation pipeline is a pair of parallel sentences $S1$ and $S2$, along with the word alignments. The $S1$ and $S2$ in our experiments are the English and Arabic languages respectively. The proposed CS generation pipeline includes four major components (also shown in Figure 3.1):

[1] A single word could represent multiple tokens. For example, the Arabic segment "وسيزر في ەقولهم" ("And they will plant it in their fields") map the first Arabic word to five English tokens, the last word represent the last two. Such 1-to-n mapping makes it difficult to build natural CS data.

[2] https://github.com/microsoft/CodeMixed-Text-Generator

1. **Parallel text translation:** We generate the parallel English text from the Arabic transcription using a public Machine Translation System [3], The system is built on transformer-based seq2seq model implemented in OpenNMT [41]. The Neural translation system is capable of translating Modern Standard Arabic as well as dialectal content [42]. It was fine-tuned on a large collection of coarse and fine-grained city-level dialectal data from diverse genres, such as media, chat, religion and travel with varying level of dialects. Such richness of the system makes it suitable for our task.

2. **Aligning the two sentences:** To generate word level alignments between the two sentences we use "fast-align" [43], and multilingual Bert (mBert) [44]. Arabic is agglutinative and morphologically complex language which is difficult to align with Romance and Germanic languages including English. To overcome this challenge, we segmented Arabic words into their stem, prefix(es) and affix(es) using the Farasa [45] segmentation. Segmentation has proven to be beneficial in reducing alignment complexity and improving tasks such as Machine Translation [46]. Figure 3.3a illustrates a complex alignment with several 1-to-many alignments. After using segmentation in Figure 3.3b, such cases largely disappear. Further, the segmentation allows resolving complex constructions that are caused by word re-ordering like " " that is aligned with "her opinion" in the reverse order; as well as making it easy to resolve co-references such as "هَا " that is mapped to both "she" and "her".

3. **Generating the Parse:** We use Stanford Parser [47] to generate a sentence level constituent parse tree for one of the source languages. Specifically, we parse the English sentence and use the alignments to generate the equivalent parse tree for the Arabic sentence.

4. **CS text Generation:** We generate CS text using two approaches: a) random lexical replacements using the alignments from step (2) and b) applying EC theory to generate Arabic-English CS text. To examine the effect of changing the percentage of substitutions in the sentence with lexical replacements, we used development set from QASR dataset [48] which contains around 16 hours of CS. Figure 3.2 shows that minimum perplexity (PPL) is found to be in the flattened part 5%-30%, hence to avoid overfitting, the code switching percentage is selected from around the middle (20%). As for the EC-based CS generation, the high level steps are described as follow:

   (a) Replace every word in the Arabic parse tree with its English equivalent.

   (b) Re-order the child nodes of each internal node in the Arabic tree such that their right-to-left order is similar to the original Arabic language.

   (c) In case of deviation between grammatical structures of the two languages then:

      i. Replace unaligned English words for any Arabic words with empty strings.

      ii. Collapse contiguous word sequences in English, aligned with same Arabic word(s), to a single multi-word node.

      iii. Flatten the entire sub-tree, between the above-collapsed nodes and their closest common ancestor, to accommodate the difference.

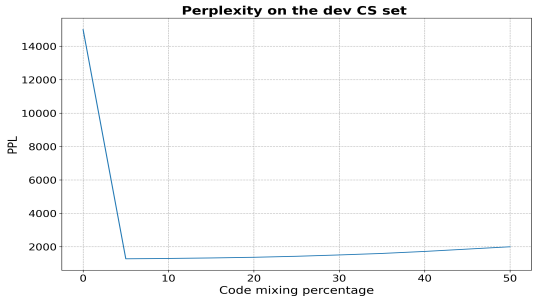
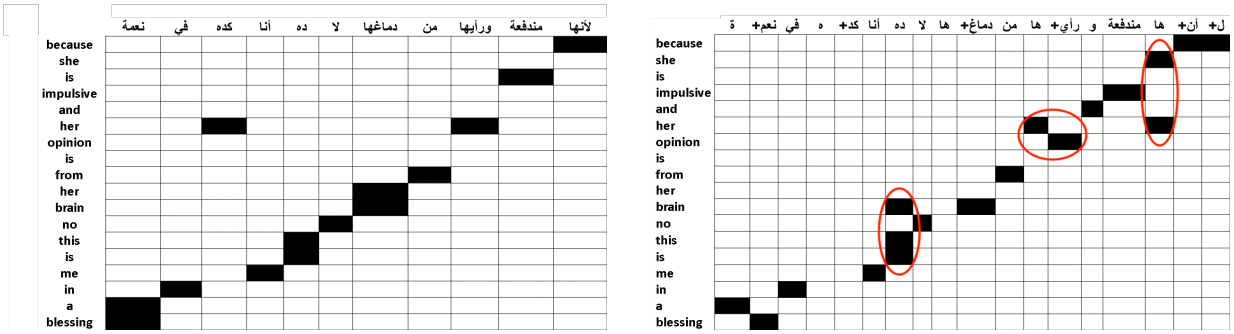


Figure 3.2: The perplexity of 3-gram model using generated code-switching text with random lexical replacement and different code mixing percentages.

[3] API access available from `https://mt.qcri.org/api`.



(a) (b)

Figure 3.3: Illustration of alignments with and without segmentation: (a) Word alignment between Arabic and English without segmentation. (b) Word alignment between Arabic and English with Arabic segmentation. Red circles shows complex construction that were solved after using segmentation.

3.2.3 Improved Naturalness through Sampling

In order to generate more natural code-switching (CS) sentences, we experiment with two sampling methods: random sampling and Switch Point Fraction (SPF) sampling. For *random sampling*, we arbitrarily pick a number of CS sentences from the generated data. For *SPF sampling*, we estimate the distribution of the number of switch points in a sentence based on empirical observations mentioned in [31, 48] and then we rank the generated CS sentences based on that distribution. We impose the following two constrains to make the generated data more acceptable to a bilingual speaker: 1) the sentence should start with an Arabic word, and 2) the number of English words should not exceed 45% of the total words in a sentence.

3.3 Corpora for Training and Evaluation

For empirical analysis of the proposed method, we trained the language models and speech recognition systems. We use several monolingual Arabic[4] and English datasets.

3.3.1 Monolingual Data Sets

MSA and Dialectal Arabic speech data: We use MSA and multidialectal training data collected from QASR [48], GALE [49], MGB3 [50], MGB5 [51], and an internal 156h Kanari multi-dialectal dataset.
English: To incorporate a variety of English data, we use the TEDLIUM 3[5] training set and a subset of 300h of SPGISpeech[6] datasets.

3.3.2 Evaluation Code-Switching Data Sets

For comprehensive model evaluation, we use three evaluation sets: monolingual Arabic with multiple dialects (MSA, Egyptian, Gulf, Leventian, North Africa), monolingual English, and two Arabic-English code-switching (CS) sets (ESCWA[7] and Kanari multi-dialectal internally collected data).

1. **Validation set:** for performance evaluation on diverse English and Arabic data, we combine MGB3-test 2h, QASR-test 2h, SPGI-test 1h, and Tedlium3-test 1h.
2. **ESCWA-CS** [29]: 2.8h of speech CS data collected over two days of United Nations meetings.
3. **Kanari-CS:** 4.8h of CS data from Levantine, Egyptian, Gulf and Moroccan dialects.

Table 3.1: Mean Opinion Score (MOS) rating scale.

Scores	Labels	Definition
1	Very Strange	The sentence is never going to be used by a human speaker
2	Strange	The sentence is unlikely to be used by a human speaker
3	Not very Natural	The sentence is probably not very natural but can be used by human speaker
4	Quite Natural	The sentence can used by human speaker
5	Perfectly Natural	The sentence is definitely used by human speaker

[4]For code-switching augmentation, we use Arabic monolingual data.
[5]https://openslr.magicdatatech.com/51/
[6]https://datasets.kensho.com/datasets/spgispeech
[7]https://arabicspeech.org/escwa

3.4 Evaluation of Model Performance

We perform objective and subjective evaluations of the quality of the generated CS text. For objective evaluation, we measure the efficacy of the data in language modeling and speech recognition. For subjective evaluation, we asked bilingual annotators to rate the naturalness/acceptability of the utterances.

3.4.1 Language Modeling Evaluation

We assess the quality of the generated text and its efficacy in handling CS in language modeling (LM) for both n-gram and neural LMs. We build a standard trigram LM using Kneser-Ney smoothing using SRILLM toolkit [52]. Moreover, we train Long short-term memory (LSTM) with 4 layers, $1,024$ output units and Stochastic gradient descent (SGD) using ESPNet toolkit [53] for 20 epoch.

3.4.2 Speech Recognition Evaluation

Our ASR system uses a hybrid HMM-DNN ASR architecture based on the weighted finite-state transducers (WFSTs) outlined in [54]. The training, development, and testing are the same as the Arabic MGB-2 [55] and the English TED-LIUM3 [56] tasks. For the hybrid ASR, we trained a Time Delay Neural Network (TDNN) [57] using sequence discriminative training with the LF-MMI objective [58] with the alignments from a context-dependent Gaussian mixture model-hidden Markov model (GMM-HMM). The input to the TDNN is composed of 40-dimensional high-resolution MFCC extracted from 25 ms frames and 10 ms shift along with 100-dimensional i-vectors computed from $1,500$ ms. Five consecutive MFCC vectors and the chunk i-vector are concatenated, forming a 300-dimensional features vector for each frame. We propose a multilingual architecture that merges all graphemes from multiple languages, keeping the language identity at the grapheme level. A multilingual n-gram language model is learned over the transcription for all the languages along with the augmented data.

3.4.3 Human Evaluation

For quality assessment of the generated CS data in terms of naturalness, we designed several crowdsourcing tasks using Amazon Mechanical Turk (MTurk).[8] The tasks aim to rate the utterances' acceptability in terms of the five quality categories (1-5) (see Table 3.1). We consider the green 3+ categories as acceptable sentences by annotators. The crowdsourcing task is performed using 3 sets: (1) 1K Generated CS data[9]; (2) ≈900 CS data, generated by random lexical replacements; and (3) ≈1.9K utterances from Kanari-CS evaluation dataset containing natural CS. For each task, we collected 3 judgements per utterances.[10] To ensure the reliability of the annotator, they have to pass an assessment test with a minimum of 80% score. The qualifying task includes answering a series of multiple-choice questions - designed by experts to reflect the annotators' language proficiency and understanding of the questions. A total of 32 annotators participated in the evaluation. We deliberately put constraints, in the experimental design, such that each evaluator can not annotate more than 15% of the utterance from each data set. This ensures that there is no implicit bias encoded in the decision that can influence the reported results. Using the three judgements, per utterance, we then calculated the mean opinion score (MOS) by averaging the judgment scores.

3.5 Empirical Results and Discussion

3.5.1 Objective Evaluation:

The perplexity (PPL) of the n-grams and RNNLM are presented in Table 3.2 and WER for the hybrid ASR with n-gram LM is reported in Table 3.3. For the n-grams, we observe a significant drop in PPL (18.6% and 22% on Kanari and ESCWA data sets respectively) when adding multi-dialectal parallel Arabic-English (Mono) text to the LM training. Adding synthetic CS based on word level alignments with equivalence constraint and random sampling improves the PPL further by 3.1%-3.5% in relative gain. Using Farasa segmentation with switching point factor (SPF) improves the PPL by 4%-7% compared to word level alignment, and by 2.5% compared to random sampling. The best PPL is achieved with Farasa segmentation and random lexical replacements with an overall relative gain of 55.5% and 65.5% on Kanari and ESCWA respectively compared to the baseline.

For the RNNLMs with fixed BPE tokenizer of size $1k$, we observe a significant reduction in PPL with respect to the monolingual LM (Baselines and Mono) after adding the augmented CS data. This shows that the model is benefiting more from the (synthetic) CS data than increasing the size of monolingual training data.

[8] `http://mturk.com`

[9] Using equivalence constraints, Farasa segmentation and SPF sampling.

[10] With a cost of 2 cent per judgment

This behavior can be attributed to the fact that subword-based LM can deal with out-of-vocabulary word and segmentation problems more effectively than a word-based n-gram model, thus reducing these factors' influence on the PPL changes. One can notice that the difference is negligible in PPL between EC-based CS generation techniques (random and SPF). This observation is aligned with the n-grams LM results. Finally, random lexical replacements perform significantly better than the EC-based approach.

A similar pattern can be observed from the HMM-TDNN performance with LM re-scoring. Table 3.3 presents the WER on two different test sets in the three experimental settings. However, WER results indicate that using SPF sampling provides almost no improvements over random sampling. The maximum improvement in WER is obtained with Farasa segmentation and random lexical replacements with a relative gain of 7.7% and 4% compared to the baseline. We test the significance in the WER improvements using Matched-Pair Sentence Segment Word Error (MAPSSWE) introduced by [59], with a significance level of p=5%. We found that the highest p-value is 0.002% as shown in Table 3.4, which is lower than the specified significance level 5%. Hence, we reject the null hypothesis and conclude that there is sufficient evidence that the differences in the results are statistically significant.

Table 3.2: Perplexity for n-gram and RNNLM on Kanari and ESCWA test sets. MGB2+Tedlium3 transcription (Baseline), collected Arabic English parallel text (Mono); Fast Align (FA); Equivalence Constraint (EC); Lexical Replacements (LR); Switch Point Fraction (SPF); random sampling with word alignment (Random); random sampling with Farasa segmentation alignment (Farasa+Random).

Perplexity	Kanari		ESCWA		
#Tokens	20,902		37,416		
	n-gram	LSTM	n-gram	LSTM	#sent
Baseline	5,284	127	5,565	87	784K
+Mono (Ar-En)	4,456	179	4,565	118	3.108M
+EC+Random (FA)	4,318	50	4,412	46	3.677M
+EC+Farasa+Random (FA)	4,206	47	4,379	46	3.677M
+EC+Farasa+SPF (FA)	4,102	51	4,272	50	3.677M
+EC+Farasa+SPF (mBert)	3,995	53.8	4,202	51	3.677M
+EC+ Farasa +SPF + Random (FA)	4,038	59	4,236	58	4.246M
+Farasa+LR (mBert)	3,398	43	3,362	45	3.603M

Table 3.3: WER with insertions: ins, deletions: del and substitutions: sub on Kanari and Escwa test sets. MGB2+Tedlium3 transcription (Baseline), (Mono): collected Arabic English parallel text; Fast Align (FA); Lexical Replacements (LR); Equivalence Constraint (EC); Switch Point Fraction (SPF); random sampling with word alignment (Random); random sampling with Farasa segmentation alignment (Farasa+Random).

Hybrid ASR	WER in % & [ins, del ,sub]	
LM Data	Kanari	ESCWA
Baseline	59.30 [532, 9,094 ,15,282]	49.25 [419, 3,668, 6,086]
+ Mono (Ar-En)	56.27 [587, 8,253, 14,796]	47.80 [365, 3,961, 5,549]
+EC + Random (FA)	56.18 [596, 8,219, 14,782]	47.79 [450, 3,418, 6,005]
+EC + Farasa + Random (FA)	55.89 [603, 8,093, 14,778]	47.62 [464, 3,372, 6,000]
+EC + Farasa +SPF (FA)	55.87 [590, 8,107, 14,769]	47.64 [370, 3,927, 5,545]
+EC+Farasa+SPF (mBert)	55.85 [608, 8,119, 14,733]	47.54 [451, 3,373, 5,996]
+EC+Farasa+SPF +Random (FA)	55.81 [603, 8,075, 14,763]	47.67 [373, 3,940, 5,535]
+Farasa+LR (mBert)	55.04 [618, 7,803, 14,698]	47.28 [491, 3,302, 5,973]

Table 3.4: MAPSSWE p-values between the baseline (B), baseline+mono (B+M), and baseline+Mono+CS with lexical replacements (B+M+LR) on Kanari and Escwa sets

	ESCWA		Kanari	
	B	B+M	B	B+M
B+M	< 0.001	-	< 0.001	-
B+M+LR	< 0.001	0.002	< 0.001	< 0.001

Table 3.5: Reported percentage of human judgement scores data falls under the MOS value range for the human-transcribed CS data (Kanari-CS), generated CS data using Equivalence Constrain (EC:SPF) and random lexical replacement (RLR). The $x <= * < y$ represent the MOS value range.

MOS	EC:SPF (# 1,170)		RLR (# 900)		Kanari-CS (# 1,921)	
$1 <= * < 2$	1.20%	16.15%	9.78%	74.67%	4.22%	41.96%
$2 <= * < 3$	14.96%		64.89%		37.74%	
$3 <= * < 4$	54.36%	83.85%	22.11%	25.33%	43.68%	58.04%
$4 <= * <= 5$	29.49%		3.22%		14.37%	

3.5.2 Subjective Evaluation:

The percentage of average judgment scores for each quality category are presented in Table 3.5. From MOS, we observe that ≈ 84% of the generated data with equivalence constraint and switch point factor is acceptable, while a random replacement is only 25% acceptable to human judges. Hence, reflecting the importance of the proposed pipeline for enriching CS data. Our result also suggests that generated CS data is cleaner than natural CS transcription, which contains overlapping speech, disfluency, and repetition among others. On the other hand, even though random replacement is less acceptable by humans, it helps the LM/ASR to see and learn more about CS as an effective augmentation technique. We further discuss this in the following Section.

3.5.3 Key Observations and Discussion

Both LM, and ASR with LM re-scoring results suggest that random lexical replacements provide the best performance on code-switching (CS) when testing on a new domain. We think that this is because CS, in general, is affected by different factors including the dialect, the topic, the social and educational status, the emotional state, the speaker's speech styles, and the proficiency in the two languages. Hence, CS in a new domain is an unpredictable phenomenon and in practice can be modeled as a random process. Furthermore, the ASR results show that the CS generation with the expected number of switching points (SPF) provides marginal improvements compared to random replacements. We think this is mainly because in a zero-shot learning scenario, in addition to SPF, the model needs information about the position where the switching is expected to happen. Human evaluation of the synthetic text shows that our method can generate natural CS augmented text. Furthermore, adding the CS data show significant improvement in PP and WER when combining Farasa with any text generation settings. Finally, a challenge in evaluating CS ASR is the metric itself. As shown in recent CS studies [31, 60], WER is not robust against partial/full transliteration of a correctly recognized word, hence doesn't fully reflect improvement in CS-ASR, as seen in the example below, where the words "International" and "Pharmatech" have a script mismatch between the reference and ASR output:

Ref	كون معروف هو وين و ال assessment بيكونوا على بعني مش personal *** نحن Pharamtech على international معمولين
Ref Translation	It is known where the assessment will be held, it is not personal we Pharamtech will do international
ASR	كون معروف هو وين assessment *** بيكونوا على بعني مش personal إن حنا فورمتك على إنترناشيونال معمولين

While the above combination of scores confirms the validity of the approach, inspecting sentences with low MOS and/or high perplexity reveals some shortcomings of the proposed approach. Several sentences that were scored lower than 3 had one common issue: the original sentences were not complete as shown below:

Example 1	**Original CS**	**وكان الأردن people و حكومة**
	Translated	**And was Jordan people and government**
Example 2	**Original CS**	**لا يـوجد support والدليل على**
	Translated	**There is not support and the proof is**

Though the CS algorithm chose the correct English replacement, the context seems missing or incomplete/broken, leading to a low score by the human judges. An additional issue was errors inherent in the MT system. Incorrect lexical choices create further ambiguity in the resulting CS sentences and result in low MOS. Some instances were a result of the neural MT generating more fluent output than its (disfluent) input, causing misalignment between the original and translated sentences, while impacting the quality of the generated CS. Restricting the generation in the step (3) to complete sentences while parsing the sentences would avoid some of the aforementioned problems. Additionally, using other MT systems can help to avoid the low-quality translations that degrade the CS generation.

Chapter 4

Investigating Lexical Replacements for Code-Switched Data Augmentation

Injy Hamed, Nizar Habash

Abstract

Code-switching (CS) poses several challenges to NLP tasks, where data sparsity is a main problem hindering the development of CS NLP systems. In this chapter, we investigate data augmentation techniques for synthesizing Dialectal Arabic-English CS text. We perform lexical replacements using parallel corpora and alignments where CS points are either randomly chosen or learnt using a sequence-to-sequence model. We evaluate the effectiveness of data augmentation on language modeling (LM), machine translation (MT), and automatic speech recognition (ASR) tasks. Results show that in the case of using 1-1 alignments, using trained predictive models produces more natural CS sentences, as reflected in perplexity. By relying on grow-diag-final alignments, we then identify aligning segments and perform replacements accordingly. By replacing segments instead of words, the quality of synthesized data is greatly improved. With this improvement, random-based approach outperforms using trained predictive models on all extrinsic tasks. Our best models achieve 33.6% improvement in perplexity, +3.2-5.6 BLEU points on MT task, and 7% relative improvement on WER for ASR task. We also contribute in filling the gap in resources by collecting and publishing the first Arabic English CS-English parallel corpus.

4.1 Introduction

Code-switching is the alternation of language in text or speech. CS can occur at the levels of sentences (inter-sentential CS), words (intra-sentential CS), and morphemes (intra-word CS or morphological CS). Data sparsity is one of the main bottlenecks hindering the development of CS NLP applications [61]. Given that CS data is scarce and that collecting such data is expensive and time-consuming, data augmentation serves as a solution that has proved to be successful at alleviating data sparsity. Most of the work done for CS data augmentation has been focused on language modeling, mostly for ASR rescoring. Several data augmentation techniques have been proposed based on linguistic theories [62, 63], heuristics [64, 65, 66], neural networks [67, 68, 69, 70], and machine translation [71].

CS data augmentation has been less investigated for the task of MT. The approaches mainly involved relying on parallel corpora and replacing source segments with their corresponding target segments. Replacements can be done using word alignments [72, 73, 74], phrase tables [75, 76], or parse trees under the Equivalence Constraint [77, 78]. In [72], all source words (except for stopwords) are replaced by the corresponding target words if one-to-one alignments exist. However, this will lead to over-modeling of CS as well as generation of CS sentences that are not linguistically correct. Researchers have also applied linguistic constraints for data augmentation. In [77], the authors provide the GCM tool which utilizes the equivalence constraint [79] to generate CS sentences. However, linguistic theories are not guaranteed to provide generalization across languages, especially for languages with syntactic divergence, such as Arabic and English. Moreover, by looking into the output of the GCM tool for the Arabic-English language pair, we notice that some sentences suffer from information loss, where the augmented sentences are incomplete, which was also reported in [78].

In this chapter, we investigate lexical replacements for augmenting Arabic-English CS data. We investigate whether a neural-based predictive model can be trained to predict CS switch points. The training of the

predictive model is done using limited amounts of CS data, and the model is then used to convert monolingual Arabic-English parallel data into CS-English data, by projecting the chosen words from the target to the source side. This has only been previously investigated for Hindi-English in [73]. In this chapter, we provide further experiments, exploring the use of different types of alignments. We evaluate the usefulness of the augmented data for language modeling, machine translation, and ASR rescoring tasks.

4.2 ArzEn Parallel Corpus

ArzEn Speech Corpus	
Duration	12h
#Speakers	40
# Sentences	6,290
# Tokens	102,332
% Code-mixed (CM) sentences	63.2%
% English words in CM sentences	18.7%
% Sentences with morphological CS	28.7%
# Morphological CS Words	2,684

(a) ArzEn corpus statistics.

#	Example
1	←انا كتبت ال project code *AnA ktbt Al* project code I wrote the #project #code
2	←عملت كذا internship *Emlt k*A* internship I did several #internships
2	←كنت ب overload الناس اللى معايا *knt b* overload *AlnAs Ally mEAyA* I was #overloading my teammates
3	←ن detect ال traffic within period معينة *n* detect *Al* traffic within period *mEynp* to #detect the #traffic #within a certain #period

(b) ArzEn corpus examples, showing source text, its transliteration [80], and translation. The arrows beside the sentences show the sentence starting direction, as Arabic is read right to left.

Table 4.1: ArzEn corpus overview.

We present our work on collecting the first parallel corpus for CS Egyptian Arabic and English[1]. We collect our corpus by translating ArzEn speech corpus [81] which consists of informal interviews discussing broad topics, such as education and life experiences. The corpus consists of $6,290$ sentences having $99,340$ words ($102,332$ tokens). In Table 4.1a, we provide an overview of CS frequency in ArzEn. The corpus was translated by professional translators and the test and dev sets were revised by the translators and one of the authors. Our guidelines generally follow the LDC Arabic-to-English Translation Guidelines [82]. We modified the guidelines with regards to CS segments. In the LDC guidelines, it is recommended that embedded English segments in the source text should be copied to target text without changes or corrections. We allow modifying CS segments in translation for better fluency and to handle morphological CS. The translators were also asked to mark the CS segments occurring on the source side in the provided translations using a hashtag prefix to allow for further linguistic analysis. Examples are given in Table 4.1b.

Given that Egyptian Arabic is a morphologically rich language [83], Egyptians attach Arabic clitics and affixes to English words when code-switching. Morphological CS is present in Arzen, mostly occurring with clitics (88%). Among the CS sentences, 45% contain morphologically code-switched words. Morphological CS exacerbates the CS data sparsity problem, as it results in higher OOV rates.

4.3 Data Augmentation

We augment monolingual Arabic-English parallel corpora by injecting words from the target side to the source side. The augmentation process consists of two main steps: (1) CS Point Prediction: identifying the target words to be borrowed, and (2) CS Generation: performing the replacements. In this section, we will elaborate on the methodology for each of the two steps. Figure 4.1 provides an overview on the approach.

4.3.1 CS Point Prediction

Similar to [73], we model the task of CS point prediction as a sequence-to-sequence classification task. The neural network takes as input the word sequence $x = \{x_1, x_2, .., x_N\}$, where N is the length of the input sentence. The network outputs a sequence $y = \{y_1, y_2, .., y_N\}$, where $y_n \in \{1,0\}$ represents whether the word x_n is to be code-switched or not.

[1]The corpus and guidelines will be made available.

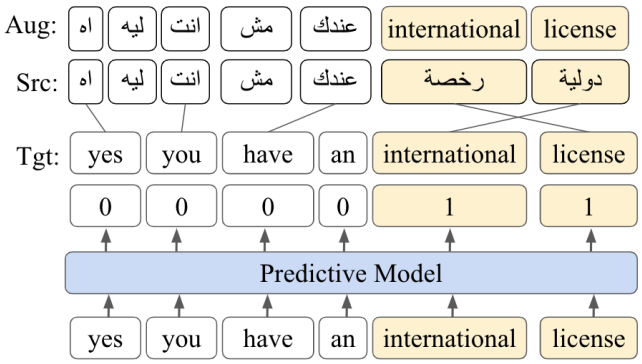


Figure 4.1: Data augmentation process.

In order to learn CS points, the neural network needs to take as input monolingual sentences from either the source or target sides, along with tags representing whether this word should be code-switched or not. In [73], the authors generated synthetic monolingual sentences from CS sentences by translating CS segments to their source language, and then learn CS points on the source side. While this approach seems more intuitive, CS segments abide by the grammatical rules of the embedded language, thus direct translation of embedded words would result in sentences having incorrect structures in the matrix language in case of syntactic divergence. In our case, both languages are syntactically divergent, and the English segments in ArzEn have an average length of 1.9 words, therefore, translating whole segments word-by-word will not produce correct source monolingual sentences to train the model on. Instead, we opt to learn the CS points on the target side. This approach provides another advantage, as English is commonly used in CS, having the predictive model trained on English as opposed to the primary language (which could be low-resourced) allows for the use of available resources such as pretrained language models.

The challenge in this approach is identifying the words on the target side which correspond to the CS words on the source side. Relying on the translators to perform this annotation task is costly, time consuming, and error-prone[2]. Relying on word alignments is also not optimal, where only 83% of CS words in ArzEn train set were matched using intersection alignment. Recall could increase using a less strict alignment approach, but would be at the risk of less accurate matches.

By examining the CS segments on source and target sides, we noticed that matching of segments can be achieved based on counts. Our matching algorithm is based on the following idea: if a CS segment occurs x times in the source and target sentences, then we identify these segments as matching segments. We match segments starting with the longest segments (and sub-segments) first. When matching words, we check their categorial variation [84] as well as stems to match words having slight modifications in translation. In case $|matches_{tgt}| > |matches_{src}|$, we first rely on alignments to make the decision, achieving 99.6% matches on ArzEn train set, then we randomly pick matched target segments to cover the number of matches on the source side in order to increase recall. This matching algorithm provides a language-agnostic approach to identify words on the target side that are code-switched segments on the source side. Examples of algorithm output are shown in Table 4.2.

Examples
Src: →i was a junior ta ف في فترة فجربت الموضوع i love ال academic life شوية.
Tgt: and #i #was #a #junior #ta for a period of time so i have tried this and #i #love the #academic #life a bit .
Src: ← ماكنتش م expect اني اشوف city زي دي اساسا
Tgt: i wasn 't #expecting to see such a #city in the first place .

Table 4.2: Examples of source and target lines showing the output of the matching algorithm, where the matched words are marked with '#'.

[2]We have tried this annotation task for ArzEn and only 72% of the CS words got annotated.

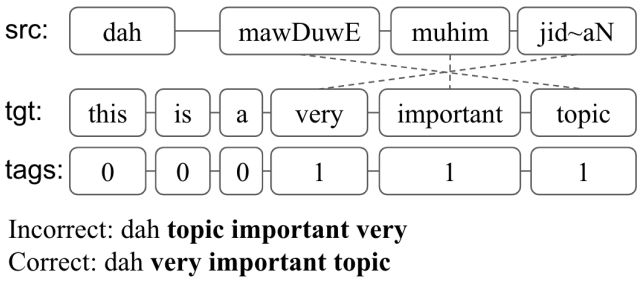


Figure 4.2: Data augmentation under the Continuity Constraint.

4.3.2 CS Generation

After identifying the English words to be embedded into the source side, we rely on alignments using GIZA++ [85] to perform the replacements. Using alignments for this task has been previously proposed [72, 73]. While direct replacements can be performed in the case of single word switches, in the case of replacing multiple consecutive words, direct word replacements would produce incorrect CS structures when the order between words in both languages is different. In the case of Arabic-English, this is particularly evident for adjectival phrases. Accordingly, when performing word replacements, we maintain the same order of words in embedded CS segments, which we refer to as the "Continuity Constraint". In Figure 4.2, the importance of applying this constraint is shown in the example. Without such a constraint, the generated sentence outlined in Figure 4.2 would follow the Arabic syntactic structure resulting in "dah topic important very" (this is a topic important very).

When performing replacements, we first explore using 1-1 alignments obtained using intersection alignments. In order to increase the alignment coverage, we rely on alignments obtained in word as well as in stem spaces. By training the alignments in word space, we achieve 67.0% coverage on source side and 46.8% on target side. By training the alignments in stem space, we achieve 75.3% and 52.4% on source and target sides, respectively. Therefore, in order to maximize CS replacements, we rely on the alignments obtained in stem space, and add remaining alignments obtained from word space, such that 1-1 alignments are retained. This approach achieves 77.0% and 53.6% coverage on source and target sides.

While intersection alignment provides high precision, relying on 1-1 alignments is not always correct, as an Arabic word can map to multiple English words and vice versa. In an approach similar to [74], we investigate the use of grow-diag-final alignments to identify aligned segments. The aligned segments consist of pairs of the minimal number of consecutive words (S,T) where all words in source segment (S) are aligned to one or more words in target segment (T) and are not aligned to any other words outside (T), with the same constraints applying in the opposite (target-source) direction. Afterwards, for each English word receiving a positive CS tag from the predictive model, the whole target segment containing this word replaces the aligned source segment. Throughout the chapter, we will refer to the two approaches as using 1-1 and n-n alignments. In Figure 4.3, we present an example showing the results of applying replacements using 1-1 versus n-n alignments for random as well as trained CS predictive models. We experiment with relying on alignments trained on word space only, stem space only, and the union of both alignments. We find that relying on the union achieves better perplexity as well as MT results. Therefore, we will only be presenting the results using the merged alignments.

4.4 Experiments

4.4.1 Data Preparation

For Arabic English CS-English parallel data, we use our collected ArzEn train set, consisting of $3.3k$ sentences. We utilize the following Arabic-English parallel corpora: Egyptian Arabic-English translations obtained from Callhome Egyptian Arabic-English Speech Translation Corpus [86], LDC2012T09 [87], LDC2017T07 [88], LDC2019T01 [89], LDC2020T05 [90], and MADAR [91]. The corpora contain $308k$ monolingual parallel sentences. These sentences are used for data augmentation and are included as part of the training set. The corpora also contain $15k$ CS parallel sentences. We do not include those in training, as we use them as reference to compare the performance of MT models using augmented CS data versus using the available real CS parallel data. For corpora with no defined data splits, we use the guidelines provided in [92]. For development and testing, we use ArzEn dev and test sets, having $1,402$ and $1,471$ sentences, containing 894 and 907 code-mixed sentences, respectively. Data preprocessing involved removing all corpus-specific annotations, removing URLs and emoticons, lowercasing, running Moses' [93] tokenizer as well as MADAMIRA [94] simple tokenization (D0), and performing Alef/Ya normalization.

	؟	لاربعة	ترابيزة	عندك
do				
you				x
have				
a				
table			x	
for				
four		x		
?	x			

Random CS Predictive Model:
Output using 1-1 alignments: have ترابيزة لاربعة ؟
Output using n-n alignments: do you have ترابيزة لاربعة ؟

mBERT$_{cs}$ Predictive Model:
Output using 1-1 alignments: ? table four عندك
Output using n-n alignments: ? a table for four عندك

Figure 4.3: Example showing replacements using 1-1 versus n-n alignments. The intersection alignments are marked with 'x' and the grow-diag-final alignments are highlighted in grey.

4.4.2 Predictive Models

Random: We randomly pick x source-target replacements from the existing alignments. We set x to 19% of the source words, where this number is chosen based on the percentage of English words in CS sentences in the ArzEn train set, given that we would like to mimic natural CS behaviour.

Finetuning BERT/mBERT: We fine-tune pretrained BERT and mBERT models using NERDA framework [3] [95]. We set the epochs to 5, drop-out rate to 0.1, warmup steps to 500, batch size to 13, and learning rate to 0.0001.

4.4.3 Machine Translation System

We train a Transformer model using Fairseq [96] on a single GeForce RTX 3090 GPU. We use the hyperparameters from the FLORES benchmark for low-resource machine translation [97]. The hyperparameters are given in Appendix 4.8. We use a BPE model trained jointly on source and target sides with a vocabulary size of $8k$ (which outperforms $1, 3, 5, 16, 32, 64$k). The BPE model is trained using Fairseq with character_coverage set to 1.0. The baseline model is trained using the parallel data outlined in Section 4.4.1. In the experiments involving augmentation, we add the augmented sentences to the original training data.

4.4.4 Automatic Speech Recognition System

We use the best performing E2E ASR system outlined in [98]. The joint CTC/attention based E2E system was trained using ESPnet [99]. The encoder and decoder consist of 12 and 6 Transformer blocks with 4 heads, feed-forward inner dimension 2048 and attention dimension 256. The CTC/attention weight (λ_1) is set to 0.3. SpecAugment [100] is applied for data augmentation. The ASR system is trained on the following Egyptian Arabic data: ArzEn [81], Callhome [101] and MGB-3 [102]. A subset of 5-hours was used from each of Librispeech [103] (English) and MGB-2 [104] (MSA), where adding more data from these corpora deteriorated the ASR performance [98]. For the LM baseline model, we use transcriptions from the Egyptian Arabic speech corpora: ArzEn, Callhome, and MGB-3. For the LM models using augmented data, we append the augmented data to those transcriptions. The RNNLM consists of 1 LSTM layer with 1000 hidden units and is trained for 20 epochs. For decoding, the beam size is 20 and the CTC weight is 0.2.

4.5 Results

4.5.1 Intrinsic Evaluation

We evaluate the accuracy of the CS predictive models and the CS nature in the synthesized sentences.

[3] We maintain the original tokenization of the input text, where we project further tokenization performed on the output into the original tokenization.

Predictive Model Evaluation

PM	A	P	R	F1
Random	77.3	19.6	22.0	0.207
BERT_{All}	91.7	77.0	55.1	0.643
mBERT_{All}	91.7	77.6	54.3	0.639
BERT_{CS}	**92.1**	**78.5**	**57.2**	**0.662**
mBERT_{CS}	**92.1**	**78.5**	56.8	0.659

Table 4.3: Evaluating the performance of the predictive models (PM) on the code-mixed sentences in ArzEn Dev set with regards to Accuracy (A), Precision (P), Recall (R), and F1 score. We report the results for training the predictive models using all ArzEn train set (BERT_{All}/mBERT_{All}) as well as using only the code-mixed sentences (BERT_{CS}/mBERT_{CS}).

We compare the CS points predictions provided by the predictive models against the actual CS points in the code-mixed sentences in ArzEn dev set. We present accuracy, precision, recall, and F1 scores in Table 4.4. While these figures give us an intuition on the performance of the predictive models, it is to be noted that false positives are not necessarily incorrect. It is also to be noted that the high accuracy values are due to the high rate of true negative predictions. We observe that training the BERT/mBERT predictive models on code-mixed sentences rather than all sentences in ArzEn train set allows the model to better mimic the actual CS pattern. Experiments showed that the best LM and MT models are achieved by fine-tuning mBERT on ArzEn CS sentences. Therefore, we will only be reporting the results of mBERT_{CS}.

CS Synthetic Data Analysis

PM	Align.	$\%\text{En}_{words}$	CMI	—$\text{CS}_{seg.}$—	$\%\text{En}_{sent.}$
Random	1-1	19.9	0.22	1.14	0.0
mBert_{CS}	1-1	16.7	0.22	1.23	7.7
Random	n-n	27.8	0.26	2.26	6.8
mBert_{CS}	n-n	28.8	0.26	2.84	20.6
ArzEn		18.6	0.19	1.88	3.8

Table 4.4: Evaluating augmented sentences in terms of CS metrics against ArzEn train set.

We look into how similar the augmented sentences are to naturally occurring CS sentences. In Table 4.4, we evaluate the augmented sentences in terms of the percentage of English words, the Code-Mixing Index (CMI) [105], and the average length of CS segments of the generated code-mixed sentences, as well as the percentage of monolingual English sentences generated. We observe that using 1-1 alignments, the generated CS sentences are close to natural occurring CS sentences in terms of CS metrics. By using n-n alignments, the amount of CS present in the synthetic data increases considerably.

4.5.2 Extrinsic Evaluation

We evaluate the improvements achieved through data augmentation on LM, MT, and ASR tasks.

Language Modeling

PM	Align.	—Train—	Dev/Test PPL	OOV(%)
baseline	-	312,030	206.3/180.8	22.3
+Random	1-1	+240,866	157.2/139.0	10.3
$+\text{mBERT}_{CS}$	1-1	+176,920	146.8/137.4	11.2
+Random	n-n	+206,883	138.2/120.2	10.6
$+\text{mBERT}_{CS}$	n-n	+137,886	**133.8/120.1**	10.3

Table 4.5: For the different settings, we report LM PPL and out-of-vocabulary (OOV) rates evaluated on the CS sentences in ArzEn test set.

We use language modeling and evaluate perplexity on the CS sentences in ArzEn test set to evaluate how close the generated synthetic data mimics real CS data. We train a Transformer model using Fairseq [106] on a

single GeForce RTX 3090 GPU. The hyperparameters are given in Appendix 4.9. The same training data used for the MT experiments is used for training the LM and ArzEn dev CS sentences are used for development, where the same BPE model is used as the MT experiments.

As shown in Table 4.5, when using 1-1 alignments, mBERT$_{CS}$ results in better PPL over random. We also observe that when using n-n alignments, the improvements achieved from relying on a trained predictive model decreases.

Machine Translation Evaluation

PM	Align.	Tok.	All	CM
baseline	-	D0	49.2	49.3
+Random	1-1	D0	51.7	52.7
+mBERT$_{CS}$	1-1	D0	52.7	53.9
+Random	n-n	D0	**53.0**	**54.4**
+mBERT$_{CS}$	n-n	D0	52.8	54.1
+realCS	-	D0	52.2	53.2
baseline	-	D3	49.8	49.7
+Random	1-1	D3	52.3	53.4
+mBERT$_{CS}$	1-1	D3	51.9	53.0
+Random	n-n	D3	**52.9**	**54.2**
+mBERT$_{CS}$	n-n	D3	52.4	53.6
+realCS	-	D3	52.0	53.0

Table 4.6: MT Results showing the Avg_{MT} scores on all and code-mixed sentences of ArzEn test set.

We evaluate the MT models using BLEU [107], chrF, chrF++ [108], and BERTScore (F1) [109]. BLEU, chrF and chrF++ are calculated using SacrebleuBLEU [110]. Given that each evaluation metric has its strengths and drawbacks, we will present the results using an average of the 4 metrics: Avg_{MT}. The full results are presented in Table 4.8 in Appendix 4.7. For each experiment, we report the Avg_{MT} for all and code-mixed sentences in ArzEn test set, as presented in Table 4.6. Motivated by the work of [111], we also investigate applying D3 tokenization on the training data before training the MT models. Results show that all models utilizing augmented data outperform the baseline. Using n-n alignments achieves improvements over using 1-1 alignments on all settings. The best model is achieved using random-based predictions utilizing n-n alignments, showing significant improvements across all evaluation metrics. Using D0 tokenization, the model achieves improvements of +4.7/+5.7 BLEU points on all/code-mixed sentences of ArzEn test set. Using D3 tokenization, the model achieves +3.2/+4.6 BLEU points improvements. We also compare the performance of our models against training an MT model with the baseline training data in addition to the $15k$ available CS parallel sentences. We report that our best-performing model outperforms the model utilizing additional real CS parallel data on all evaluation metrics, confirming the effectiveness of the data augmentation technique.

Qualitative Analysis When looking into the translations provided by the baseline model, we observe that many CS words get dropped in translation or get mistranslated. This is the case for 23% of the embedded words, covering 52% of the sentences. When checking the translations provided by the MT systems that are trained using augmented sentences, whether using random approach or trained predictive model, we observe that the majority of the CS words are retained through translation. We also observe that these MT systems are able to retain CS OOV words, where the OOV words are not available in the baseline training set, nor introduced in the synthetic data. This shows that by adding CS synthetic sentences to the training set, whether using random approach or trained predictive model, the main improvement in performance is achieved as the models learn to retain English words in translation. In Table 4.9, we provide examples of reference and translation pairs provided by different models. We observe that the models trained with synthetic data, in opposition to the baseline model, are able to retain all CS words in the examples, including the words “poker”, “garou”, and “belle” (french) which are OOV words.

4.5.3 ASR

In Table 4.7, we report the ASR results. Results show that all models utilizing augmented data outperform the baseline. Using n-n alignments achieves improvements on all settings. The best model is achieved using random-based approach utilizing n-n alignments, achieving 2.5% absolute WER reduction. We compare our results against the E2E ASR model in [98] where the LM is trained using the transcriptions in addition to: (1) 5,427 sentences (having 3,772 CS sentences) gathered from interview transcriptions in a similar setting as

PM	Align.	PPL	WER	CER
baseline	-	467.6	35.5	20.7
+Random	1-1	302.8	33.5	19.3
+mBERT$_{CS}$	1-1	274.3	33.8	19.2
+Random	n-n	**255.9**	**33.0**	**18.9**
+mBERT$_{CS}$	n-n	287.3	33.4	19.1
+realCS	-	116.1	32.8	18.8

Table 4.7: ASR Results showing Word Error Rate (WER) and Character Error Rate (CER) on ArzEn test set.

ArzEn data [112] and (2) 1,574,208 CS sentences collected from social media platforms [113]. We observe that the random model utilizing n-n alignments achieves WER and CER that is close to that achieved by the model trained on 7 times more of real CS data.

4.6 Conclusion

In this chapter, we investigate data augmentation for CS Egyptian Arabic-English. We rely on monolingual parallel corpora and alignments to make lexical replacements, where CS switch points are either selected randomly or based on predictions of a sequence-to-sequence model that is trained on a limited amount of CS data. We investigate word replacements using 1-1 alignments as well as segment replacements (n-n) identified using grow-diag-final alignments. We evaluate the effectiveness of data augmentation on the tasks of LM, MT, and ASR. We achieve improvements on the extrinsic tasks by using n-n alignments, where we use the union of alignments trained in word and stem spaces. Under this setting, random-based replacements outperform trained predictive models on all downstream tasks. Our best models achieve 33.6% improvement in perplexity, +3.2-5.6 BLEU points on MT task, and 7% relative improvement on WER for ASR task. As part of our work, we also collect the first parallel corpus for CS Egyptian Arabic English, which we plan to make publicly available.

Appendix

4.7 MT Results

In Table 4.8, we report the figures for MT evaluation, showing the scores on the following evaluation metrics: BLEU, chrF, chrF++, and BERTScore(F1).

			All Sentences					CM Sentences				
PM	Align	Tok.	BLEU	chrF	chrF++	F_{BERT}	Avg$_{MT}$	BLEU	chrF	chrF++	F_{BERT}	Avg$_{MT}$
baseline	-	D0	31.2	54.1	52.8	0.587	49.2	31.8	55.1	53.8	0.566	49.3
+Random	1-1	D0	33.7	56.4	54.9	0.617	51.7	34.8	58.0	56.5	0.613	52.7
+mBERT$_{CS}$	1-1	D0	35.0	57.4	55.9	**0.624**	52.7	36.5	59.3	57.7	0.622	53.9
+Random	n-n	D0	**35.9**	**57.6**	**56.1**	0.623	**53.0**	**37.5**	**59.6**	**58.0**	**0.623**	**54.4**
+mBERT$_{CS}$	n-n	D0	35.2	57.5	**56.1**	0.623	52.8	36.7	59.4	57.9	0.622	54.1
+realCS	-	D0	34.4	57.0	55.6	0.617	52.2	35.8	58.7	57.3	0.612	53.2
baseline	-	D3	32.3	54.7	53.4	0.588	49.8	32.4	55.6	54.2	0.5674	49.7
+Random	1-1	D3	34.5	57.1	55.7	0.619	52.3	35.9	58.9	57.5	0.613	53.4
+mBERT$_{CS}$	1-1	D3	34.0	56.6	55.2	0.619	51.9	35.3	58.4	56.9	0.614	53.0
+Random	n-n	D3	**35.5**	**57.6**	**56.3**	**0.623**	**52.9**	**37.0**	**59.5**	**58.1**	**0.621**	**54.2**
+mBERT$_{CS}$	n-n	D3	34.6	57.1	55.7	0.620	52.4	36.0	59.0	57.5	0.620	53.6
+realCS	-	D3	34.1	56.7	55.2	0.618	52.0	35.4	58.5	56.9	0.613	53.0

Table 4.8: MT evaluation showing the following scores on ArzEn test set: BLEU, chrF, chrF++, F1 BERTScore (F_{BERT}), and Avg$_{MT}$.

4.8 MT Hyperparameters

The following is the train command:
python3 fairseq_cli/train.py $DATA_DIR –source-lang src –target-lang tgt –arch transformer –share-all-embeddings –encoder-layers 5 –decoder-layers 5 –encoder-embed-dim 512 –decoder-embed-dim 512 –encoder-ffn-embed-dim 2048 –decoder-ffn-embed-dim 2048 –encoder-attention-heads 2 –decoder-attention-heads 2 –encoder-normalize-before –decoder-normalize-before –dropout 0.4 –attention-dropout 0.2 –relu-dropout 0.2 –weight-decay 0.0001 –label-smoothing 0.2 –criterion label_smoothed_cross_entropy –optimizer adam –adam-betas '(0.9, 0.98)' –clip-norm 0 –lr-scheduler inverse_sqrt –warmup-updates 4000 –warmup-init-lr 1e-7 –lr 1e-3 –stop-min-lr 1e-9 –max-tokens 4000 –update-freq 4 –max-epoch 100 –save-interval 10 –ddp-backend=no_c10d

4.9 LM Hyperparameters

The following is the train command:
python3 fairseq_cli/train.py –task language_modeling $DATA_DIR –save-dir $CHECKPOINTS_DIR –arch transformer_lm –share-decoder-input-output-embed –dropout 0.1 –optimizer adam –adam-betas '(0.9, 0.98)' –weight-decay 0.01 –clip-norm 0.0 –lr 0.0005 –lr-scheduler inverse_sqrt –warmup-updates 4000 –warmup-init-lr 1e-07 –tokens-per-sample 512 –sample-break-mode none –max-tokens 2048 –update-freq 16 –fp16 –max-update 50000 –patience 10

4.10 Translation Examples

In Table 4.9, we show examples of source-target pairs with their translations obtained from different MT models.

#	Model	**Example**
1	src	ما هو المفروض ال . . ال . . الناس اللي بت adjudicate يبقوا poker face فهو ماينفعش يفهمني اي حاجة بس بعديها بيبقي يعني بعرف غلطي ، بس
	tgt-ref	those one who adjudicate should have a poker face, so i can't get any signal from them, but afterwards i know my mistake, that's all
	baseline	the.. the.. the. the people who are administrative, they become a pokir face, so he can't understand anything, but after that i know my mistake, that's it
	random(1-1)	the.. the.. the.. the people who adjudicate should be poker face, so it can't explain anything to me, but after that i know my mistake, that's it
	random(n-n)	the.. the.. the.. the people who adjudicate should be poker face, so he cannot understand me anything, but after that i know my mistake, that's it
	mBERT$_{CS}$(1-1)	it's supposed that. the.. the.. the people who adjudicate become a poker face, so he can't explain anything to me, but after that i know my mistake, that's it
	mBERT$_{CS}$(n-n)	the.. the. people who adjudicate are supposed to be poker face, so it can't understand me anything, but after that i know my mistake, that's it
2	src	لو ماقدرش مثلا يبقي عنده feedback ان استهلاكه لسه عالي
	tgt-ref	if he failed for instance, he will get a feedback that his consumption is still high.
	baseline	if he can't, for example, he has a tank that his consumer is still high.
	random(1-1)	if he can't for example have feedback that his consumption is still high.
	random(n-n)	if i can't for example have feedback to consumption is still high.
	mBERT$_{CS}$(1-1)	if he can't, for example, he has a feedback to consume it is still high.
	mBERT$_{CS}$(n-n)	if he can't, for example, have feedback that his consumption is still high.
3	src	طيب ال favorite song بتاعتي هي french اسمها belle بتاعة garou
	tgt-ref	okay, my favorite song is french, it's called belle by garou.
	baseline	okay, my favorite song is french, it's called belly of cylinder
	random(1-1)	ok my favorite song is french its name belle of garou
	random(n-n)	ok my favorite song is french its name belle of garou
	mBERT$_{CS}$(1-1)	well, my favorite song is french. its name is the belle of garou
	mBERT$_{CS}$(n-n)	well, my favorite song is french called belle of garou

Table 4.9: Examples of translation outputs by different MT systems. CS words that are incorrectly translated are underlined in the translations.

Chapter 5

Generating Code-Switching Sentences using Pretrained Models

Barah Fazili, Shammur Chowdhury, Kenton Murray, Ahmed Ali, Preethi Jyothi

5.1 Introduction

Pretrained multilingual transformers, such as mBERT [114] and XLM- RoBERTa [115], have been shown to enable effective cross-lingual transfer. However, due to interference from other languages and constrained model capacity [116, 117, 118] these models suffer from the "curse of multilinguality", where performance in some languages degrades as the model includes more languages during pretraining. Monolingual models, Roberta and GottBERT, significantly outperform multilingual models including XLM-R for English to German and German to English translation tasks, respectively, despite XLM-R being trained on roughly twice the number of English tokens used to train Roberta. This problem is alleviated for the language pair of English and German by [119] by training a Roberta model that is exclusively trained on the two languages of interest, resulting in better translation performance than its multilingual counterpart. Similarly, [120] studied the performance of multilingual models on Arabic information extraction (IE) tasks, including named entity recognition, part-of-speech tagging, argument role labeling, and relation extraction while pretraining a customized bilingual BERT, dubbed GigaBERT, that is designed specifically for Arabic NLP and English-to-Arabic zero-shot transfer learning.

Motivated by these observations from prior work on the benefits of limiting pretraining to two languages (say L_1 and L_2) for better transfer to downstream tasks involving the same language pair, we hypothesize that such a bilingually pretrained model could also be leveraged for generating code-mixed text involving both L_1 and L_2.

5.2 Methodology

We sample from pretrained models to generate code-switched text using the following steps. First, we start with a pretrained multilingual model that is further predominantly exposed to monolingual text in the component languages of a code-switched language. Next, this pretrained bilingual model is further finetuned on a relatively smaller amount of real code-switched text. From the resulting fine-tuned model, we sample code-switched text.

In our experiments, we focus on the Arabic-English language pair. We start with the Arabic-English version of BiBERT [119] which is a bilingual encoder consisting of 24 Transformer layers, trained using 9.2B words of Arabic text and 26.8B words of English text. This bilingual encoder is further exposed to a limited amount of Arabic-English code-switched text using the Masked Language Modeling (MLM) pretraining objective. (BiBERT is a variant of the multilingual XLMR model further pretrained on a large amount of Arabic and English text.) For Arabic-English code-switched text generation, we finetuned the BiBERT model on a small set of real code-switched text of 2.5K sentences taken from ArZEn [121] and mgb3-cs [122] for Egyptian Arabic.

Bidirectional encoders like BERT are mostly used as parameter initialization for finetuning on downstream tasks. Using such models in a generative manner is not as straightforward. [123] presented a viable approach for generating text from BERT based on Gibbs sampling that does not require any additional parameters or training. They formulate BERT as a Markov Random Field Language Model (MRF-LM). The idea is to begin with a seed sentence taken from a corpus in the target domain, or initialize with a sequence of only [MASK] tokens. Masking and subsequently substituting the sampled token corresponding to masked position

$t\epsilon\{1, ..., T\}$ can generate a sentence of length T tokens in T timesteps. We adopt a sequential sampling scheme for code-switched generation with our finetuned Arabic-English BiBERT model that is detailed below.

5.2.1 Sequential Sampling for Code-Switched text Generation

For deriving code-switched text using the finetuned BiBERT model, we feed monolingual Arabic sentences as input and mask a random word within these sentences. While decoding at a position corresponding to a masked token, we add a simple constraint of sampling only English tokens from the predicted distribution. One token is decoded at a time and then substituted in the previous input for decoding the next token. We make multiple passes through the model until all the tokens are predicted for the chosen word. Say the sentence in monolingual Arabic, X, comprises M words and equivalently N subword tokens depending on the tokenizer: $X = w_1, w_2, \ldots, w_M = x_1, x_2, \ldots, x_N$. For masking we choose a word position k uniformly at random, and let the word w_k span token positions i to j. Each token in $[x_i, \ldots, x_j]$ is masked and replaced using greedy sampling on the predicted distribution corresponding to its position in the sequence. This is done sequentially over the span of the entire word. Expressions 1 and 2 illustrate the input sequence at the start and at time step k respectively where $\tilde{x}$ represents the token substitutions from previous time steps. X_k corresponds to the sequence after k token substitutions are made (where $k \in \{1, j - i + 1\}$) via sequential sampling. Figure 5.1 further illustrates this sequential sampling.

$$X_0 = x_1, x_2, ..., x_{i-1}, [\text{MASK}], ..., [\text{MASK}], x_{j+1}, ...x_N \tag{1}$$

$$X_k = x_1, x_2, ..., x_{i-1}, \tilde{x}_i, ..., \tilde{x}_{i+k-1}, [\text{MASK}], ..., [\text{MASK}], x_{j+1}, ...x_N \tag{2}$$

We note here that such a sequential sampling generates code-switched sentences that are not necessarily semantic equivalents of the monolingual inputs. The model can introduce any English word in the masked span that is appropriate given the context. This leverages the power of the underlying pretrained model to insert appropriate English words within monolingual inputs.

5.2.2 Experimental Results and Discussion

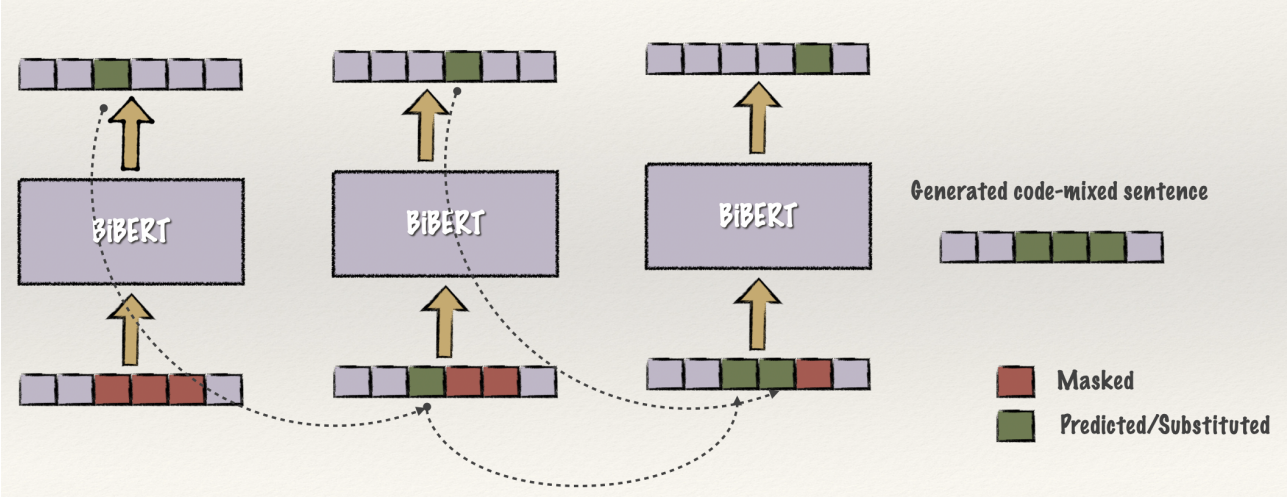


Figure 5.1: Sequential sampling for Code-mixed text generation.

Figure 5.2 shows two examples of monolingual Arabic inputs and the corresponding code-switched Arabic-English (Ar-En) outputs that were generated with our method. A striking feature about these generations in contrast to other generation methods is that the replacements are not necessarily the translation of the original text. Sampled and substituted English words can be anything that makes sense to the model based on the bidirectional context. In Example 1, the replacement conflicts with the original semantics but the rendered code switched sentence is no less acceptable. This shows the potential of using such pretrained models for generating very diverse code-switched outputs compared to the traditional methods where we mostly rely on the translations or a bilingual dictionary.

Table 5.2 shows some preliminary results where we trained n-gram models on synthetic text generated from our model and evaluated the perplexity on real code-switched Ar-En text (ESCWA). We get large improvements

Example 1	
Original (Monolingual)	أعمل إيه بقى يا شوكت أنا مش عايزة وجع دماغ [What do I do Shawkat, I **don't** want problems]
Synthetic (Code-switched)	أعمل إيه بقى يا شوكت أنا **genuinely** عايزة وجع دماغ [What do I do Shawkat, I **genuinely** want problems]
Example 2	
Original (Monolingual)	أوكي أوعدك فهميني بدل ما أنا خلاص قربت أتجنن [Okay, I promise, help me understand, as **that's it**, I'm about to go crazy]
Synthetic (Code-switched)	أوكي أوعدك فهميني بدل ما أنا قربت أتجنن **offended** [Okay, I promise, help me understand, as I'm **offended**, I'm about to go crazy]

Figure 5.2: Sample generations

Training corpora	**Perplexity**
Baseline (transcription)	7081
Monolingual English + Monolingual Arabic + BiBERT with 5% sampling	3706
Monolingual English + Monolingual Arabic + BiBERT with 10% sampling	4475

Table 5.1: Perplexity over real Ar-En code-switched text.

with respect to the baseline text drawn from the speech transcriptions.

Another evaluation is with rescoring hybrid ASR lattices using 4-gram LMs trained on monolingual text augmented with the synthetic text generated using our model. The word error rates also reduce significantly with using our generation text.

Over the test corpus that we used to compute the perplexities, we also computed the average cross-entropy across the bigrams for all four permutations of languages in the bigram. Figure 5.3 shows how the average values for the 4-gram model trained on the BiBERT generations are found to be lower overall.

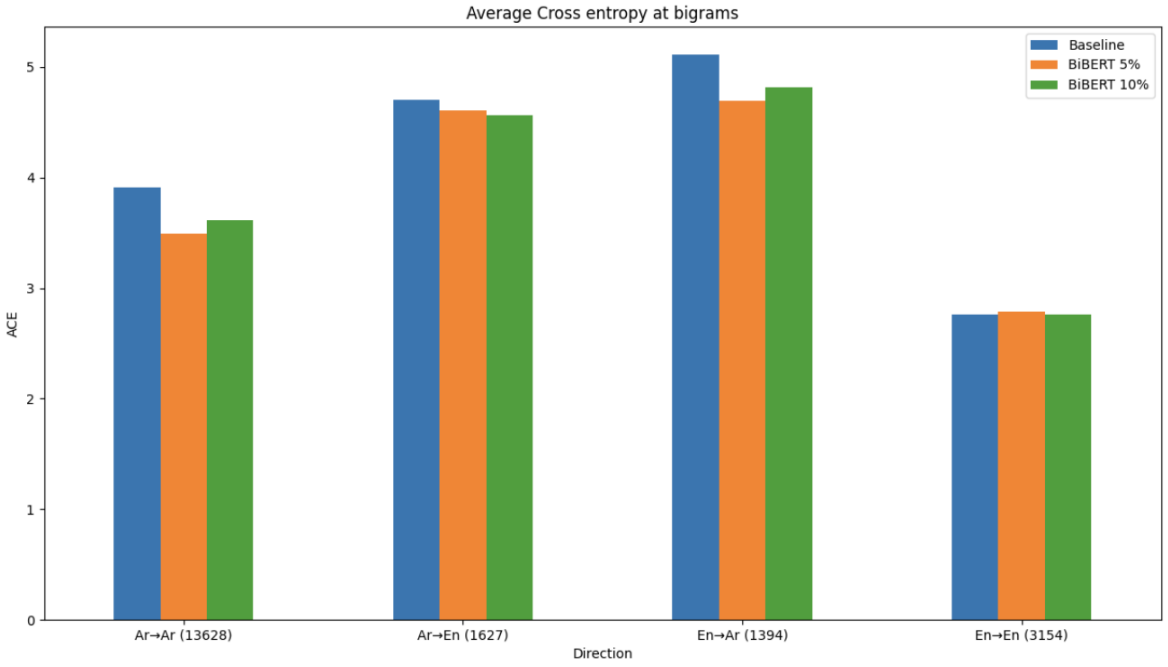


Figure 5.3: Average cross-entropy on real code-switched Ar-En corpus for 4-gram language models trained on BiBERT generations with $\sim 5\%$ and $\sim 10\%$ of English words in each sentence.

Training corpora	**WER**
Baseline (transcription)	49.25
Monolingual English + Monolingual Arabic + BiBERT with 5% sampling	48.07
Monolingual English + Monolingual Arabic + BiBERT with 10% sampling	48.52

Table 5.2: ASR Word Error Rates

5.3 Conclusion and Future Work

We proposed a simple method for generation of code-switched text using sequential sampling leveraging pretrained bidirectional bilingual encoders finetuned on a small amount of code-switched text. The generations offer much more variety than traditional approaches owing to the vast pretraining knowledge of large multilingual and bilingual models. As future work, we plan to expand from word-level substitutions to phrase-level substitutions guided by linguistic theories. Also, we aim to explore constrained decoding at the entire sequence level for code-switched sentence generation. Systematic human evaluations on the generated text would also be performed. Lastly, we hope to extend this technique to generate code-switched text in other language-pairs.

Chapter 6

Unsupervised Code-switched text generation from parallel text

Jie Chi, Brian Lu, Peter Bell, Preethi Jyothi, Ahmed Ali

Abstract

There has been great interest in developing automatic speech recognition (ASR) systems that can handle code-switched (CS) speech to meet the needs of a growing bilingual population. However, existing datasets are limited in size. It is expensive and difficult to collect real transcribed spoken CS data due to the challenges of finding and identifying CS data in the wild. As a result, many attempts have been made to generate synthetic CS data. Existing methods either require the existence of CS data during training, or are driven by linguistic knowledge. We introduce a novel approach of forcing a multilingual MT system that was trained on non-CS data to generate CS translations. Comparing against two prior methods, we show that simply leveraging the shared representations of two languages (Mandarin and English) yields better CS text generation and, ultimately, better CS ASR.

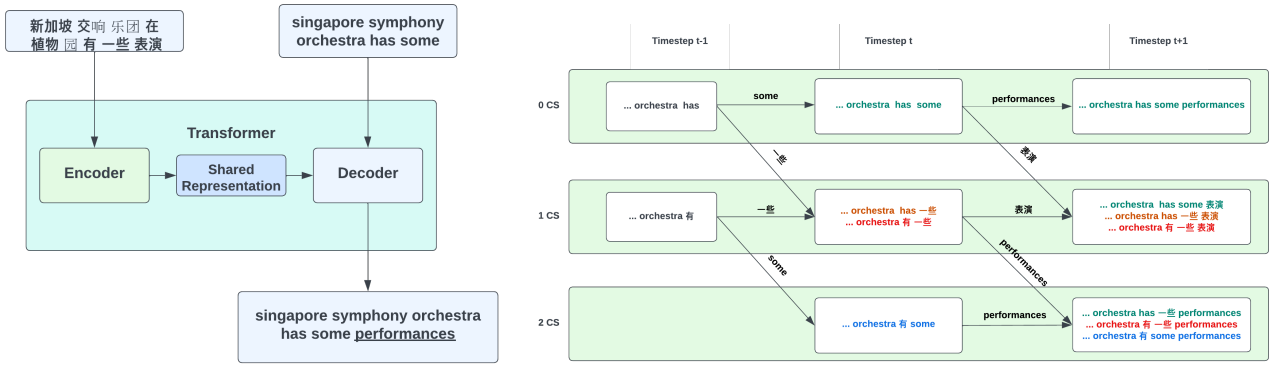
6.1 Introduction

Code-switching (CS) refers to a common phenomenon whereby speakers shift between languages during conversation, especially in multilingual areas, for example, in India and Singapore. Linguists categorize code-switched language according to where the switching occurs [124]: at sentence or clause boundaries (inter-sentential CS), within a sentence or clause (intra-sentential CS, also known as code-mixing), or by inserting a tag phrase (tag switching). There has been great interest in developing ASR systems for such settings to meet the demand of a growing bilingual population. However, the lack of CS data is a major hindrance to these efforts. This motivates techniques for generating synthetic code-switched sentences, which can be used to augment text [125] or speech [126] training data.

In contrast to existing methods of code-switched text generation that either rely on real code-switched training text or commit to specific linguistic theories [127], we propose a novel method. We pretrain a Transformer encoder-decoder model on parallel text without any real code-switched data, and then force the decoder to switch languages a given number of times. Our method simply leverages the shared representations induced by pretraining a multilingual translation model. Our experiments focus on intra-sentential CS and demonstrate improvements against two standard methods on the SEAME corpus [128].

6.2 Related work

There have been many efforts to construct synthetic data to augment the small existing datasets of code-switched text. As parallel monolingual texts are far more plentiful (or can be created by machine translation), a popular research direction is to align parallel sentences and mix them under the guidance of linguistic theories of code-switching, such as Functional Head Constraints or Equivalence Constraints [129, 127, 130]. These rule-based methods can only extract and concatenate monolingual fragments from the parallel texts. Another line of work directly generates synthetic code-switched sentences from a language model or conditional language model that has been trained using a small amount of code-switched data. This work variously uses RNN models [131, 132, 133, 134], Pointer-Generator networks [125], GANs [135, 136, 137, 138], and VAEs [139].



(a) 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

(b) 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.

Figure 6.1: Model training and decoding

Similar to the linguistic rule-based approaches, our proposed method requires only parallel monolingual text. However, our method uses a sequence-to-sequence model, which enjoys the same flexibility as prior methods that take the conditional language modeling approach. When a small amount of code-switched text is available, we can use it to additionally fine-tune our model—which improves the quality of the generation to be on par with or better than other supervised methods.

6.3 Methodology

6.3.1 Parallel Text Pretraining

Word embeddings play an integral role in modern Speech and NLP models. Questions about the degree to which the embedding spaces of different languages share a similar structure have received much interest. Early work in the cross-lingual word embedding literature showed that separately learned, non-contextual word embeddings of different languages can be aligned via linear mappings [140, 141]. In the case of contextual word embeddings, similar alignment results (using more sophisticated mappings like centered kernel alignment) have been reported for separate monolingual BERTs as well as for a multilingual BERT [142, 143].

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 machine translation 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 [144] 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, via grid beam search [145]. Finally, we evaluate the utility of our synthetic CS sentences by using them to train a n-gram language model to use in a downstream ASR system.

6.3.2 Translation Model

We use a Transformer encoder-decoder architecture [146] (Figure 6.1a), with a vocabulary that is the disjoint union of the vocabularies of the two languages of interest.[1] These two vocabularies are harvested from the two sides of the parallel training corpus. 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 MT systems that translate between many language pairs using the same parameters [147], 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.

[1]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.

6.3.3 Grid Beam Search

We follow [145] to constrain the decoding by the number of code-switching points. Different from regular beam search, the set of prefixes in the beam is partitioned into subsets according to the value of some feature (for us, the number of code-switching points). Each subset is separately pruned back to its top-k prefixes. Extending a prefix may change its feature value (for us, if the extension creates a new switching point), in which case the extension will fall into a different subset. At the end, the algorithm returns the top elements of each subset—in our case, the best outputs with 0, 1, 2, ... code-switching points. Figure 6.1b illustrates with a simplified example.

6.3.4 Other Approaches

To compare to prior methods for generating CS text [127, 125], we also implement models based on Equivalence Constraint Theory (ECT) and on Pointer-Generator Networks (PGN). Both of these models depend on parallel data. The PGN additionally requires CS 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 [127], 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 [148]. In contrast to the EC baseline in [125], 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 [127] and use the alignment together with the constituency parses to determine if a substitution is acceptable.

Pointer-Generator Networks require supervised training. The input is the *concatenation* of the parallel sentences x and y and the output is the desired code-switched sentence. We re-implement the model introduced by [125], except that we do not use part-of-speech tags as additional input features.

6.4 Experimental setup

6.4.1 ASR Framework

We begin by pretraining an acoustic model on the union of the training portions of **TED-LIUM 3** [149], an English speech corpus collected from TED talks, and **AISHELL-1** [150], a Mandarin speech corpus consisting of over 170 hours of speech. We then fine-tune these acoustic models on the *monolingual* utterances from **SEAME** [128], a Mandarin-English code-switching speech corpus collected in conversations and interviews from Malaysian and Singapore bilingual speakers. SEAME labels each utterance as English, Mandarin, or CS. As our acoustic model is not trained on any of the CS utterances, it is typical of those used in existing multilingual ASR systems.

Below, we try combining this hybrid acoustic model with language models trained on CS text, where we experiment with different ways of obtaining CS text. We compare the performance of the resulting ASR systems on the held-out portion of SEAME (11852 utterances),[2] which contains both monolingual (5384) and code-switched (6468) spoken utterances.

6.4.2 Real CS Text

SEAME contains around 50K spoken utterances for training that are labeled as CS. 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.

SEAME includes an additional 11K spoken utterances held out for evaluation: around 5K are monolingual and 6.5K are CS. We use this held-out set of real utterances as our test set.

6.4.3 Parallel Non-CS Text

For each CS 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).

[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 CS 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 (§6.3.2) on the 50K (x, y) pairs and then use grid beam search (§6.3.3) to decode 3 CS translations of each x and each y. Specifically, for each $c \in [1, 3]$, 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** (§6.3.4) on the (x, y) pairs. Hybridizing each pair in all legal ways yields about 12 CS 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.[4] We refer to this model as **CST** (code-switched translation) and use it to retranslate each x and each y to 2 new CS utterances (totaling 4), which we then subsample to 3.

As our supervised baseline, we train **PGN** (§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 CS utterances.

Note that we used only the CS portion of SEAME, not the monolingual portion, to generate our synthetic CS utterances. This ensured a (unrealistically good) match with the topics and lengths of the held-out CS utterances.[5]

6.4.5 Model Architectures and Training

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 Pointer-Generator Network following [125] using a one-layer Bi-LSTM encoder and a one-layer LSTM decoder with hidden dimension 256.[6]

We tokenize all Mandarin parts of the data using JieBa[7] 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.

Language Models

For each dataset described in §6.4.2, §6.4.3, and §6.4.4, we use the SRILM toolkit to train a trigram model with Kneser-Ney smoothing [152]. For each *generated* dataset in §6.4.3, and §6.4.4, we additionally train a trigram model by *combining* it with real code-switched data (§6.4.2) via LM interpolation. This leads to improvements in WER and PPL discussed in §6.5. Interpolation weights are optimized on the monolingual SEAME data, disjoint from both SEAME code-switched training and test set.

Acoustic Model

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 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 CS 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 [133, 153].

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 [154]. In the end, we have 180K entries in the lexicon and any uncovered words are treated as UNK.[8] Compared with training two monolingual models using the same architecture, the performance of the obtained bilingual ASR model only drops by 0.5 absolute word error rate.

[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.

[5] The CS sentences in SEAME differ significantly in length from the monolingual sentences. In general, longer sentences are more likely to code-switch [151].

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

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

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

Table 6.1: 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 §6.4.5

			Supervised		Unsupervised		
		RealCS (50K)	CST (150K)	PGN(150K)	CT(150K)	ECT(150K)	Non CS(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

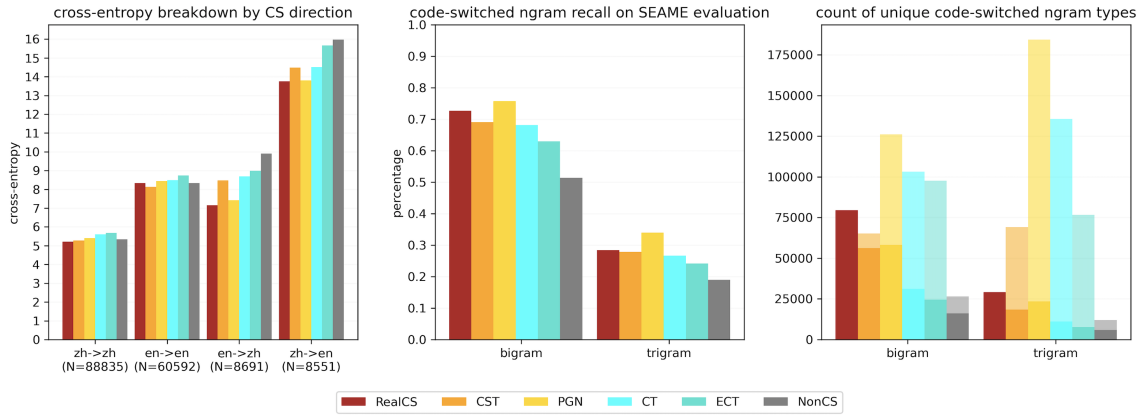


Figure 6.2: Cross-entropy breakdown and n-gram coverage. Left: Cross-entropy breakdown by the transition of language IDs. Middle: *Token*-level code-switched bigram and trigram recall on the SEAME evaluation set. Right: *Type*-level code-switched bigram and trigram counts. Darker bars count the number of shared n-gram types between a particular dataset and the SEAME training data.

6.5 Results and Discussion

6.5.1 ASR Results

Table 6.1 presents word error rate (WER) of our ASR system on SEAME test set. The ASR system uses the pretrained acoustic model with trigram LMs trained on synthetic CS 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 CS and monolingual utterances. Among supervised methods, **CST**'s superior performance compared to **PGN** on the monolingual subset and worse performance on the CS 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.

6.5.2 Language Modeling Results

We also evaluate the various LMs on the text transcripts of SEAME test set, ignoring the audio. The perplexity (PPL) results are in Table 6.1, with a cross-entropy breakdown in Figure 6.2 (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 CS data are less surprised.

Figure 6.2 (Mid) shows the percentage of the CS bi/trigrams contained in the test set that appear in the synthetic texts.[9] Figure 6.2 (Right) shows the total counts of unique CS bi/trigrams in the synthetic text (lighter bars) as well as the number of those bi/trigrams that also appear in the real CS training data (darker bars).[10] Unsurprisingly, **RealCS** and supervised methods generate data with much better n-gram coverage.

[9] A truly Non CS corpus would contain no code-switching bigrams and trigrams. However they exist in our Non CS dataset because our Non CS 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*.

[10] Recal that the training data of the various synthetic generations methods were derived from SEAME CS training set.

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 CS Text

The code-switched sentences generated by our **CT** model are not always perfect translations of the input, but are they reasonable CS text? Most of the code-switching consists of lexical substitutions (e.g. "you go to take 营销(marketing) loh you are the 最好的(best)"). We find that the resulting sentences are mostly understandable, but errors occur (e.g. "it s fun to 火车(train, the noun) with them"), and they don't always code-switch in the same places that a bilingual speaker would (e.g. "my dad is the one who 给(give) him the 工作(job)"). Some sentences could code-switch *too often* because **CT** required them to do so.

6.5.4 Limitations

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 chapter, which generally perform better when limited data is available [153], it is interesting to investigate if the same finding still holds in an End-to-End framework. We plan to use the synthetic CS text for LM rescoring in an End-to-End framework as in [155] or LM decoding for RNN-T [156] in the future.

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 CS 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 the data it generates does not outperform methods that use real CS 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 CS 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 CS generation.

Part III

Automatic Speech Recognition for Code-Switched Speech

Building robust speech recognition systems capable of dealing with multilingual and code-switching speech poses several challenges. The fact that code-switching is an unwritten phenomenon that rarely occurs in most text and transcribed speech corpora makes it challenging to build speech recognition system using real data.

We address here the following three research topics: (*i*) the lack of multilingual and code-switching speech data, (*ii*) the challenge of building code-switching ASR system using monolingual data and (*iii*) using self-supervised models for Code-Switching ASR as follows:

Getting access to large amounts of speech code-switching data. We introduce a speech collage, a framework for generating synthetic code-switched data given two monolingual corpora, and code-switching text. We show that this synthetic data benefits both the encoder and decoder in an end-to-end system, and has significant gains over training with only monolingual data, and also over a shallow integration of a code-switched language model. This is the first work applying this framework of unit-selection for generating code-switched data. Our results suggest that speech collage is efficient for code-switching data augmentation and it benefits from smoothing and energy normalization.

Building robust speech recognition system for code-switching using monolingual data. We build effective code-switched ASR under a zero-shot setting where no transcribed code-switching speech data is available for training. We propose to use multiple monolingual modules to transcribe all speech segments indiscriminately with a monolingual script (i.e. transliteration). This passes the responsibility of code-switching-point detection to subsequent bilingual modules which determine the final output by considering multiple monolingual transliterations along with external language model information. We demonstrate the effectiveness of our transliteration-based method using Conditional CTC models deployed for zero-shot Mandarin-English code-switching ASR.

Using self-supervised models for Code-Switching ASR. We investigate using untranscribed speech data to improve the code-switching ASR system using a small amount of manually transcribed speech data, along with weak language. We simulated low-resource settings on English, with a strong language model. Our results show significant gain on South African code-switching languages. We obtained a substantial improvement by using this model as a seed model for semi-supervised training.

Chapter 7

Speech Collage: Code-Switched Audio Generation Using Monolingual Corpora

Dorsa Zeinali, Amir Hussein, Matthew Wiesner, Ondrej Klejch
Shammur Chowdhury, Ahmed Ali

Abstract

Code-switching is a linguistic phenomenon that occurs commonly in dialectal and informal conversations, but is poorly represented in most speech corpora, making training of ASR models for code-switched speech difficult. To address this training data scarcity we propose a method, called *Speech Collage*, producing code-switched transcribed speech training data from purely monolingual speech corpora and code-switched text for training code-switched speech recognition systems. Audio examples for code-switched text are constructed by splicing together existing examples of words drawn from monolingual corpora based on different criteria. We examine the impact of the generated speech quality on speech recognition in Mandarin-English. We show that this technique beneficial in end-to-end ASR systems and narrows the performance gap between models trained using monolingual and natural code-switched speech and models trained using monolingual and synthetic code-switched speech. However, models trained on natural speech still achieve better results.

7.1 Introduction

As speech technologies have improved, they have been applied to increasingly more difficult domains [157], [158]. One domain that remains challenging even for state-of-the-art speech processing is code-switched speech recognition. Intra-sentential code-switching is one such challenging multilingual scenario in which speakers use multiple languages within the same sentence [159]. However, while this style of speech is exceedingly common in multilingual societies [160], and serves as an important role in allowing people to express themselves more clearly [161], it is often an unwritten phenomenon that rarely occurs in most text and transcribed speech corpora.

This is a unique low-resource scenario as there are much fewer examples of code-switched transcribed speech than monolingual transcribed speech. Code-switching is also prevalent in many low-resource languages which makes creating language technologies for them much more difficult. To be able to build technologies for code-switching without large amounts of transcribed code-switched speech, we need to train using monolingual data. However, training a model using only monolingual data is particularly problematic for end-to-end models. E2E models learn an implicit language model that characterizes the training data [162], and the syntactical structure of code-switched speech is inherently different than monolingual speech. As a result, the end-to-end models will produce only monolingual output even when transcribing limited code-switched speech [163].

To alleviate this problem, we propose a method for using monolingual data to construct synthetic code-switched audio examples using our method, *Speech Collage*. Our work focuses on applying a relaxation of unit selection speech synthesis given only monolingual data, i.e. splicing, for generating synthetic training examples to be used along with monolingual data during training. We test our proposed method on the SEAME Mandarin-English corpus, and we show that training both attentional encoder-decoder and CTC models with monolingual data and our synthetically generated code-switched speech outperforms training with only monolingual data. To test whether the same improvement can be gained with just the addition of unpaired code-switched text, we

show that the same improvement cannnot be gained with a shallow fusion [164] of a language model trained on code-switched text.

7.2 Related Work And Motivation

There has been a large body of previous work on generating synthetic data to train machine learning models. Specifically, there has been much work done in using TTS systems to generate synthetic audio data to train ASR models or to improve ASR model performance [165, 166, 167, 168]. [165] used an existing TTS system for a higher resource language, Hindi, to augment the training data for a low resource indian language with OOV syllabels. This work improved performance on the low resource language. [167] showed that generating synthetic data using TTS systems with diverse speaker representations helped ASR performance.

However, using TTS systems to generate data has limitations. The lack of speaker variability and a lack of training data have led to an interest in adapting the unit selection speech synthesis framework for data augmentation. These methods, which we will call splicing, take segments of audios from existing data or databases and concatenate them together based on different criteria. [169] explores using splicing to adapt a baseline ASR model for a new domain. By using real text from the desired domain, the authors sample random audio segments corresponding to words in the text and concatenate them together to generate new audio examples. This method gave them a relative 81.7% reduction in WER compared to the baseline on the new domain. [170] explores generating training examples using splicing on the fly. Given an original utterance and its corresponding audio, random tokens from the original utterance are selected and either guided by a language model or randomly replaced with other tokens. Audio segments corresponding to these new tokens are then taken and spliced in with the original audio, to create a new training example. Data augmentation via splicing has also been used in the field of speech translation [171, 172]. [172] creates new pairs of unseen training data by sampling a new transcription from a bank of text segments in one language by translating it, and concatenating segments of existing audio data. This method delivers consistent improvements in BLEU score on multiple language pairs.

Due to the challenges in building ASR models for code-switched data described earlier, there has been work done to test the impact of using TTS for code-switched data augmentation. [173] explores three data augmentation methods for code switched ASR, among which is leveraging multi-lingual TTS systems (Baidu and Unisound) using real code switched texts to create code switched audios. They found that this method had limited impact, possibly due to the lack of variability in the synthetic audios. [174] explores training a code-switched TTS system based on FastSpeech using a code switched corpus, and using this system to generate code switched audio. [175] also explores using TTS as a way of augmenting code switched data, but employs an interpolation technique that mixes up TTS and real speech samples to smooth out some artifacts unique to TTS.

Due to the challenges of TTS being more present in the code-switching domain, splicing techniques have also been used to generate synthetic code switched data. [174] augment Mandarin-English code switched data using an existing code-switched corpus by randomly selecting two utterances from the same speaker, and replacing the English segment of one utterance with that of another. [176] create code switched audio by concatenating segments from monolingual corpora in a way that preserves the probability distributions that govern the transition of phones across languages at code-switch boundaries. In addition, [177] explore concatenating whole utterances of the two target languages to create a code-switched corpus.

Several works also investigate leveraging unpaired text data by using external language models with ASR models [164], [178], [179], [180], [181]. Of these, shallow fusion [164] is used in this work.

7.3 Speech Collage

We propose a framework for splicing together segments of monolingual corpora given a code-switched text example, based on unit-selection speech synthesis.

7.3.1 The Unit Selection Speech Synthesis Framework

Unit selection speech synthesis uses a single speaker dataset of utterances, along with linguistic annotations. Units (phones, syllables, words, etc.) are extracted from this database based on certain criteria and concatenated together to form new utterances, with some post-smoothing. To synthesize a new utterance, a target is constructed, which is based on an input text, with added desired linguistic features. Units are taken from the database which best match the target specifications, and they are picked based on minimizing the sum of two costs. The *join cost*, $C^j(u_{i-1}, u_i)$, estimates how well two consecutive units (u_{i-1} and u_i) will join based on acoustic features. The *target cost*, $C^t(u_i, t_i)$, measures how close the unit u_i is to its target specification t_i. A

search algorithm can be used to find this best matching unit, and the details of the two cost functions, and the search differ between systems [182].

7.3.2 Our Approach Within Unit Selection Speech Synthesis

Our approach relaxes the framework of unit selection speech synthesis in several ways. 1) It aims to create code-switched speech, and as such needs two different corpora. The corpora we used were the monolingual English and Mandarin portions of SEAME [128]. 2) Both portions had multiple speakers, instead of just one. 3) The constructed target was just the input code-switched text, without any further linguistic or acoustic specifications. As such, the target cost was set to zero. Lastly, we set the join cost function, $C^j(u_{i-1}, u_i) = 0$ and removed the searching requirement completely. For each word, we picked a random instance of it from the dataset.

The units we chose were words for English, and characters (syllables) for Mandarin. We used these self-contained units to not increase the degradation of the synthesized audio, as smaller units like phones lead to more flexibility but a lower quality of audio [183], and each unit in our dataset was spoken by multiple speakers.

In this work, we explored whether the speech quality of the synthesized speech had any impact on the recognition performance. To test this, we created three different versions of the synthesized speech with varying degrees of speech quality, achieved by pre or post processing. They are presented here from smoothest quality to most noisy.

Normalized energy

In this version, for each word, we picked a random instance of it from the dataset and concatenated them together to create a synthetic utterance, X'. We then calculated the energy, e, per synthesized utterance.

For a T-length speech feature sequence, $\mathbf{X} = \{\mathbf{x_t} \in \mathbb{R}^D \| t = 1, \cdots, T\}$, the energy value was calculated as follows:

$$e = \frac{1}{T} \sum_{T} \mathbf{x_t}^2$$

To remove artifacts between segments introduced only by changes in energy, which was a heuristic for volume, we normalized the speech feature sequence by the square root of energy value, as post processing:

$$\mathbf{X}' = \{\frac{\mathbf{x'_t}}{\sqrt{e}} \in \mathbb{R}^D | t = 1, \cdots, T\}$$

Random

In this version, for each word in the underlying text, we picked a random instance of it from the dataset, and concatenated them together. An example of this is show in in Figure 1. No pre or post processing was done.

Random with added noise

In this version, for each word in the underlying text, we also randomly picked an instance of it from the dataset, and concatenated them together. In addition to randomly picking, we also randomly sampled a noise recording from the MUSAN dataset [184] and added it to each synthesized utterance as post-processing, to introduce further artifacts.

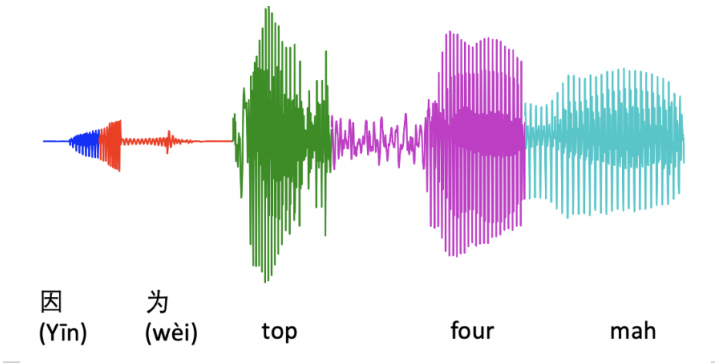


Figure 7.1: Diagram of SpeechCollage

	English-Mandarin	
	CTC WER	Attentional Enc/Dec WER
Monolingual Baseline	42.5	48.4
+ CS language model	41.9	48.2
+ synthetic CS speech (energy)	31.6	31.6
+ synthetic CS speech (random)	31.4	32.2
+ synthetic CS speech (random + noise)	32.4	34.1
CS Topline	19.8	18.9

Table 7.1: WER on a code-switched dev set for CTC and Attentional Enc/Dec models trained on different types of data

7.4 Data and Experimental Setup

7.4.1 Data

We use 29 hours of monolingual Mandarin and English data from the SEAME corpus [128] for training the baseline models. In addition, we generate 45 hours of synthetic code-switched data from the code-switched transcripts of SEAME. We also use the code-switched text of SEAME for training the language model. We use the provided test code-switched data, where Mandarin is the matrix langauge and English is the embedded language.

7.4.2 Experimental Setup

All of our models were trained using the ESPNet toolkit [185]. Our input features were either raw wav form audios, or global mean-normalized 83 log-mel filterbank and pitch features [186]. We apply the Switchboard Strong (SS) augmentation policy of SpecAugment [187]. We combine 2,622 Mandarin characters with 3,000 English BPE [188] units to form the output vocabulary. We trained two types of models. One model was a CTC model where the ENCODERS are conformers [189, 190]. with 12 blocks, kernel size of 15, 2,048 feed-forward dim, 256 attention dim, and 4 heads. This model was trained with the CTC [191] framework, and used greedy decoding. Another was an attentional decoder model, where the DECODERS are transformers [192] with 6 blocks, positional dropout rate of 0.1, dropout rate 0.1, 2048 feed-forward dim, and 4 heads. These models were decoded with the Attention framework with a beam size of 10. We use the Adam optimizer to train 50 epochs with an inverse square root decay schedule, and 25k warmup steps.

In addition to these two types of models, we also trained a language model on purely code-switched text, to be shallowly integrated for decoding. The language model was a transformer with dropout rate of 0.1, feed-forward dim of 2048, embedding dim of 128, 512 attention dim, and 8 heads. We use the Adam optimizer to train 17 epochs with an inverse square root decay schedule, 25k warmup steps, and 100 batch size. When integrating, we used a language model weight of 0.1, with length penalty of 0.8 for the attentional encoder-decoder model, and a language model weight of 0.2 and length penatly of 0.2 for the CTC model.

7.5 Results and Analysis

In Table 1, we compare the difference between the performance of the two different models trained solely on monolingual data, and also models either trained with CS data or decoded with a CS language model.

As shown in the table, training a model using only monolingual data, but shallowly integrating a language model for decoding does not improve performance significantly. This suggests that the real benefit of the synthetic CS data is not from the addition of just CS text via a language model. It suggests that the E2E model is actually learning to code-switch via the synthetic examples.

The gain is further shown by the addition of the 45 hours of synthetic CS training data to the baseline models. All three methods gave significant gains, with the biggest gains from the addition of the synthetic data generated based on energy normalization. This method decreased WER by 25.7% and 34.7% for the CTC and attentional Enc/Dec models respectively. This further suggests that both the encoder and decoder within the E2E model are improving with the synthetic data.

The difference in performance between the three different methods is quite small, which might be due to the fact they were all similarly unnatural, with a lot of artifacts in the data. The randomly generated data with noise only increased WER by 1-2 points, which further supports this. Overall, the addition of the 45 hours of synthetic data improved the baseline significantly.

7.6 Conclusion

We present a framework for generating synthetic code-switched data given two monolingual corpora, and code-switched text. We show that this synthetic data benefits both the encoder and decoder in an End-to-End system, and has significant gain over training with only monolingual data, and also shallow integration of a code-switched language model. This is the first work applying this framework of unit-selection for generating code-switched data.

While we observe many improvements, the gap between the efficacy of real speech and synthetic speech is wide, and future work is needed to close this gap.

Chapter 8

Code-Switched Modeling

Brian Yan, Matthew Wiesner, Ondrej Klejch, Preethi Jyothi, Shinji Watanabe

8.1 Abstract

In this work, we seek to build effective code-switched (CS) automatic speech recognition systems (ASR) under the zero-shot setting where no transcribed CS speech data is available for training. Previously proposed frameworks which conditionally factorize the bilingual task into its constituent monolingual parts are a promising starting point for leveraging monolingual data efficiently. However, these methods require the monolingual modules to perform *language segmentation*. That is, each monolingual module has to simultaneously detect CS points and transcribe speech segments of one language while ignoring those of other languages – not a trivial task. We propose to simplify each monolingual module by allowing them to transcribe all speech segments indiscriminately with a monolingual script (i.e. *transliteration*). This simple modification passes the responsibility of CS point detection to subsequent bilingual modules which determine the final output by considering multiple monolingual transliterations along with external language model information. We apply this transliteration-based approach in an end-to-end differentiable neural network and demonstrate its efficacy for zero-shot CS ASR on Mandarin-English SEAME test sets.

8.2 Introduction

In order to build multilingual automatic speech recognition (ASR) systems that are robust to code-switching (CS), practitioners must tackle both the long-tail of possible language pairs [193] and the relative infrequency of intra-sententially CS examples within collected training corpora [2]. Therefore, a preeminent challenge in the CS ASR field is to build effective systems under the zero-shot setting where no CS ASR training data is available. Recent advancements in multilingual speech recognition have demonstrated the impressive scale of cross-lingual sharing in neural network approaches [194, 195, 196, 197, 198, 199, 200, 201], and these works have shown that jointly modeling ASR with language identity (LID) grants some intra-sentential CS ability [200, 201, 177]. However, most of these large scale models skew towards high-resourced languages [198] and do not seek to directly optimize for intra-sentential CS ASR between particular language pairs.

A more promising direction towards zero-shot CS ASR can be found in prior works which seek to incorporate monolingual data directly to improve CS performance [202, 203, 204, 176, 205, 206, 207, 208, 209, 210, 211]. In particular, there are several works which achieve joint modeling of CS and monolingual ASR by conditionally factorizing the overall bilingual task into monolingual parts [212, 213, 214]. By using label-to-frame synchronization, this *conditionally factorized* approach can make a CS prediction given only the predictions of the monolingual parts [212] – theoretically these conditionally factorized models can model CS ASR without any CS data, but this has not been previously confirmed.

In this work, we seek to build CS ASR systems under two zero-shot data conditions: 1) monolingual speech and CS text data are available, 2) only monolingual speech and text data are available. In particular, we are interested in exploring the zero-shot capability of conditionally factorized joint CS and monolingual ASR models.

We first re-formulate the initial monolingual stage of these conditionally factorized models in terms of their *language segmentation* burden, showing that prior works expect each monolingual module to perform CS point detection and transcription in tandem. Any errors in CS point detection are thus propagated downstream to the final bilingual stage which attempts to stitch multiple monolingual predictions into an output which may or may not be CS. To improve model robustness towards zero-shot CS ASR, we propose an alternative formulation of the monolingual stage such that each module is an indiscriminate *transliterator*, transcribing all speech using a

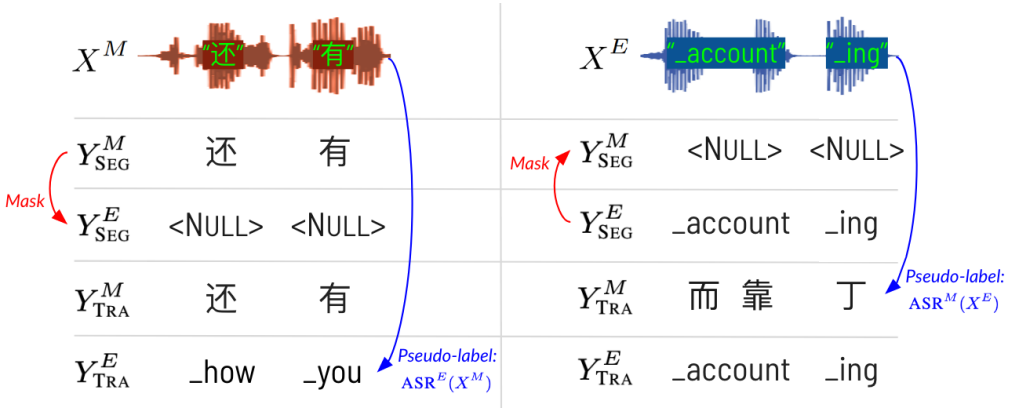


Figure 8.1: Examples showing the difference between language segmentation targets $Y^{M/E}_{\text{SEG}}$ obtained via masking (§8.3.2) vs. transliteration targets $Y^{M/E}_{\text{TRA}}$ obtained via cross-lingual pseudo-labeling (§8.4.1).

monolingual script without any regard for potential CS points. As a result we delay CS point detection until the final bilingual stage, allowing our models to condition this critical decision on multiple monolingual inputs and incorporate additional information from external language models. Our transliteration-based method yielded 5 absolute error-rate reduction in our zero-shot CS ASR experiments.

8.3 Background and Motivation

In this section, we examine the language segmentation role of the monolingual modules in previously proposed conditionally factorized models [212], motivating our transliteration-based approach (§8.4).

8.3.1 Joint Modeling of Code-Switched and Monolingual ASR

Let us take the Mandarin-English bilingual pair as an example for the following formulations. Bilingual ASR, where speech may or may not be CS, is a sequence mapping from a T-length speech feature sequence, $X = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, ..., T\}$, to an L-length label sequence, $Y = \{y_l \in (\mathcal{V}^M \cup \mathcal{V}^E) | l = 1, ..., L\}$ consisting of Mandarin $\mathcal{V}^M$ and English $\mathcal{V}^E$. The conditionally factorized framework [212] decomposes this bilingual task into three sub-tasks: 1) recognizing Mandarin, 2) recognizing English, and 3) composing recognized monolingual segments into a bilingual sequence.

The basis of this approach is to model the label-to-frame alignments. For each T-length observation sequence X and L-length bilingual label sequence Y there are a number of possible T-length label-to-frame sequences $Z = \{z_t \in \mathcal{V}^M \cup \mathcal{V}^E \cup \{\varnothing\} | t = 1 \dots T\}$, where $\varnothing$ denotes a blank symbol as in Connectionist Temporal Classification (CTC) [191] or RNN-T [215]. Further consider that for each bilingual Z there are two corresponding monolingual label-to-frame sequences $Z^M = \{z^M_t \in \mathcal{V}^M \cup \{\varnothing\} | t = 1 \dots T\}$ and $Z^E = \{z^E_t \in \mathcal{V}^E \cup \{\varnothing\} | t = 1 \dots T\}$. The label posterior, $p(Y|X)$, can thus be represented in terms of bilingual, $p(Z|X)$, and monolingual, $p(Z^M|X)$ and $p(Z^E|X)$, label-to-frame posteriors as follows:

$$p(Y|X) = \sum_{Z \in \mathcal{Z}} \sum_{Z^M \in \mathcal{Z}^M} \sum_{Z^E \in \mathcal{Z}^E} p(Z, Z^M, Z^E | X) \tag{8.1}$$

where $\mathcal{Z}$ and $\mathcal{Z}^{M/E}$ denote sets of all possible bilingual and monolingual label-to-frame alignments for a given Y. Eq. (8.1) is the exact *joint* bilingual and monolingual ASR likelihood which can be further factorized using independence assumptions to obtain the form:

$$p(Y|X) \approx \underbrace{\sum_{Z} p(Z|Z^M, Z^E)}_{\text{Bilingual Posterior}} \underbrace{\sum_{Z^M} p(Z^M|X) \sum_{Z^E} p(Z^E|X)}_{\text{Monolingual Posteriors}} \tag{8.2}$$

From Eq. (8.1) to Eq. (8.2), the first assumption is that given Z^M and Z^E, no other information from the observation X is required to determine Z, allowing for conditional modeling of the bilingual posterior $p(Z|Z^M, Z^E, \cancel{X})$ given only monolingual information. The second assumption is that given X, Z^M and Z^E are independent, allowing for separate modeling of monolingual posteriors $p(Z^M|\cancel{Z^E}, X)$ and $p(Z^E|\cancel{Z^M}, X)$. Note we abbreviate this pair of separate monolingual modules as $p(Z^{M/E}|X)$ in future sections.

8.3.2 Modeling $p(Z^{M/E}|X)$ with Language Segmentation

What should be the behavior of the monolingual Mandarin module $p(Z^M|X)$ when encountering a segment of English speech and vice versa? Monolingual modules in prior works [212, 213, 214] determine each label-to-frame alignment $z_t^{M/E}$ by first determining the language identity of each speech frame $\text{LID}(\mathbf{x}_t)$ [216]. If the speech frame $\mathbf{x}_t$ is from a foreign language then the module will ignore it by emitting a special <NULL> token, otherwise it will transcribe using its monolingual vocabulary. This monolingual *language segmentation* decision is defined as follows (shown for Mandarin):

$$z_t^M = \begin{cases} \arg\max_{m \in \mathcal{V}^M \cup \{\varnothing\}} p(z_t^M = m | X, z_{1:t-1}^M) & \text{if } \text{LID}(\mathbf{x}_t) \text{ is } M \\ \arg\max_{m \in \{<\text{NULL}>, \varnothing\}} p(z_t^M = m | X, z_{1:t-1}^M) & \text{if } \text{LID}(\mathbf{x}_t) \text{ is } E \end{cases} \tag{8.3}$$

Note that the frame-wise $\text{LID}(\mathbf{x}_t)$ is not a separate module, but rather an implicit decision within the posterior maximization over the <NULL> augmented monolingual label-to-frame alignments $Z^{M/E} = \{z_t^{M/E} \in \mathcal{V}^{M/E} \cup \{\varnothing, <\text{NULL}>\} | t = 1 \dots T\}$. This language segmentation behavior is learned by optimizing likelihoods of <NULL> masked label targets Y_{SEG}^M and Y_{SEG}^E (e.g. in Figure 8.1).

It follows that the bilingual $p(Z|Z^M, Z^E)$ (Eq. (8.2)) behaves as:

$$z_t = \begin{cases} m & \text{if } m \in \mathcal{V}^M \wedge e = <\text{NULL}> \\ e & \text{if } e \in \mathcal{V}^E \wedge m = <\text{NULL}> \\ b & \text{otherwise} \end{cases} \tag{8.4}$$

where m and e are the arguments maximizing $p(z_t^M | X, z_{1:t-1}^M)$ and $p(z_t^E | X, z_{1:t-1}^E)$ respectively and b is the argument maximizing $p(z_t | Z^M, Z^E, z_{1:t-1})$. If either monolingual module predicts a CS point by emitting <NULL> then the bilingual module defaults to the prediction of the other monolingual module – in other words, the first two cases of Eq. (8.4) expect that the language segmentation in Eq. (8.3) is mistake-free. The third fall-back case is considered for ambiguous language segmentation, such as if m and e are both <NULL> or both non <NULL>. This case-by-case bilingual decision is an adverse design for our zero-shot objective – models are likely to become over-reliant on the first two cases during training. Language segmentation while training on purely monolingual utterances boils down to an over-simplified *utterance-level* language identification task which may not generalize to *intra-sententially* CS test utterances. If CS point detection is expected to be tricky, then a more robust strategy should *always* expect ambiguous monolingual inputs to the final bilingual decision as in the third case of Eq. (8.4).

8.4 Proposed Framework

In this section, we propose to completely remove language segmentation from monolingual modules using a transliteration-based formulation of $p(Z^{M/E}|X)$. We then present a neural model of our modified conditionally factorized approach for zero-shot CS ASR.

8.4.1 Modeling $p(Z^{M/E}|X)$ with Transliteration

Rather than detecting CS points at the monolingual stage in order to know which speech segments to transcribe vs. which to ignore, we propose to simply allow each monolingual module to transcribe everything. This means that for speech of a foreign language the monolingual modules are producing *transliterations*, mapping sounds to phonetically similar units within their monolingual vocabularies $\mathcal{V}^M$ and $\mathcal{V}^E$. In other words, the monolingual modules simplify from Eq. (8.3) to the following form (shown for Mandarin):

$$z_t^M = \underset{m \in \mathcal{V}^M \cup \{\varnothing\}}{\arg\max} \; p(z_t^M = m | X, z_{1:t-1}^M) \tag{8.5}$$

where the speech X may contain any language. This form completely removes any sense of frame-wise language identity $\text{LID}(\mathbf{x}_t)$.

To see why this modification is advantageous for zero-shot CS ASR, consider the corresponding change to the bilingual module:

$$z_t = \underset{b \in \mathcal{V}^M \cup \mathcal{V}^E \cup \{\varnothing\}}{\arg\max} \; p(z_t = b | Z^M, Z^E, z_{1:t-1}) \tag{8.6}$$

Note that this new bilingual form in Eq. (8.6) never defaults to the prediction of one monolingual module as in the first two cases of the previously proposed bilingual form in Eq. (8.4), reducing the risk of propagating errors made in the monolingual stage. In other words, the bilingual decision now determines each z_t by directly considering the conditional likelihood $p(z_t | Z^M, Z^E, z_{1:t-1})$ (Eq. (8.2)). This modification effectively delays CS

point detection from the monolingual stage (where we would have to simultaneously transcribe and perform frame-wise language identification per §8.3.2), to the bilingual stage (where transcription information is already given).

To train monolingual modules to transliterate speech segments of a foreign language, we obtain transliteration targets Y^M_{TRA} and Y^E_{TRA} using cross-lingual pseudo-labeling.[1] For instance, we pass monolingual English speech X^M to a monolingual Mandarin ASR model $\text{ASR}^M(\cdot)$ for inference and vice versa as follows:

$$Y^M_{\text{TRA}} \leftarrow \text{ASR}^M(X^E) \tag{8.7}$$

$$Y^E_{\text{TRA}} \leftarrow \text{ASR}^E(X^M) \tag{8.8}$$

where $\text{ASR}^{M/E}(\cdot)$ denote generic label-to-frame models – if we use the same architecture for pseudo-labeling as we do for our monolingual modules then these transliteration targets are cross-lingual semi-supervisions [218, 219, 220, 221].[2] Swapping the language segmentation targets Y^M_{SEG} and Y^E_{SEG} (§8.3.2) for these transliteration targets Y^M_{TRA} and Y^E_{TRA} is the *only* modification required to realize our desired monolingual and bilingual module behaviors in Eq. (8.5) and (8.6).

8.4.2 Conditional CTC with External LM Architecture

Finally, let us consider how to construct a neural architecture for our modified conditionally factorized framework. Monolingual and bilingual label-to-frame posteriors (§8.3.1) may be modeled using CTC or RNN-T networks as demonstrated by prior works [212, 213, 214]. However for zero-shot CS ASR, the conditional independence assumption of CTC vs. the internal language modeling of RNN-T is a critical difference. A RNN-T based model may require internal language model (LM) adaptation [215, 223, 224] to alleviate monolingual biases while a CTC based model can be directly applied to CS test sets with optional shallow external LM fusion [225].

We therefore model monolingual, $p(Z^M|X)$ and $p(Z^E|X)$, and bilingual likelihoods, $p(Z|Z^M, Z^E)$, using CTC networks, $P_{\text{M\_CTC}}(\cdot)$, $P_{\text{E\_CTC}}(\cdot)$, and $P_{\text{B\_CTC}}(\cdot)$, as follows:

$$P_{\text{M\_CTC}}(z^M_t|X, \cancel{z^M_{1:t-1}}) = \text{SOFTMAXOUT}^M(\mathbf{h}^M_t) \tag{8.9}$$

$$P_{\text{E\_CTC}}(z^E_t|X, \cancel{z^E_{1:t-1}}) = \text{SOFTMAXOUT}^E(\mathbf{h}^E_t) \tag{8.10}$$

$$P_{\text{B\_CTC}}(z_t|\mathbf{h}^M, \mathbf{h}^E, \cancel{z_{1:t-1}}) = \text{SOFTMAXOUT}^B(\mathbf{h}^M_t + \mathbf{h}^E_t) \tag{8.11}$$

where speech encoders, ENCODER^M and ENCODER^E, map the speech signal, X, to latent monolingual representations, $\mathbf{h}^M = \{\mathbf{h}^M_t \in \mathbb{R}^D | t = 1, ..., T\}$ and $\mathbf{h}^E = \{\mathbf{h}^E_t \in \mathbb{R}^D | t = 1, ..., T\}$ followed by softmax normalized linear projections to monolingual or bilingual vocabularies. Then addition fusion yields a bilingual latent representation which is finally fed to the bilingual CTC. These three CTC networks are jointly optimized with an interpolated multi-task objective: $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{B\_CTC}} + (1 - \lambda_1)(\mathcal{L}_{\text{M\_CTC}} + \mathcal{L}_{\text{E\_CTC}})/2$.

During decoding, we first merge all CTC likelihoods, $P_{\text{M\_CTC}}(\cdot)$, $P_{\text{E\_CTC}}(\cdot)$, and $P_{\text{B\_CTC}}(\cdot)$, following the interpolation procedure described in Eq. (6) of [214]; we denote this merged CTC likelihood as $P_{\text{CTC}}(Z|X)$. We then jointly decode $P_{\text{CTC}}(\cdot)$ with an external bilingual LM, $P_{\text{B\_LM}}(Y)$, using the time-synchronous beam search described in [225], which approximates the following decision:

$$\underset{Y \in \{\mathcal{V}^M \cup \mathcal{V}^E\}*}{\arg\max} \lambda_2 (\prod_{Z \in \mathcal{Z}} \log P_{\text{CTC}}(\cdot)) + (1 - \lambda_2) \log P_{\text{B\_LM}}(\cdot) \tag{8.12}$$

where $\{\mathcal{V}^M \cup \mathcal{V}^E\}*$ denotes the set of all possible bilingual outputs.[3] This architecture, which we refer to as Conditional CTC, is depicted by the block-diagram in Figure 8.2. The monolingual modules of these Conditional CTC models can perform either language segmentation (§8.3.2) or transliteration (§8.4.1) depending on which set of monolingual targets (e.g. Figure 8.1) is used during training. For transliteration, we obtain Y^M_{TRA} and Y^E_{TRA} (Eq. (8.7) and (8.8)) by greedily decoding monolingual CTC models (Eq. (8.9) and (8.10)) and then applying repeat and blank removal.

8.5 Data and Experimental Setup

Data: As shown in Table 8.1, we split SEAME [6] training data into CS and monolingual (Mandarin + English) parts to create two zero-shot settings. The first setting allows 99h of monolingual labeled speech data (for ASR

[1] Unlike text-based transliteration [217], pseudo-labeling relies solely on the resources presumed to be available in our zero-shot CS ASR settings.

[2] We can apply transliteration to CS speech by stitching predictions corresponding to forced aligned [222] foreign segments between true native targets.

[3] For language segmentation variants of Conditional CTC, we do not expand hypotheses with the special <NULL> token to avoid corrupt outputs.

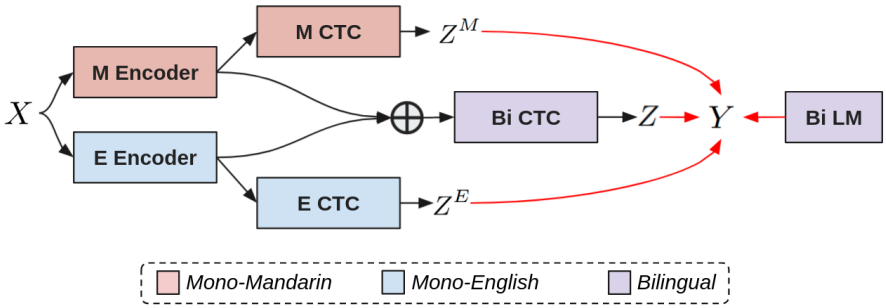


Figure 8.2: Conditional CTC architecture consisting of monolingual and bilingual CTC's plus an external bilingual LM. Red lines indicate joint decoding via time-synchronous beam search.

Table 8.1: SEAME train, devman, and devsge sets broken down by language with hours of duration and number of sentences for speech and text data respectively. †Allowed in fully zero-shot settings. *Original train split [226] was up-sampled by 3x via 0.9 and 1.1 speed perturbations [53].

Set	Type	Full	CS	Mono
TRAIN*	Speech	303h	204h	99h†
TRAIN	Text	89k	50k	39k†
DEVMAN	Speech	8h	6h	2h
DEVSGE	Speech	4h	2h	2h

training) and 89k lines of unpaired CS or monolingual text data (for LM training). The second fully zero-shot setting removes the CS unpaired text data, leaving 39k lines of unpaired monolingual text data. Monolingual CTC's trained on the English and Mandarin only SEAME splits were used for cross-lingual pseudo-labeling §8.4.1.

Models: Models are trained using ESPnet [53]. We apply speed perturbations to up-sample training data by 3x. We combine 4000 Mandarin characters with 4000 English BPE [227] units to form the output vocabulary. Conditional CTC models have two conformer encoders [228, 229] with 12 blocks, 4 heads, 15 kernel size, 2048 feed-forward dim, 256 and attention dim. Vanilla CTC baselines with only one encoder use 512 attention dim, so all models have about 80M parameters. All models are initialized with encoder(s) pre-trained on 150h of Mandarin AISHELL-1 [230] and/or 118h of English TED-LIUM-v1 [231]. We set $\lambda_1 = 0.7$ (§8.4.2) during training for 40 epochs. We set $\lambda_2 = 0.8$ (§8.4.2) during decoding with beam size 10. We use RNN-LMs with 4 layers and 2048 dim trained for 20 epochs.

Evaluation: Systems are evaluated on the full SEAME test sets (devman and devsge) and also scored individually on the CS and monolingual portions of these sets. We measure mixed error-rate (MER) that considers word-level English and character-level Mandarin.

8.6 Results

Table 8.2 presents results in three horizontal partitions where 1) all SEAME training data is allowed 2) CS speech data is removed and 3) CS speech and text data are removed; the latter two settings emulate practical zero-shot scenarios. When CS speech data is available, language segmentation is reliable and thus the transliteration-based method is not necessary (`A2` vs. `A3`). However, once CS speech data is removed the language segmentation approach degrades 13 absolute MER on both full test sets; as a result the transliteration approach outperforms by 5 absolute MER, a wide margin, owing primarily to superior performance on CS utterances (`B2` vs. `B3`). When CS text data is also removed both variants of Conditional CTC degrade only by an additional 2 absolute MER and the gap between remains (`C2` vs. `C3`). In all three data settings both Conditional CTC models outperform Vanilla CTC baselines.

8.6.1 Ablations on the Conditional CTC Model

Our Conditional CTC models consist of three types of modules: monolingual CTC's ($P_{\text{M\_CTC}}$ and $P_{\text{E\_CTC}}$), bilingual CTC ($P_{\text{B\_CTC}}$), and bilingual LM ($P_{\text{B\_LM}}$). In Table 8.3, we examine the relative contributions of these modules by removing each from model `B3` of Table 8.2 during joint decoding (described in §8.4.2). Removing the bilingual LM (line `2`) degrades performance more than removing the bilingual CTC (line `5`), showing the

Table 8.2: Results comparing Conditional CTC models with *transliteration*-based monolingual modules to their *language segmentation* counterparts and Vanilla CTC baselines. The 1st horizontal partition shows top-line results when CS ASR training data is available. The 2nd and 3rd partitions show zero-shot results when only monolingual ASR training data is available. Performances on the full, CS only, and monolingual only splits of the SEAME test sets are measured by % mixed error rate (MER ↓). All models use CTC + LM decoding.

			ASR	LM	DevMan			DevSge		
ID	Model	Monolingual Behavior	Data	Data	Full	CS	M	Full	CS	M
P1	Transducer [209]	*No Monolingual Modules*	CS + M	-	18.5	-	-	26.3	-	-
P2	CTC/Attention [232, 53]	*No Monolingual Modules*	CS + M	-	16.6	-	-	23.3	-	-
A1	Vanilla CTC [225]	*No Monolingual Modules*	CS + M	CS + M	18.8	18.2	21.5	26.2	23.7	29.8
A2	Conditional CTC [213, 214]	Language Segmentation	CS + M	CS + M	**17.1**	**16.5**	19.9	**23.5**	**21.4**	**26.5**
A3	Conditional CTC (Ours)	Transliteration	CS + M	CS + M	17.3	16.9	**19.1**	24.0	22.1	26.7
B1	Vanilla CTC [225]	*No Monolingual Modules*	M	CS + M	36.6	38.9	27.0	42.5	47.0	36.1
B2	Conditional CTC [213, 214]	Language Segmentation	M	CS + M	30.1	32.0	22.0	35.7	39.7	**30.1**
B3	Conditional CTC (Ours)	Transliteration	M	CS + M	**25.2**	**26.0**	**21.9**	**31.0**	**31.5**	30.2
C1	Vanilla CTC [225]	*No Monolingual Modules*	M	M	39.1	41.6	28.4	44.8	50.0	37.3
C2	Conditional CTC [213, 214]	Language Segmentation	M	M	32.2	34.4	23.0	37.8	42.6	31.1
C3	Conditional CTC (Ours)	Transliteration	M	M	**27.3**	**28.5**	**22.6**	**32.7**	**34.0**	**30.8**

Table 8.3: Ablation study examining the relative importance of monolingual CTC, bilingual CTC, and bilingual LM modules during decoding as measured by % mixed error rate (MER ↓) on the devman test set. Bilingual modules are shown in blue and the most severely degraded combination (with no bilingual modules) is **bolded**.

#	Model	Decoding Likelihoods	MER(↓)
1	Cond. CTC w/ Trans.	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$, $P_{\text{B\_CTC}}$, $P_{\text{B\_LM}}$	25.2
2	– Bilingual LM	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$, $P_{\text{B\_CTC}}$	27.4
3	– Monolingual CTCs	$P_{\text{B\_CTC}}$, $P_{\text{B\_LM}}$	25.7
4	– Bilingual LM	$P_{\text{B\_CTC}}$	27.9
5	– Bilingual CTC	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$, $P_{\text{B\_LM}}$	26.0
6	– Bilingual LM	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$	**48.1**

importance of utilizing CS textual data when available. Further, note that monolingual CTCs do contribute (line 3), but are insufficient on their own (line 6). Finally, the fact performance is still reasonable without the bilingual CTC (line 5) suggests that separately trained monolingual CTCs may be directly applied to CS ASR if a CS LM is available – this direction may offer a high degree of scalability towards the long-tail of possible CS pairs and towards CS between three or more languages.

8.6.2 Relaxing the Zero-Shot Setting

How much CS ASR training data do we need for the originally proposed language segmentation method (§8.3.2) to be sufficient? The answer depends on the proximity of the particular language pair and characteristics of the dataset being used, but in our experimental setup we find that the answer is 2h of CS speech data (see Figure 8.3). The decreasing effectiveness of our transliteration method for increasing amounts of CS ASR training data suggests that the cross-lingual pseudo-labels are noisy to a degree. Future investigations into improving pseudo-labeling quality (e.g. via constrained decoding) may benefit this work and other related techniques which employ cross-lingual semi-supervision [218, 219, 220, 221].

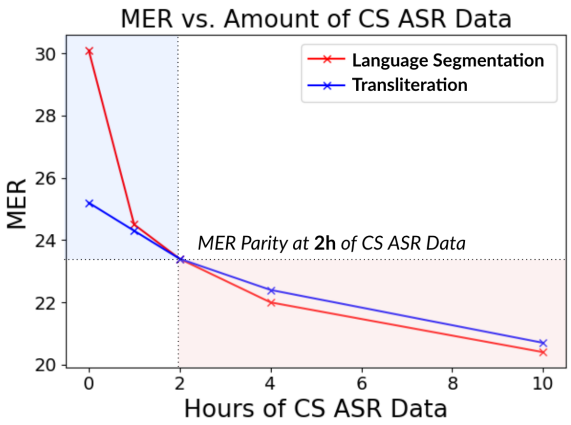


Figure 8.3: Analysis on the amount of CS ASR training data required for conditional CTC with language segmentation to outperform conditional CTC with transliteration. MER(↓) on devman is shown.

8.7 Conclusion

We identify that the promising conditionally factorized joint CS and monolingual ASR framework has an acute weakness which limits its applicability to zero-shot CS ASR; the original formulation expects that each monolingual module can cleanly transcribe native speech while ignoring foreign speech. We propose a simple modification via cross-lingual pseudo-labeling to allow the monolingual modules to instead produce transliterations of foreign speech, thereby avoiding error propagation of frame-wise LID decisions. We demonstrate the effectiveness of our transliteration-based method using Conditional CTC models deployed for zero-shot Mandarin-English CS ASR. In future work, we will extend to other languages, scale beyond bilingualism, and refine our pseudo-labeling technique.

Chapter 9

Using self-supervised models for Code-Switching ASR

Léa-Marie Lam-Yee-Mui, Lucas Ondel, Ondrej Klejch

This chapter investigates the potential of improving a hybrid automatic speech recognition model trained on 10 hours of transcribed data with 200 hours of untranscribed data in low-resource languages. First, we compare baseline methods of cross-lingual transfer with MFCC features and features extracted with the multilingual self-supervised model XLSR-53. Subsequently, we compare two approaches that can leverage the untranscribed data: semi-supervised training with LF-MMI and continued self-supervised pre-training of XLSR-53. Our results on well-resourced English broadcast data derived from MGB show that both methods achieve 18% and 27% relative improvements compared to the baseline, respectively. On the low-resource South African Soap Opera dataset, the relative improvement with semi-supervised training is only 3% due to the inherently weak language model. However, continued pre-training achieves 8.6% relative improvement because it does not rely on any external information.

9.1 Introduction

Automatic speech recognition (ASR) systems have recently demonstrated great accuracy improvements in well-resourced languages [233, 234, 235]. This accuracy has been achieved thanks to the modelling improvements and hundreds of thousands of hours of transcribed speech. However, ASR performance in low-resource languages is still lacking due to limited amounts of transcribed speech for training of acoustic models and limited amounts of text for the training of language models. This lack of training data in low-resource languages is even further exacerbated by code-switching between embedded and matrix languages [236]. In this work, we study how self-supervised training [233, 234, 237] and semi-supervised training [238, 239] can be used to leverage untranscribed audio data to improve the performance of ASR models for underrepresented languages. Consequently, we address the case where only small amounts of manually transcribed speech with an inherently weak language models are available, due to code-switching and the lack of text corpora.

In this chapter, we use the South African Soap Opera dataset [240] which contains 14.3 hours of code-switched speech between four Bantu languages (Sesotho, Setswana, isiXhosa, and isiZulu) and English. Building a good language model for this domain is difficult due to code-switching, variation in orthography and lack of text data on the internet. To improve the ASR performance in these languages, previous works explored transfer learning with a multilingual model [241] and semi-supervised training with a weak language model [242, 243]. Following these works, we focus on building a five-lingual model for the South African languages and on taking advantage of 200 hours of untranscribed data for self-supervised and semi-supervised training. We also run contrast experiments on British English broadcast data from the Multi-Genre Broadcast (MGB) Challenge [244], for which we can train a strong language model using the provided historical BBC subtitles. We artificially mimic low-resource settings by sampling 10 hours of transcribed data and another 200 hours as untranscribed data from the MGB dataset.

In our experiments on the South African Soap Opera dataset and the English MGB dataset, we show that:

- when training the seed five-lingual South African acoustic model, cross-lingual transfer from a matching domain in a well-resource language (MGB) works better than transfer from a mismatched domain in the same languages (NCHLT).
- cross-lingual transfer is comparable with using multi-lingual self-supervised features extracted from XLSR-53.

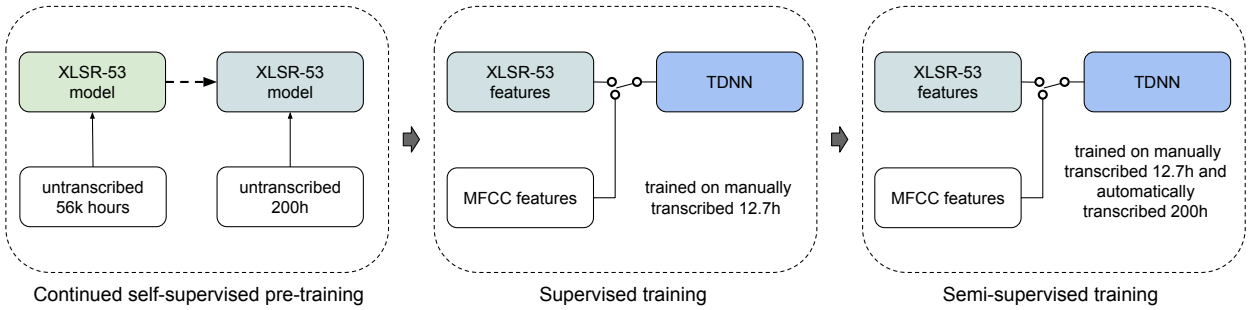


Figure 9.1: Diagram of the combination of self-supervised and semi-supervised trainings for 12.7 hours of transcribed South African data and 200 hours of untranscribed data from other South African soap operas. XLSR-53 is further trained with the 200 hours of untranscribed data (continued self-supervised pre-training) and used as features extractor to train a TDNN. The TDNN model can also be trained with MFCC features. The obtained TDNN model is the seed model for the subsequent semi-supervised training with both transcribed and untranscribed data. We follow the same procedure when running experiments on MGB.

- both semi-supervised training and continued self-supervised pre-training work in well-resourced settings.
- continued self-supervised pre-training works better than semi-supervised training with an inherently weak language model for code-switched speech in South African languages.
- continued self-supervised pre-training and semi-supervised training are complementary even in low-resource languages.

9.2 Related work

In this section, we review some common methods in speech recognition for low-resource languages: cross-lingual transfer, self-supervised pre-training and semi-supervised training.

9.2.1 Cross-Lingual transfer

In cross-lingual transfer, we first train an acoustic model for a set of well-resourced languages and then transfer the parameters of the acoustic model to the new low-resource language. Then, we train the final acoustic model by fine-tuning a subset of the parameters on a small amount of available transcribed data [245, 246]. These multilingual models can also be used to extract bottleneck features which are used to train another model with data from a low-resource language[247].

9.2.2 Self-Supervised Training

In self-supervised learning (SSL), the acoustic model parameters are pre-trained on thousands of hours of untranscribed data. The model learns to recognize latent speech representations from raw signal with a constrastive loss, such as InfoNCE [248]. Architectures based on convolutional layers and Transformers [249] have been proposed and pre-trained with English datasets, such as wav2vec2.0 [233] and HuBERT [250].

For speech recognition, these pre-trained self-supervised models are usually fine-tuned on the transcribed training data with a standard supervised loss such as Connectionist Temporal Classification (CTC) loss [191], or lattice-free maximum mutual information (LF-MMI) loss [58, 251]. A multilingual pre-trained model XLSR-53 [234], which is based on the wav2vec2.0 architecture [233], performs well when used to train ASR models for low-resource languages [234, 252, 253, 254]. However, these large pre-trained models can still suffer from mismatch between training and testing conditions. Therefore, continued pre-training with untranscribed data from the target domain can be used to alleviate the domain mismatch [237, 255]. This procedure of continual pre-training [256] alleviates the domain mismatch while requiring orders of magnitude less data by retaining the knowledge from the original dataset, which makes this approach especially useful for low-resource languages.

9.2.3 Semi-Supervised Training

In semi-supervised training, we start by training a seed model on the transcribed training data. Then, we use this acoustic model with a language model to produce pseudo-labels for the untranscribed data. These pseudo-labels are then used as training targets for fine-tuning the acoustic model [238]. Traditional approaches for semi-supervised training use one-best transcripts as pseudo-labels. However, these transcripts might contain

transcription errors negatively affecting the acoustic model. Therefore, it is necessary to perform some form of confidence filtering to discard utterances with noisy transcripts [257]. Another option is to use lattices as pseudo-labels [239, 258]. This approach circumvents the issue of erroneous one-best transcript by representing possible alternative transcriptions and their corresponding uncertainties within the lattice. Popular way of performing lattice-based semi-supervised training is to use lattice-free maximum mutual information criterion [58, 239]. The limiting factor of semi-supervised training is the quality of the language model used to produce the pseudo-labels [259]. A good language model typically needs to be trained on large amounts of text data; in our experience, at least several hundred million words are needed to train a strong general language model. Unfortunately, such large amounts of text are not available online for many low-resource languages [260]. Furthermore, the language modelling in these low-resource languages can further be exacerbated by the presence of code-switching [261]. The performance of semi-supervised training can be improved by starting from a stronger seed acoustic model, trained with cross-lingual transfer [262] or fine-tuned from a self-supervised model [263].

9.3 Improving the Acoustic Model with Untranscribed Data

The performance of ASR systems in low-resource languages is limited by the small amount of manually transcribed data. In this section, we describe how we collected untranscribed speech and how we used it to improved the performance of ASR systems using self-supervised training and semi-supervised training. Our approach consisted of three steps. First, we used continued self-supervised pre-training of XLSR-53 on our untranscribed speech data. Then, we used the adapted XLSR-53 model to extract features for training of the seed model. Finally, we used this seed model to produce better pseudo-labels for semi-supervised training on the untranscribed speech data. The whole pipeline is illustrated in Figure 9.1.

9.3.1 Datasets

We conducted experiments on the South African Soap Operas dataset [240]. This dataset contains 14.3 hours of transcribed code-switched speech with people alternating between four Banto languages and English. We used the official training (12.7 hours) and test splits (1.3 hours). The represented pairs of languages are: English-Sesotho (eng-sot), English-Setswana (eng-tsn), English-isiXhosa (eng-xho) and English-isiZulu (eng-zul). We used the training transcripts (155k tokens) to train a LM. Note that we tried to train a stronger language model using text crawled from internet, but the resulting model was worse than the one trained only on the training transcripts.

We also collected untranscribed speech for these South African languages. To avoid noisy collected data for South African languages, we identified South African soap operas on Wikipedia and we downloaded trailers for these soap operas. This method ensured that the crawled data was in-domain and contained relevant languages together with other South African languages used in the soap operas. In total we were able to collect 200 hours of raw recordings which were segmented with WebRTC VAD[1] for further processing.

As a constrast, we also experimented with the BBC broadcasts from the MGB dataset [244]. We selected 10 hours from the MGB dataset for the training dataset and used another 200 hours from the MGB dataset as untranscribed data. We trained a language model on the provided BBC subtitles (650M words). We evaluated the performance of the models trained with the official MGB development set using the manual segmentation.

9.3.2 ASR model training

We trained a five-lingual (four Bantu languages + English) South African acoustic models which used either 40-dimensional MFCC features or 1024-dimensional XLSR-53 features as inputs. Both types of models were trained using Kaldi toolkit [264] and used the same alignments obtained with a standard GMM model. We used a CNN-TDNN architecture with 16.7M parameters for the acoustic model using MFCC features and a TDNN-F architecture with 23.7M parameters for the acoustic model using XLSR-53 features. We did not use convolutional layers with the XLSR-53 features because the XLSR-53 model acts as a convolutional front-end. Both types of models were trained with the LF-MMI criterion [58] for six epochs on speed-perturbed training data. The pronunciation dictionaries for the four South African languages and English were built using the NCHLT dictionaries and corresponding grapheme-to-phoneme rules [265]. The final merged dictionary had 88 phones, including 12 phones shared by all five languages. The vocabulary size was 18.8k tokens. Furthermore, we trained a 3-gram language model on the available training transcripts.

To improve the performance of the model using MFCC features, we transferred parameters from a model trained either on the NCHLT dataset [265], which contains read speech from 11 official South African languages, or on the 200 hours of English broadcasts from the MGB dataset [244]. To deal with the mismatch in acoustic

[1] `https://github.com/wiseman/py-webrtcvad`

units, we replaced the final layer of the pre-trained model and retrained the whole acoustic model. Based on our experience with fine-tuning of hybrid models, we used a 10-times smaller learning rate than during training from scratch, for all layers except for the newly initialized final layer.

The acoustic model for MGB has the same CNN-TDNN or TDNN-F architecture depending on the input features. The language model for MGB was trained on all available MGB subtitles making it a very strong language model.

9.3.3 Continued self-supervised pre-training

We used the collected 200 hours of South African speech untranscribed data or the 200 hours of MGB for self-supervised training. However, since training the self-supervised models from scratch is computationally expensive and requires a lot of data, which might not be available for low-resource languages, we only performed continued pre-training of a self-supervised model using these 200 hours with the contrastive loss of wav2vec2.0 [233]. As pre-trained model, we chose to use the multilingual model XLSR-53, which is based on wav2vec2.0 LARGE architecture and is trained on 53 languages using a total of 56k hours of training data. Instead of using the pre-trained model directly as an acoustic model, we used it as a multilingual bottleneck feature extractor. We experimented with various pre-trained self-supervised models and extracted the features from different layers and we found that the last layer of XLSR-53 worked best in our scenario.[2] We extracted these representations from the last layer with the S3PRL toolkit [266] and used them as inputs for a standard hybrid TDNN-F model [267]. We performed the continued self-supervised pre-training of the XLSR-53 model with the fairseq toolkit [268] using the 200 hours of untranscribed data. We kept the hyperparameters identical to the ones used to train wav2vec2.0 LARGE [233] on Librivox. We continued pre-training for 8k iterations equivalent to 67 epochs, with a batch size of at most 1.4M tokens and we used gradient accumulation to simulate training on a bigger batch size using only two Tesla V100 GPUs.

9.3.4 Semi-supervised training

In semi-supervised training, we use the seed acoustic model trained on the manually transcribed data together with a language model trained on available text data to produce pseudo-labels for the untranscribed speech. The semi-supervised training was done using the lattice-free maximum mutual information (LF-MMI) training criterion [58] and followed the semi-supervised training approach proposed in [239].

In semi-supervised training, we use the seed acoustic model trained on the manually transcribed data together with a language model trained on available text data to produce pseudo-labels for the untranscribed speech. The semi-supervised training was done using the lattice-free maximum mutual information (LF-MMI) training criterion [58] and followed the semi-supervised training approach proposed in [239]. We performed semi-supervised training on a combination of the manually transcribed training data and the untranscribed 200 hours. We decoded the untranscribed data with the seed model trained on the transcribed data and we used the decoded lattices as pseudo-labels for semi-supervised training. Note that, we filtered the untranscribed data with the minimum mean recording confidence threshold of 0.8 and the minimum speaking rate threshold 1.25 words per second prior to the semi-supervised training as suggested in [269]. We trained the semi-supervised models for six epochs.

Table 9.1: Word Error Rate (WER) on the test set of the South African Soap Operas dataset and the development set of MGB. For the South African Soap Operas, the results are split by language pairs: English-Sesotho (eng-sot), English-Setswana (eng-tsn), English-isiXhosa (eng-xho) and English-isiZulu (eng-zul).

		South African Soap Operas					**MGB**
		eng-sot	eng-tsn	eng-xho	eng-zul	all	dev
(1)	CNN-TDNN baseline	55.9	46.6	63.9	56.6	54.7	29.4
(2)	with cross-lingual transfer from NCHLT	52.8	44.5	60.4	54.1	52.0	-
(3)	with cross-lingual transfer from MGB	50.1	43.8	60.8	52.8	50.8	-
(4)	+ semi-supervised training	**48.3**	**42.6**	**59.2**	**51.1**	**49.3**	**24.0**
(5)	TDNN-F using XLSR-53 bottleneck features	49.8	45.1	61.7	51.6	50.9	25.4
(6)	+ continued self-supervised pre-training	45.5	40.9	**56.2**	47.6	46.5	21.2
(7)	+ semi-supervised-training	**44.3**	**38.5**	56.5	**46.4**	**45.2**	**19.6**

[2]Note that XLS-R [252] achieved better results than XLSR-53 in our preliminary experiments. However due to its size we were not able to continue pre-training it and therefore we left it for future work.

9.4 Results

9.4.1 Baselines acoustic models

In the first set of experiments, we assessed how well cross-lingual transfer works for South African languages. We compared training the acoustic model from scratch, denoted as (1) in Table 9.1, using cross-lingual transfer from a model trained on the NCHLT dataset (2) and using cross-lingual transfer from a model trained on the MGB challenge dataset (3). Our results demonstrated that a domain-match and data diversity in the MGB dataset (3) is more important than training on additional data for the target languages (2) with overall word error rates (WER) of 50.8% and 52.0% respectively. This is because the NCHLT data contains clean read speech, which is acoustically very different from the speech in the Soap Operas dataset. In addition to the noticeable background noise in the Soap Operas dataset, the speech characteristics are also very different, for example the average speaking rate in the NCHLT dataset is three times slower than in the Soap Operas dataset. Subsequently, we compared the model trained with cross-lingual transfer from MGB (3) with a model using the multilingual self-supervised representations obtained with XLSR-53 as input features (5). We found that these two approaches achieved similar WER of 50.8% and 50.9%. The benefit of (3) is that it is trained on very well matched data and (5) benefits from being pre-trained on large amounts of multilingual data. We hypothesize that (3) would improve even further if cross-lingual transfer was done from a model trained on a diverse multilingual dataset, not only on British English dataset.

9.4.2 Semi-supervised vs self-supervised training

In the next set of experiments, we investigated how 200 hours of untranscribed data can help improve the performance of the initial acoustic model. We observed that semi-supervised training with the seed models using MFCC features (3) in Table 9.1 performed worse than continued self-supervised pre-training of the multilingual model XLSR-53 (6). It is worth noting that it was crucial to combine the transcribed 10 hours of data with the 200 hours of untranscribed data to make semi-supervised training work on the South African dataset. Without the transcribed data the semi-supervised model was worse than the seed model. When we combined the continued self-supervised pre-training with semi-supervised training we achieved further gains (7). The WER of for the Soap Operas dataset is reduced by 11% relative compared to the baseline model trained with cross-lingual transfer from MGB $(3 \rightarrow 7)$ while the WER of the MGB dataset has a higher relative 33% gain compared to the baseline trained with MFCC from a flat-start initialization $(1 \rightarrow 7)$, thanks to the stronger language model. Also note that the comparison between semi-supervised training and continued self-supervised pre-training are not completely fair, because XLSR-53 is a much bigger model with another architecture than our CNN-TDNN acoustic model. We plan to conduct a fair comparison in future.

To overcome the code-switched aspect of the Soap Operas dataset, we tried using a language model with South African monolingual texts crawled from the web and MGB [244] transcriptions but the resulting WER were worse than using the in-domain language model trained only on the Soap Operas transcriptions. This shows that the language model domain match is very important, especially since this in-domain language follows clear transcription conventions consistent with the test transcription. This explains why the in-domain language model is stronger because not all South African languages have standardised orthography and the crawled online texts might follow different spelling rules, which negatively affects the WER. Code-switching is also a mostly unwritten phenomenon, which makes language modelling even more difficult. Furthermore, the performance of the model could be improved by fine-tuning the model for each language pair individually, but we chose not to do it since we were interested in building a unified five-lingual model. Previous works on this South African data indeed demonstrated that the semi-supervised training batch by batch yields improvement over training in a single pass, as does training bilingual acoustic models instead of a five-lingual model [242]. Improvement also comes from adding generated texts [270] and automatic transcriptions to build a strong LM. However, this type of bilingual training with a strong LM is less practical because it requires to obtain untranscribed data with a specific type of code-switching to train the model.

9.5 Conclusions

In this chapter, we explored using untranscribed speech data to improve the accuracy of the speech recognition, using small amount of manually transcribed speech data. We added a constraint of using a weak language model coming from the difficulty of finding accurate code-switched training texts to build a stronger language model for South African languages. We also evaluated our approach on a simulated low-resource settings on English, with a strong language model. We found that the best approach to improve the initial acoustic models is using features from the multilingual XLSR-53 model with continued self-supervised pre-training with the unstranscribed data, which does not require any language model. This approach is particularly useful in cases like South African code-switching, where we can only train a weak language model due to the lack of sufficient amount of in-domain text. When we subsequently used this model as a seed model for semi-supervised training, we obtained a relative improvement of 11% relative compared to the baseline model trained with cross-lingual transfer from MGB for the Soap Operas dataset while for English, we obtained a relative gain of 33% compared to the baseline trained with MFCC. The improvements on the Soap Operas dataset were much smaller than we would expect with a language model with external language resources. For this reason, in the future, we would like to explore ways of training better language models in low-resource and especially code-switched settings to improve the performance during semi-supervised training and decoding.

Finally, we would also like to follow [237] and use better regularization methods during continued self-supervised pre-training. Both these methods should allow for a more efficient continued pre-training. We believe that our findings will be applicable to other low-resource languages with limited amounts of text corpora available.

Part IV

Evaluating Code-Switching ASR

In order to improve a speech recognition system, we should be able to evaluate the accuracy and report error rate using robust and fair evaluation metrics. The current word error rate (WER) evaluation method assumes a single reference transcription for each speech segments; however, in code-switching language pairs with different scripts, there is a tendency to cross-transcribe words, which creates a large number in both hypothesis and reference. Furthermore, we may expect partial (cross-script) transcription, the English word *artificial* can be transliterated to آرتفيشيال in Arabic script (transliteration: *Artfy$yAl*). In some cases, ASR systems can recognize the speech as آرت*ficial*. Thanks to recent end-to-end ASR techniques such as word-pieces. In this section, we have the following contributions:

1. We define the first Human Acceptability Corpus for code-switching (HAC), a reference for benchmarking speech recognition hypotheses with human judgments.
2. We evaluate many metrics in correlation with human judgments.
3. We release the first corpus for human acceptance of code-switching speech recognition results in dialectal Arabic/English conversation speech.

Our experiments suggest the following:

- WER and CER are not adequate for evaluating code-switching speech recognition systems.
- The highest correlation to human judgment is achieved using **transliteration** followed by **text normalization**.

Chapter 10

Benchmarking Evaluation Metrics for Code-Switching Automatic Speech Recognition

Injy Hamed, Amir Hussein, Oumnia Chellah, Shammur Chowdhury
Sunayana Sitaram, Nizar Habash, Ahmed Ali

Abstract

Code-switching poses a number of challenges and opportunities for multilingual automatic speech recognition. In this chapter, we focus on the question of robust and fair evaluation metrics. To that end, we develop a reference benchmark data set of code-switching speech recognition hypotheses with human judgments. We define clear guidelines for minimal editing of automatic hypotheses. We validate the guidelines using 4-way inter-annotator agreement. We evaluate a large number of metrics in terms of correlation with human judgments. The metrics we consider vary in terms of representation (orthographic, phonological, semantic), directness (intrinsic vs extrinsic), granularity (e.g. word, character), and similarity computation method. The highest correlation to human judgment is achieved using transliteration followed by text normalization. We release the first corpus for human acceptance of code-switching speech recognition results in dialectal Arabic/English conversation speech.

10.1 Introduction

Code-switching (CS) is the act of using more than one language within the same discourse. The prevalence of CS across multi-cultural and multi-lingual societies has been met with a growing interest in the NLP and speech processing fields. Automatic Speech Recognition (ASR) for CS is a well studied problem [271] conducted in several language pairs on acoustic modeling, language modeling and novel system architectures.

In code-switched language pairs with different scripts, there is a tendency to cross-transcribe words that creates a large number of homophones in the data, leading to challenges in training and evaluation. Rendering the CS transcription accurately is important, however, is often not straight-forward, and can be inconsistent, leading to the same word being transcribed using different scripts [272]. Cross-transcription is particularly challenging in languages having a high amount of loanwords, as it is not always clear which language a word belongs to, and hence, which script to transcribe it in.

Investigating the ideal ASR performance metric for CS, [272] propose a modified WER metric based on mapping both languages into the pronunciation space, which leads to improvements in accuracy and evaluation. [273] propose a transliteration-based WER metric by transliterating mixed-script utterances into a single script. The authors demonstrate the robustness of the proposed approach on several Indic languages. However, these techniques can also lead to false positives when there are words in the two languages that sound the same but have a different meaning. It also has limitations in cases where different parts of the word are written in different scripts. e.g., the word آرت**ficial**, which is **artificial** in English script or آرتفيشيال in Arabic script. Thanks to word-pieces [274], we may expect partial (cross-script) transcription in CS ASR results.

Researchers have also investigated techniques for handling the problem of non-standardized orthography for Dialectal Arabic ASR evaluation. [275] proposed the use of Multi-Reference Word Error Rate to allow for a wider coverage of different spelling variants for Dialectal Arabic ASR evaluation. Whereas, [276] investigated approaches to reduce spelling variations, which included normalization and following the Conventional

Orthography for Dialectal Arabic (CODA) [277] guidelines to spell words, as well as relying on morphologically abstracted forms obtained from different tokenization schemes and lemmatization.

This work builds upon previous contributions from [273] and [272]. In this study, we investigate various methods that go beyond orthographic transliteration methods. We explore lexical and phonetic representations for evaluation, and study various weighted edit-distance methods and semantic similarity evaluation. We designed a guideline and developed a reference human acceptability corpus (HAC) – quantifying the human judgments in terms of minimal editing of ASR hypothesis. We evaluate a large number of metrics in correlation with human judgments. We believe that this is the first study on human acceptability for CS speech recognition. The contributions of this chapter are as follows:

- We design and develop the first corpus for human acceptance for CS speech recognition. The corpus and its guidelines are publicly available.[1]
- We introduce phone similarity edit distance (PSD) and show its correlation with human acceptance.
- We propose a novel approach to use machine translation on hypotheses and references and report results using semantic evaluation to overcome the cross-transcription challenge.

10.2 HAC: Human Acceptability Corpus for Code-Switching

For developing the Human Acceptability Corpus for Code-switching (HAC), we use a subset from the ArzEn Egyptian Arabic-English CS conversational speech corpus [81] and obtain the hypotheses using different ASR systems trained on publicly available corpora. We carefully design the human annotation task to indicate the amount of post-editing effort needed to correct the hypotheses. While previous work has relied on human judgments in the form of systems' ranking [276, 278], we opt for a more fine-grained human evaluation, where each hypothesis is evaluated independently in terms of the number of edits performed by the annotators. Throughout the chapter, we will refer to the annotators' post-edited text as 'minimal edits annotations' and to the original ArzEn transcriptions as 'references'.

10.2.1 Annotation Guidelines

The annotators were provided with an audio file for each utterance and its hypothesis. The references were not provided to avoid biasing the annotations. The annotators were asked to perform minimal edits to make the hypotheses acceptable, obeying the following rules:

- **Rule of Script Segregation:** Words should be written in Arabic or Roman script, and not a mix of the two. The only exception is the writing of Arabic affixes and clitics in conjunction with English words. Arabic words should be in Arabic script. English words can be in Arabic or Roman script (default is Roman). For missing words, the default script of the language should be used. English origin words that have been integrated in Arabic templatic morphology, and phonology will be treated as Arabic words. These guidelines differ from the guidelines used by [81] for collecting ArzEn transcriptions, where the transcribers were suggested to use Arabic script for Arabic words and Roman script for English words.
- **Rule of Acceptable Readability:** The transcript should be made readable enough to allow someone to reproduce the original audio and intended meaning.
- **Rule of Minimal Edit:** Spelling variations are acceptable as long as they do not break the rule of acceptable readability. This rule covers the cases where certain letters can be used interchangeably, such as ت-ث, ق-ئ, ي-ى, أ-إ-ا, and ة-ه. Non-verbal speech effects should not be added if absent from the ASR output. If they are included and in the audio, they should be accepted. If they are not in the audio, they should be deleted.

In Table 10.1, we provide an example demonstrating the minimal edits annotation. While the CER and WER for the hypothesis and reference are 50.0% and 78.6%, the error rates are dropped to 21.6% and 46.2% when comparing minimal edits annotation and hypothesis, showing the high amount of characters and words mispenalized by CER and WER.

10.2.2 Design Consideration

We sampled two hours of speech from ArzEn training set, consisting of seven recordings and covering a total of 1,304 utterances. For each utterance, we obtained the ASR hypotheses from three different ASR systems, resulting in a total of 3,903 hypotheses to be annotated for minimal correction.[2] These hypotheses were

[1] http://arzen.camel-lab.com/

[2] Three utterances were excluded from the dataset as they only contain non-speech tags, resulting in 1,301 utterances, for each of which we have three ASR hypotheses.

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]
R	ال	weekends	mainly	family	فينزور	ال	two	families	و او	لو	عندنا	تمارين	برضه
H	الويك	أند	زمايلي	فاعملي	فا بالنسور			families	واو	لولا	عندنا	تمارين	بوردو
ME	الويك	أند	ماينلي	فاميلي	فينزور	ال	two	families	واو	لولا	عندنا	تمارين	بردو
R_BW	Al	weekends	mainly	family	fbnzwr	Al	two	families	w Aw	lw	EndnA	tmAryn	brDh
H_BW	Alwyk	>nd	zmAyly	fAEmly	fA bAlnswr			families	wAw	lwlA	EndnA	tmAryn	bwrdw
ME-BW	Alwyk	>nd	mAynly	fAmyly	fbnzwr	Al	two	families	wAw	lwlA	EndnA	tmAryn	brdw
A-Decisions	A	A	S	S	S, D	-	I	-	A	A	-	-	S
CER(ME,H)	**0**	**0**	**33.3**	**33**	83.3	100	100	0	**0**	**0**	0	0	**25**
WER(ME,H)	**0**	**0**	100	100	200	100	100	0	**0**	**0**	0	0	100
CER(R,H)	**150**	**100**	**100**	**100**	83.3	100	100	0	**25**	**100**	0	0	**75**
WER(R,H)	**100**	**100**	100	100	200	100	100	0	**100**	**100**	0	0	100

Table 10.1: This example shows a hypothesis (H), the minimal edits annotation (ME), the ArzEn transcription reference (R), and their Buckwalter transliterations [279] (R_BW, H_BW, and ME_BW). We denote annotation decisions (A-Decision) as substitution (S), insertion (I), deletion (D), or acceptance (A) for words having different forms across H and ME. In this example, CER and WER reductions are seen for minimal edits annotations over reference due to cross-transcription (segments 1, 2, 3, and 4) and unstandardized orthography (segments 9 and 13) issues.

	A2	**A3**	**A4**	**H**	**R**
A1	*8.3* / 16.7	*6.3* / 15.0	8.3 / 18.4	*14.9* / 27.8	*19.9* / 44.0
A2		*8.9* / 17.6	*10.8* / 20.7	*14.1* / 24.6	*20.5* / 44.9
A3			*6.9* / 15.8	*15.1* / 27.9	*18.6* / 41.4
A4				*16.0* / 29.0	*20.5* / 44.9
Avg			*8.2* / 17.4	*15.0* / 27.3	*19.9* / 43.8

Table 10.2: Inter-annotator agreement, showing *CER* / WER between annotators' minimal edits, as well as each annotator with ASR hypotheses (H) and ArzEn references (R).

annotated by four Arabic-English bilingual speakers. We use three pretrained bilingual (Arabic-English) ASR systems, to generate the hypotheses. The systems vary either in architecture or in the decoding parameters, as described below.

HMM-DNN: We used a grapheme-based model trained using a Time Delay Neural Network (TDNN) [57] with the LF-MMI objective [58]. For training the model, we used the alignments from the context-dependent Gaussian Mixture Model and Hidden Markov model (GMM-HMM) system. For decoding, we opt for the 4-gram model, trained on the ASR transcription.

End-to-End ASR: For the end-to-end (E2E) ASR [274] system we used a transformer based architecture [249], comprised of two sub-networks: conformer encoders and the transformer decoders [280]. The ASR system consists of 12 encoder layers and 6 decoder layers, each with 2,048 encoder/decoder units from the feed-forward layers, and 8 attention heads with 512 transformation dimensions. Note that E2E model uses word-piece byte-pair-encoding (BPE) [281], with size of $\sim 10k$. For the study, we adopted two variations of this model, by changing the size of the beam search in the decoding space. The variants are: *(1)* Conformer-Accurate (*Conformer-A*) – with a beam size of 60 and *(2)* Conformer-Fast (*Conformer-F*) with a beam size of 2.

10.2.3 Inter-annotator Agreement

We randomly sampled 203 sentences, annotated by the four annotators, for inter-annotator agreement (IAA).[3] The IAA evaluation is presented in Table 10.2, where we report the CER/WER between every two annotators. As shown, on average, the CER and WER measured between annotators are 8.2% and 17.4% respectively. While these figures are relatively high, it is not surprising, and reflects the complexity of the task, where word acceptability and choice may differ across annotators due to unstandardized orthography. We present the CER/WER between minimal edits annotations and hypotheses reflecting the amount of edits performed by the annotators. Moreover, we show the CER/WER between minimal edits annotations and references reflecting the amount of characters/words that would be mispenalized when evaluating minimal edits annotations against

[3]Two sentences were excluded from the IAA calculations as they were annotated as unclear.

	[1]	[2]	[3]	[4]	[5]	[6]	CER	WER
R	i	have to	say	اخر	سفرية	لي		
H	أي	هفتو	ساي	آخر	سفرية	ليا	53.8	85.7
Transliteration > Arabic							**CER**	**WER**
R_trAr	ي	هافي تو	ساي	اخر	سفرية	لي		
H_trAr	أي	هفتو	ساي	آخر	سفرية	ليا	23.1	71.4
Transliteration > English							**CER**	**WER**
R_trEn	i	have to	say	acher	Safariyah	li		
H_trEn	ai	Haveto	Sai	acher	Safariyah	lia	18.8	71.4
Phonetic Representation							**PER**	**PSD**
R_phone	aj	hæv tə	sej	axr	sfrit	li		
H_phone	ai	hftu	sai	ăxr	sfrit	lia	40.0	30.1

Table 10.3: Example for a reference (R) and hypothesis (H), showing their corresponding transliteration output to Arabic (R_trAr and H_trAr) and Roman (R_trEn and H_trEn) scripts, as well as their IPA phone mapping (R_phone and H_phone) using Epitran. In this example, we separate the phones at word boundaries for better readability, however, the spaces are not included in PER and PSD calculations.

references using CER/WER. These numbers are also a good indicator of the limitation of CER and WER as accurate evaluation metrics.

10.3 Metrics Under Evaluation

Using the developed corpus, we assess multiple evaluation metrics against the ground truth error, measured in terms of human post-editing effort. The metrics we consider vary in terms of representation (orthographic, phonological, semantic), directness (intrinsic vs extrinsic), granularity (word vs character), and similarity computation method.

10.3.1 Orthographic Metrics

In the orthographic space, we investigate the use of four performance measures: WER, CER, Match Error Rate (MER), and Word Information Lost (WIL) [282].

$$WER = \frac{S + D + I}{H + S + D} \tag{10.1}$$

$$MER = \frac{S + D + I}{H + S + D + I} \tag{10.2}$$

$$WIL = 1 - \frac{H^2}{(H + S + D)(H + S + I)} \tag{10.3}$$

where H, S, D and I correspond to the number of word hits, substitutions, deletions and insertions. MER computes the probability of a given match between reference and hypothesis being incorrect. By including S, D and I in the denominator, the range of MER has the bound [0,1]. WIL, introduced in [283], is an approximate measure reflecting the proportion of words lost between hypothesis and reference. While these measures vary in granularity, they all fall short against the cross-transcription issue. Therefore, following the work of [272], we investigate the effect of transliterating hypotheses and references into primary as well as secondary language scripts on alleviating the cross-transcription problem. We perform automatic transliteration using the transliteration API provided by QCRI [284][4]. In Table 10.3, we present an example showing the reduction achieved in CER and WER by mapping the texts into one common script, handling both cross-transcription (columns 1-3) and orthography unstandardization (column 4) challenges. One drawback of this technique though is its dependence on the availability of a language-specific transliteration system, and that its effectiveness is tied to the performance of that system. Following the work of [276], in order to reduce the spelling variation resulting from dialectal Arabic unstandardized orthography, we investigate applying Alif/Ya normalization as a variant for the experiments relying on orthographic evaluation metrics.

[4] `https://transliterate.qcri.org/api`

	Text	Sim
R	كان تعليمي لغاية الجامعة كان في **ناشيونال سكولز** عادي وكان كلهم عربي	
H	كان تعليمي لغاية الجامعة كان في **national schools** عادي و كان كلهم عربي	0.906
$\mathbf{R}_{En}$	My education until university was in normal **national schools**, and they were all Arab	
$\mathbf{H}_{En}$	My education until university was in a normal **National School**, and they were all Arab	0.942

Table 10.4: Example presenting a reference (R) and hypothesis (H) along with their translations (R_{En}) and (H_{En}), showing how the cross-transcription issue was resolved through translation. The cosine similarities (Sim) between R-H and R_{En}-H_{En} are shown, where the embeddings are obtained from mBERT.

10.3.2 Phonological Metrics

The orthographic-based metrics only consider literal correction, and do not adequately handle orthographic unstandardization. To overcome such limitation, we propose phone similarity edit distance (PSD),[5] where we measure the edit distance between hypotheses and references in the shared phone space using International Phonetic Alphabet (IPA) mapping. The main advantage of this approach over the transliteration is that the IPA mapping is model independent and deterministic based on the grapheme to phoneme (G2P) dictionary. The IPA uses standardized representation of speech sounds across different languages. When calculating PSD, we scale the substitution cost by the dissimilarity between the phones based on the articulation features. As a result, PSD adds partial substitution penalty when the phones are different but close in pronunciation. We use the Epitran toolkit [285] that contains massive multilingual G2P including Arabic (ara-Arab) and English (eng-Latn) languages. PSD is calculated as:

$$PSD = \frac{w_S \sum S_i + w_D \sum D_j + w_I \sum I_k}{N} \tag{10.4}$$

where S, D, I are number of phones substitutions, deletions, and insertions, respectively; N is the number of phones in the reference; $S_i = 1 - sim(x_i, y_i)$ where x and y are the aligned phones. The w_S, w_D, w_I are the costs for substitution, deletion and insertion respectively with value of 1 by default. We investigate different values for w_S $(1, 2, 4, 8)$, while keeping $w_D = 1, w_I = 1$. In Table 10.3, it can be seen that our PSD implementation can easily handle code switching. However, the main shortcoming of PSD approach is that it does not consider the semantics of the words.

10.3.3 Semantic Metrics

Next, we assess the ASR output considering semantic similarity. Following [286], we measure the semantic similarity between reference and hypothesis pairs as the cosine similarity between their embeddings obtained from pretrained transformer models. The embeddings are generated using mean pooling over token embeddings.[6] While this approach has been previously investigated for monolingual ASR evaluation [286, 278], it has not been investigated in the scope of CS.

We also introduce a novel pipeline for semantic-based ASR evaluation, where we translate the hypotheses and references into monolingual sentences using Google Translate API[7]. The translations, as well as the original reference-hypothesis pairs, are then evaluated in terms of the following machine translation (MT) evaluation metrics: BLEU [107], chrF [108], and BertScore (F1) [109], in addition to cosine similarity. We explore translating the sentences into the primary language (Arabic), secondary language (English), as well as a completely independent language (Japanese), to investigate the ability of the approach to generalize on different languages. For calculating cosine similarity, we explore the use of several pretrained models, including mBERT and language-specific BERT models, for obtaining the sentence embeddings. For the original texts containing CS, we use mBERT [287] and BiBERT [288]. For Arabic translations, we use mBERT, CAMeLBERT [289], and AraBERT [290]. For English translations, we use mBERT and BERT-base [287]. For Japanese translations, we use mBERT, BERT-base-japanese[8], and BERT-large-japanese[9].[10]

[5] `https://github.com/JSALT2022CodeSwitchingASR/Evaluation`

[6] We also investigate the use of CLS token, however the correlations are significantly lower.

[7] `https://cloud.google.com/translate`

[8] `https://huggingface.co/cl-tohoku/bert-base-japanese`

[9] `https://huggingface.co/cl-tohoku/bert-large-japanese`

[10] We only present the results for the best settings. We show the results with lowercasing and performing Alif/Ya normalization, which have shown to improve correlations. We do not present results for chrF++ as it gave slightly lower correlations compared to chrF. For monolingual English translations, BertScore(F1) using mBERT gave higher correlations than BertScore(F1) using roberta-large. For cosine similarity, using bert-base-multilingual-cased has also shown to give overall higher correlations over bert-base-multilingual-uncased.

Orthographic Metrics (Error Measures)					
	Base	**Translit>Ar**	**Translit>En**	**Base +ArNorm**	**Translit>Ar +ArNorm**
CER	0.66	0.76	0.72	0.67	**0.80**
WER	0.55	0.55	0.60	0.59	0.59
MER	0.54	0.54	0.60	0.59	0.59
WIL	0.44	0.44	0.51	0.50	0.50

Phonological Metrics (Error Measures)						
			w_s=1	**w_s=2**	**w_s=4**	**w_s=8**
PER	0.74	**PSD**	0.70	0.73	**0.75**	**0.75**

Semantic Metrics (Accuracy Measures)						
	Base	**MT>Ar**	**MT>En**	**MT>Ja**	**Average (Base, MT>*)**	**Max (Base, MT>*)**
BLEU	0.45	0.37	0.44	0.39	0.48	0.51
chrF	0.63	0.52	0.56	0.44	0.62	0.66
BertScore(F1)	0.58	0.43	0.53	0.47	0.58	0.58
CosineSim[mBERT]	0.62	0.49	0.59	0.54	**0.67**	**0.67**
CosineSim[*BERT]	0.45	0.53	0.57	0.55		

Table 10.5: Sentence-level correlations calculated between *GoldCER* and the scores of different metrics. Given that the orthographic and phonological evaluation metrics used are error metrics, while the semantic metrics are accuracy metrics, we present the correlations against (1-*GoldCER*) for semantic metrics, to consistently report positive correlations for easier readability. CosineSim[mBERT] are the results achieved using mBERT. CosineSim[*BERT] are the results achieved using BiBERT for the original hypothesis-reference pairs (Base), CAMeLBERT for Arabic translations (which outperformed AraBERT), BERT-base for English translations, and BERT-base-japanese for Japanese (which outperformed BERT-large-japanese).

This approach provides the following advantages: (1) through translation, words in different scripts or with spelling variations can be mapped to the same/similar word(s), and (2) by using semantic similarity, such words can be assigned lower errors if closely represented in the embedding space. As seen in Table 10.4, the words 'ناشيونال سكولز' were successfully mapped through translation to 'National Schools'. One limitation of this approach, however, is that it is highly dependent on the quality of the embeddings obtained from the pretrained models as well as MT performance.

10.4 Experimental Results

10.4.1 Experimental Setup

We define the ground truth error for each hypothesis to be the amount of human post-editing effort required to correct it. Accordingly, we define *GoldCER* to be the edit distance between the hypothesis and minimal edits annotation calculated using CER. We opt for using CER in this calculation over other evaluation metrics as it provides higher granularity in reflecting the effort done by annotators and is consistent with the annotation guidelines.

In order to assess the performance of the evaluation metrics, we compare their scores against *GoldCER* on both the *sentence-level* and *system-level*.[11] In the sentence-level evaluation, we calculate the correlation between the scores provided by each metric for every hypothesis-reference pair against their corresponding *GoldCER* values. This evaluation provides a fine-grained assessment demonstrating the ability of each metric to distinguish the amount of errors in each hypothesis. In the system-level evaluation, we calculate the overall *GoldCER* for each of the three systems (*HMM-DNN*, *Conformer-A*, and *Conformer-F*) which acts as the ground truth score. We then obtain the overall scores for the three systems using each evaluation metric, assessing its ability to provide correct system ranking.

10.4.2 Results and Discussion

Overall Sentence-level Evaluation

In Table 10.5, we present the sentence-level correlations between *GoldCER* and different metrics' scores. For the **orthographic metrics**, we demonstrate the correlations using CER, WER, MER, and WIL applied on the

[11] We exclude 17 utterances that are annotated as unclear.

	CMI	CER	WER	MER	WIL	Tr+	PSD	Sem
All	14.8	0.66	0.55	0.54	0.44	**0.80**	0.75	0.67
R02	13.4	0.53	0.49	0.47	0.39	**0.77**	0.69	0.67
R05	11.5	0.78	0.58	0.50	0.42	**0.83**	0.78	0.67
R07	19.7	0.70	0.61	0.66	0.55	**0.75**	0.68	**0.75**
R09	19.6	0.64	0.62	0.48	0.41	**0.84**	0.81	0.56
R10	10.8	**0.88**	0.53	0.57	0.46	0.86	0.87	0.67
R12	10.7	0.68	0.61	0.66	0.56	**0.69**	0.65	**0.69**
R14	15.2	0.66	0.67	0.67	0.57	**0.88**	0.83	0.76
R* Avg	14.4	0.70	0.59	0.57	0.48	**0.80**	0.76	0.68
R* StDev	3.9	0.11	0.06	0.09	0.08	0.07	0.08	0.07

Table 10.6: Reported sentence-level correlations per recording, with recording-level Code-Mixing Index (CMI in percentage). We show the correlations for CER, WER, MER, WIL, in addition to transliterating to Arabic followed by Alif/Ya normalization (Tr+), PSD ($w_s = 4$), and $Average(Base, MT > *)$ for the semantic measure (Sem).

original hypothesis-reference pairs (Base) as well as their transliterations into Arabic (*Translit>Ar*) and Roman (*Translit>En*) scripts. We also show the effect of applying Alif/Ya normalization on the original sentences (*Base+ArNorm*) and the transliterations in Arabic script (Translit>Ar+ArNorm). We observe that across the different settings, the highest correlations are achieved using CER, followed by WER and MER, then WIL. Similar to the findings in [276], with normalization, higher correlations are achieved, which we report for all the four metrics. When applying CER, transliterating to Arabic outperforms transliterating to English. However, for word-level evaluation metrics (WER, MER, and WIL), transliterating to English gives higher correlations, even though the references are dominated by Arabic words (77%). This can be justified by the ability to resolve Arabic unstandardized orthography issues when transliterating into English.

In the scope of **phonological metrics**, we find that $w_S = 4$ provides the highest correlation of PSD with human judgment. The efficacy of incorporating phonological similarity in the error calculation is demonstrated, where PSD outperforms PER (Phoneme Error Rate). For **semantic metrics**, the highest correlations are achieved using cosine similarity, followed by chrF, BertScore(F1), then BLEU. Across the different metrics, higher correlations are achieved by applying the metrics directly on the original text rather than on translated text. This can be foreseen, as the success of this approach is dependent on the performance of the underlying MT system. By looking into the translations, it is obvious that a significant amount of translation errors is introduced, which is propagated to the metric scores. However, the translation step still proves to be beneficial, where across the metrics, higher correlations are achieved by applying sentence-level aggregation to the scores achieved across the different language setups, where we have tried Avg and Max functions.[12] The highest correlation for semantic metrics is achieved by aggregating the cosine similarity scores using the Avg function across the original sentences and the three translations. While the semantic-based metrics provide lower correlations compared to transliteration and phonetic similarity, we believe that the efficacy of the proposed translation pipeline should be revisited with future advances in CS MT systems.

In general, results show that transliteration and phonetic similarity outperform conventional CER, WER, MER, and WIL evaluation metrics applied directly on the original sentences. Despite the limitations of the semantic measures, it outperforms WER, MER, and WIL, and performs equally well as CER. The highest correlation is achieved by using CER over the texts transliterated into Arabic script followed by Arabic normalization.

Recording-based Sentence-level Evaluation

To further investigate the validity of our results, we perform the same sentence-level evaluation across the seven different recordings in our corpus, as presented in Table 10.6. Given that the recordings contain different degrees of CS, this analysis allows us to evaluate the consistency of our results. We measure CS levels in terms of *Code-Mixing Index* (CMI) [291, 32], calculated on the utterance-level as follows:

$$C_u(x) = 100 * \frac{\frac{1}{2} * (N(x) - max_{L_i \in \mathbf{L}}\{t_{L_i}\}(x)) + \frac{1}{2}P(x)}{N(x)}$$

[12] For Avg, the score for each reference-hypothesis pair is calculated as the average of cosine similarity scores achieved across the four language setups, where we choose the best-performing pretrained model for each language. For Max, the score for each pair is calculated as the maximum cosine similarity score across the four language setups using mBERT model. We also investigate taking the average of cosine similarity scores using mBERT model, however, it gives lower correlations.

		Orthographic (Error Measures)			Phonological (Error Measures)	Semantic (Accuracy Measures)		
	Gold CER	WER	CER	CER	PSD (ws=4)	CosineSim	Av(CosineSim)	Max(CosineSim)
		Base	Base	Translit>Ar +Norm		Base	(Base, MT>*)	(Base, MT>*)
Conformer-A	7.9 (1)	47.0 (1)	23.5 (2)	15.3 (1)	15.5 (1)	0.872 (1)	0.886 (1)	0.914 (1)
Conformer-F	13.5 (2)	50.1 (2)	23.0 (1)	18.6 (2)	18.0 (2)	0.850 (2)	0.862 (2)	0.891 (2)
HMM-DNN	21.0 (3)	57.6 (3)	32.9 (3)	26.1 (3)	25.4 (3)	0.820 (3)	0.845 (3)	0.871 (3)

Table 10.7: System-level overall scores and ranking (denoted between parentheses) across the different evaluation metrics.

where N is the number of language-dependent tokens in utterance x; $L_i \in \mathbf{L}$ the set of all languages in the corpus; $max\{t_{L_i}\}$ represents the number of tokens in the dominating language in x, with $1 \leq max\{t_{L_i}\} \leq N$; and P is the number of code alternation points in x; $0 \leq P < N$. We calculate the recording-level CMI by averaging the utterance-level values. We note that there is a degree of variety in the CMI values across recordings ranging from 11 to 20. In Table 10.6, we show the recording-level CMI values and the correlations for CER, WER, MER, WIL, transliterating to Arabic followed by Alif/Ya normalization (Tr+), PSD ($w_s = 4$), and $Average(Base, MT > *)$ for the semantic measure (Sem). We observe that for the conventional metrics, it is mostly the case that CER outperforms WER/MER (performing on-par), which outperform WIL. We confirm that transliterating to Arabic followed by Alif/Ya normalization is the best-performing metric, outperforming CER for 6/7 recordings. In the case where CER achieved highest correlation, it is only slightly better than transliteration. By looking into the standard deviation of the metrics' correlation scores, we observe that CER has the highest value (0.11), compared to transliteration (0.07), phonological (0.08) and semantic (0.07) measures. This reflects that CER's performance is less consistent than other metrics. In future work, we plan to understand the relation between the correlation scores and CS behaviour as well as other variables.

System-level Evaluation

Results for the system-level evaluation are presented in Table 10.7. For semantic-based metrics, the overall score is calculated as the average of sentence-level scores. As indicated by the *GoldCER*, the ranking of the systems is: *Conformer-A*, *Conformer-F*, and *HMM-DNN*. We show that, apart from CER applied directly on the original sentences, all the evaluation metrics provide the same ranking conclusion. The relative scores of the systems are not equivalently reflected in all metrics though, which is worth further investigations. In the future, we plan to include more systems, in order to be able to derive correlations between systems' overall scores.

Conclusion and Future Work

In this work, we *(i)* develop a corpus of human judgment – with minimal edits of different ASR output; and *(ii)* benchmark the performance of different evaluation metrics and their ability to correctly evaluate CS ASR outputs in correlation to the ground truth human post-editing effort. We cover commonly-used evaluation metrics, in addition to three approaches aiming at handling CS challenges: transliteration, phonetic similarity, and semantic similarity. Our results show that WER and CER are not adequate for evaluating CS languages having cross-transcription and spelling variation. The highest correlation to the post-editing effort is achieved by transliteration followed by phonetic similarity, semantic similarity, CER, and WER, in order. In future, we plan to evaluate the proposed methods for the MUCS2021 [12] challenge to ensure generalization across more languages. Furthermore, we plan to create the human acceptability corpus for language pairs sharing the same writing script.

Bibliography

[1] G. A. Guzman, J. Serigos, B. Bullock, and A. J. Toribio, "Simple tools for exploring variation in code-switching for linguists," in *Proceedings of the second workshop on computational approaches to code switching*, pp. 12–20, 2016.

[2] B. Gambäck and A. Das, "Comparing the level of code-switching in corpora," in *Proc. of the International Conference on Language Resources and Evaluation*, 2016.

[3] G. A. Guzmán, J. Ricard, J. Serigos, B. E. Bullock, and A. J. Toribio, "Metrics for modeling code-switching across corpora.," in *INTERSPEECH*, pp. 67–71, 2017.

[4] R. Barnett, E. Codó, E. Eppler, M. Forcadell, P. Gardner-Chloros, R. Van Hout, M. Moyer, M. C. Torras, M. T. Turell, M. Sebba, *et al.*, "The lides coding manual: A document for preparing and analyzing language interaction data version 1.1–july 1999.," *International Journal of Bilingualism*, vol. 4, no. 2, pp. 131–271, 2000.

[5] B. Gambäck and A. Das, "On measuring the complexity of code-mixing," in *Proceedings of the 11th international conference on natural language processing, Goa, India*, pp. 1–7, 2014.

[6] D.-C. Lyu, T.-P. Tan, E. S. Chng, and H. Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia," in *Interspeech*, 2010.

[7] I. Hamed, N. T. Vu, and S. Abdennadher, "Arzen: A speech corpus for code-switched egyptian arabic-english," in *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4237–4246, 2020.

[8] T. Niesler *et al.*, "A first south african corpus of multilingual code-switched soap opera speech," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[9] V. Srivastava and M. Singh, "Challenges and limitations with the metrics measuring the complexity of code-mixed text," *arXiv preprint arXiv:2106.10123*, 2021.

[10] E. Taljard and S. E. Bosch, "A comparison of approaches to word class tagging: Disjunctively vs. conjunctively written bantu languages," *Nordic journal of African studies*, vol. 15, no. 4, 2006.

[11] P. Kodali, A. Goel, M. Choudhury, M. Shrivastava, and P. Kumaraguru, "Symcom-syntactic measure of code mixing a study of english-hindi code-mixing," in *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 472–480, 2022.

[12] A. Diwan, R. Vaideeswaran, S. Shah, A. Singh, S. Raghavan, S. Khare, V. Unni, S. Vyas, A. Rajpuria, C. Yarra, A. Mittal, P. K. Ghosh, P. Jyothi, K. Bali, V. Seshadri, S. Sitaram, S. Bharadwaj, J. Nanavati, R. Nanavati, K. Sankaranarayanan, T. Seeram, and B. Abraham, "Multilingual and code-switching ASR challenges for low resource Indian languages," in *Proc. of Interspeech*, 2021.

[13] K. Bali, J. Sharma, M. Choudhury, and Y. Vyas, ""i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook," in *Proceedings of the first workshop on computational approaches to code switching*, pp. 116–126, 2014.

[14] R. Begum, K. Bali, M. Choudhury, K. Rudra, and N. Ganguly, "Functions of code-switching in tweets: An annotation framework and some initial experiments," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1644–1650, 2016.

[15] C. Myers-Scotton, *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press, 1997.

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

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

[18] S. Poplack, "Contrasting patterns of code-switching in two communities," *Codeswitching: Anthropological and sociolinguistic perspectives*, vol. 48, pp. 215–244, 1988.

[19] G. A. Guzmán, J. Ricard, J. Serigos, B. Bullock, and A. J. Toribio, "Moving code-switching research toward more empirically grounded methods.," in *CDH@ TLT*, pp. 1–9, 2017.

[20] S. Poplack, ""sometimes i'll start a sentence in spanish y termino en español": Toward a typology of code-switching," *Linguistics*, vol. 51, no. s1, pp. 11–14, 2013.

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

[22] F. Heylighen and J. Dewaele, "Formality of language: definition, measurement and behavioral determinants," 1999.

[23] B. Winter and S. Grawunder, "The phonetic profile of Korean formal and informal speech registers," *Journal of Phonetics*, vol. 40, p. 808–815, 11 2012.

[24] E. Sherr-Ziarko, "Acoustic Properties of Formality in Conversational Japanese," pp. 1285–1289, 09 2016.

[25] A. S. Doğruöz, S. Sitaram, B. E. Bullock, and A. J. Toribio, "A survey of code-switching: Linguistic and social perspectives for language technologies," *arXiv preprint arXiv:2301.01967*, 2023.

[26] Y. Li and P. Fung, "Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints," in *ICASSP*, 2013.

[27] G. Sreeram and R. Sinha, "Exploration of end-to-end framework for code-switching speech recognition task: Challenges and enhancements," *IEEE Access*, 2020.

[28] D. Amazouz, M. Adda-Decker, and L. Lamel, "Addressing code-switching in French/Algerian Arabic speech," in *Interspeech*, 2017.

[29] A. Ali, S. Chowdhury, A. Hussein, and Y. Hifny, "Arabic code-switching speech recognition using monolingual data," *Interspeech 2021*, 2021.

[30] I. Hamed, P. Denisov, C. Li, M. Elmahdy, S. Abdennadher, and N. Vu, "Investigations on speech recognition systems for low-resource dialectal Arabic-English code-switching speech," *Computer Speech & Language*, p. 101278, 2021.

[31] S. Chowdhury, A. Hussein, A. Abdelali, and A. Ali, "Towards one model to rule all: Multilingual strategy for dialectal code-switching Arabic Asr," *Interspeech 2021*, 2021.

[32] S. A. Chowdhury, Y. Samih, M. Eldesouki, and A. Ali, "Effects of dialectal code-switching on speech modules: A study using Egyptian Arabic broadcast speech," in *Proc. of Interspeech*, pp. 2382–2386, 2020.

[33] A. Ali, S. Chowdhury, M. Afify, W. El-Hajj, H. Hajj, M. Abbas, A. Hussein, N. Ghneim, M. Abushariah, and A. Alqudah, "Connecting Arabs: bridging the gap in dialectal speech recognition," *Communications of the ACM*, 2021.

[34] C. Myers-Scotton, "11 a lexically based model of code-switching," *One speaker, two languages: Cross-disciplinary perspectives on code-switching*, 1995.

[35] D. Sankoff, "A formal production-based explanation of the facts of code-switching," *Bilingualism: language and cognition*, 1998.

[36] H. Belazi, E. Rubin, and A. Toribio, "Code switching and x-bar theory: The functional head constraint," *Linguistic inquiry*, 1994.

[37] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali, "Language modeling for code-mixing: The role of linguistic theory based synthetic data," in *ACL*, 2018.

[38] G. Winata, A. Madotto, C. Wu, and P. Fung, "Code-switched language models using neural based synthetic data from parallel sentences," *arXiv preprint arXiv:1909.08582*, 2019.

[39] L. Qin, M. Ni, Y. Zhang, and W. Che, "Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp," *arXiv preprint arXiv:2006.06402*, 2020.

[40] Y. Lee, "Morphological analysis for statistical machine translation," 2004.

[41] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *ACL*, 2017.

[42] H. Sajjad, A. Abdelali, N. Durrani, and F. Dalvi, "AraBench: Benchmarking dialectal Arabic-English machine translation," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[43] C. Dyer, V. Chahuneau, and N. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *NACL: Human Language Technologies*, 2013.

[44] Z.-Y. Dou and G. Neubig, "Word alignment by fine-tuning embeddings on parallel corpora," in *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

[45] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for Arabic," in *NACL: Demonstrations*, 2016.

[46] D. Ataman and M. Federico, "An evaluation of two vocabulary reduction methods for neural machine translation," in *Association for Machine Translation in the Americas*, 2018.

[47] D. Klein and C. Manning, "Accurate unlexicalized parsing, in proceedings of the 41st meeting of the association for computational linguistics," 2003.

[48] H. Mubarak, A. Hussein, S. Chowdhury, and A. Ali, "QASR: QCRI Aljazeera speech resource A large scale annotated arabic speech corpus," in *ACL*, 2021.

[49] J. Olive, C. Christianson, and J. McCary, *Handbook of natural language processing and machine translation: DARPA global autonomous language exploitation*. Springer Science & Business Media, 2011.

[50] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic MGB-3," in *ASRU*, 2017.

[51] A. Ali, S. Shon, Y. Samih, H. Mubarak, A. Abdelali, J. Glass, S. Renals, and K. Choukri, "The MGB-5 challenge: Recognition and dialect identification of dialectal Arabic speech," in *ASRU*, 2019.

[52] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[53] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[54] M. Mohri, "Weighted automata algorithms," in *Handbook of weighted automata*, 2009.

[55] A. Ali, P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "The MGB-2 challenge: Arabic multi-dialect broadcast media recognition," in *SLT*, 2016.

[56] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*, 2018.

[57] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. of Interspeech*, 2015.

[58] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. of Interspeech*, 2016.

[59] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *International Conference on Acoustics, Speech, and Signal Processing,*, pp. 532–535, IEEE, 1989.

[60] I. Hamed, A. Hussein, O. Chellah, S. Chowdhury, H. Mubarak, S. Sitaram, N. Habash, and A. Ali, "Benchmarking evaluation metrics for code-switching automatic speech recognition," in *SLT*, 2023.

[61] Ö. Çetinoğlu, S. Schulz, and N. T. Vu, "Challenges of computational processing of code-switching," in *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, 2016.

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

[63] G. Lee, X. Yue, and H. Li, "Linguistically motivated parallel data augmentation for code-switch language modeling," in *Interspeech*, pp. 3730–3734, 2019.

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

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

[66] G. Kuwanto, A. F. Akyürek, I. C. Tourni, S. Li, and D. Wijaya, "Low-resource machine translation for low-resource languages: Leveraging comparable data, code-switching and compute resources," *CoRR*, vol. abs/2103.13272, 2021.

[67] C.-T. Chang, S.-P. Chuang, and H.-Y. Lee, "Code-switching sentence generation by generative adversarial networks and its application to data augmentation," in *Interspeech*, pp. 554–558, 2018.

[68] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Learn to code-switch: Data augmentation using copy mechanism on language modeling," *CoRR*, vol. abs/1810.10254, 2018.

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

[70] C.-Y. Li and N. T. Vu, "Improving code-switching language modeling with artificially generated texts using cycle-consistent adversarial networks," in *Interspeech*, pp. 1057–1061, 2020.

[71] I. Tarunesh, S. Kumar, and P. Jyothi, "From machine translation to code-switching: Generating high-quality code-switched text," *arXiv preprint arXiv:2107.06483*, 2021.

[72] R. Appicharla, K. K. Gupta, A. Ekbal, and P. Bhattacharyya, "Iitp-mt at calcs2021: English to hinglish neural machine translation using unsupervised synthetic code-mixed parallel corpus," in *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pp. 31–35, 2021.

[73] A. Gupta, A. Vavre, and S. Sarawagi, "Training data augmentation for code-mixed translation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5760–5766, 2021.

[74] J. Xu and F. Yvon, "Can you traducir this? machine translation for code-switched input," *arXiv preprint arXiv:2105.04846*, 2021.

[75] M. A. Menacer, D. Langlois, D. Jouvet, D. Fohr, O. Mella, and K. Smaïli, "Machine translation on a parallel code-switched corpus," in *Canadian Conference on Artificial Intelligence*, pp. 426–432, Springer, 2019.

[76] K. Song, Y. Zhang, H. Yu, W. Luo, K. Wang, and M. Zhang, "Code-switching for enhancing nmt with pre-specified translation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).*, pp. 449–459, 2019.

[77] M. S. Z. Rizvi, A. Srinivasan, T. Ganu, M. Choudhury, and S. Sitaram, "Gcm: A toolkit for generating synthetic code-mixed text," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 205–211, 2021.

[78] A. Hussein, S. A. Chowdhury, A. Abdelali, N. Dehak, and A. Ali, "Code-switching text augmentation for multilingual speech processing," *arXiv preprint arXiv:2201.02550*, 2022.

[79] S. Poplack, "Sometimes i'll start a sentence in spanish y termino en español: Toward a typology of code-switching," *The bilingualism reader*, vol. 18, no. 2, pp. 221–256, 1980.

[80] T. Buckwalter, "Buckwalter Arabic morphological analyzer version 1.0." Linguistic Data Consortium (LDC) catalog number LDC2002L49, ISBN 1-58563-257-0, 2002.

[81] I. Hamed, N. T. Vu, and S. Abdennadher, "Arzen: A speech corpus for code-switched Egyptian Arabic-English," in *Proc. of the Language Resources and Evaluation Conference*, 2020.

[82] L. D. C. LDC, "Bolt program: Arabic to english translation guidelines." `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/bolt-phase2-arabic-cts-translation-guidelines-v1.pdf`, 2013.

[83] N. Habash, R. Eskander, and A. Hawwari, "A morphological analyzer for egyptian arabic," in *Proceedings of the twelfth meeting of the special interest group on computational morphology and phonology*, pp. 1–9, 2012.

[84] N. Habash and B. Dorr, "Catvar: A database of categorial variations for english," in *Proceedings of Machine Translation Summit IX: System Presentations*, 2003.

[85] F. Casacuberta and E. Vidal, "Giza++: Training of statistical translation models," *Polytechnic University of Valencia, Valencia, Spain*, 2007.

[86] G. Kumar, Y. Cao, R. Cotterell, C. Callison-Burch, D. Povey, and S. Khudanpur, "Translations of the callhome egyptian arabic corpus for conversational speech translation," in *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, 2014.

[87] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. F. Zaidan, and C. CallisonBurch, "Machine translation of arabic dialects," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 49–59, 2012.

[88] S. Chen, D. Fore, S. Strassel, H. Lee, and J. Wright, "Bolt egyptian arabic sms/chat and transliteration ldc2017t07." Philadelphia: Linguistic Data Consortium, 2017.

[89] S. Chen, J. Tracey, C. Walker, and S. Strassel, "Bolt egyptian arabic parallel discussion forums data." Linguistic Data Consortium (LDC) catalog number LDC2019T01, ISBN 1-58563-871-4, 2019.

[90] X. Li, S. Grimes, and S. Strassel, "Bolt egyptian arabic-english word alignment – conversational telephone speech training." Linguistic Data Consortium (LDC) catalog number LDC2020T05, ISBN 1-58563-920-6, 2020.

[91] H. Bouamor, N. Habash, M. Salameh, W. Zaghouani, O. Rambow, D. Abdulrahim, O. Obeid, S. Khalifa, F. Eryani, A. Erdmann, and K. Oflazer, "The madar arabic dialect corpus and lexicon.," in *LREC*, 2018.

[92] M. Diab, N. Habash, O. Rambow, and R. Roth, "Ldc arabic treebanks and associated corpora: Data divisions manual," *arXiv preprint arXiv:1309.5652*, 2013.

[93] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pp. 177–180, 2007.

[94] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, "Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic.," in *LREC*, vol. 14, pp. 1094–1101, 2014.

[95] L. Kjeldgaard and L. Nielsen, "Nerda," GitHub, 2021.

[96] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[97] F. Guzmán, P.-J. Chen, M. Ott, J. Pino, G. Lample, P. Koehn, V. Chaudhary, and M. Ranzato, "The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english," *arXiv preprint arXiv:1902.01382*, 2019.

[98] I. Hamed, P. Denisov, C.-Y. Li, M. Elmahdy, S. Abdennadher, and N. T. Vu, "Investigations on speech recognition systems for low-resource dialectal arabic–english code-switching speech," *Computer Speech & Language*, vol. 72, p. 101278, 2022.

[99] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, *et al.*, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, pp. 2207–2207, 2018.

[100] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, pp. 2613–2617, 2019.

[101] H. Gadalla, H. Kilany, H. Arram, A. Yacoub, A. El-Habashi, A. Shalaby, K. Karins, E. Rowson, R. MacIntyre, P. Kingsbury, D. Graff, and C. McLemore, "Callhome egyptian arabic transcripts," *Linguistic Data Consortium, Philadelphia*, 1997.

[102] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic mgb-3," in *ASRU*, pp. 316–322, IEEE, 2017.

[103] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, pp. 5206–5210, IEEE, 2015.

[104] A. Ali, P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "The mgb-2 challenge: Arabic multi-dialect broadcast media recognition," in *SLT*, pp. 279–284, IEEE, 2016.

[105] A. Das and B. Gambäck, "Identifying languages at the word level in code-mixed indian social media text," in *Proc. of the International Conference on Natural Language Processing*, 2014.

[106] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[107] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of the annual meeting of the Association for Computational Linguistics*, 2002.

[108] M. Popović, "chrf++: words helping character n-grams," in *Proc. of the conference on machine translation*, 2017.

[109] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *Proc. of the International Conference on Learning Representations*, 2019.

[110] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, (Belgium, Brussels), pp. 186–191, Association for Computational Linguistics, Oct. 2018.

[111] M. Oudah, A. Almahairi, and N. Habash, "The impact of preprocessing on arabic-english statistical and neural machine translation," *arXiv preprint arXiv:1906.11751*, 2019.

[112] I. Hamed, M. Elmahdy, and S. Abdennadher, "Collection and analysis of code-switch egyptian arabic-english speech corpus," in *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, 2018.

[113] I. Hamed, M. Zhu, M. Elmahdy, S. Abdennadher, and N. T. Vu, "Code-switching language modeling with bilingual word embeddings: A case study for egyptian arabic-english," in *International Conference on Speech and Computer*, pp. 160–170, Springer, 2019.

[114] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[115] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019.

[116] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry, W. Macherey, Z. Chen, and Y. Wu, "Massively multilingual neural machine translation in the wild: Findings and challenges," 2019.

[117] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," 2016.

[118] X. Tan, Y. Ren, D. He, T. Qin, Z. Zhao, and T.-Y. Liu, "Multilingual neural machine translation with knowledge distillation," 2019.

[119] H. Xu, B. Van Durme, and K. Murray, "Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation," 2021.

[120] W. Lan, Y. Chen, W. Xu, and A. Ritter, "An empirical study of pre-trained transformers for Arabic information extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 4727–4734, Association for Computational Linguistics, Nov. 2020.

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

[122] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic mgb-3," 2017.

[123] A. Wang and K. Cho, "Bert has a mouth, and it must speak: Bert as a markov random field language model," 2019.

[124] *The Cambridge Handbook of Linguistic Code-switching*. Cambridge Handbooks in Language and Linguistics, Cambridge University Press, 2009.

[125] G. I. Winata, A. Madotto, C. Wu, and P. Fung, "Code-switched language models using neural based synthetic data from parallel sentences," *CoRR*, vol. abs/1909.08582, 2019.

[126] C. Du, H. Li, Y. Lu, L. Wang, and Y. Qian, "Data augmentation for end-to-end code-switching speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 194–200, 2021.

[127] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali, "Language modeling for code-mixing: The role of linguistic theory based synthetic data," in *Proceedings of ACL*, pp. 1543–1553, July 2018.

[128] D.-C. Lyu, T. P. Tan, C. E. Siong, and H. Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia," in *INTERSPEECH*, 2010.

[129] Y. Li and P. Fung, "Code switch language modeling with functional head constraint," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4913–4917, 2014.

[130] A. Hussein, S. Absar Chowdhury, A. Abdelali, N. Dehak, and A. Ali, "Code-switching text augmentation for multilingual speech processing," 2022.

[131] H. Adel, N. T. Vu, and T. Schultz, "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)*, pp. 206–211, Aug. 2013.

[132] E. Yılmaz, H. van den Heuvel, and D. van Leeuwen, "Acoustic and Textual Data Augmentation for Improved ASR of Code-Switching Speech," in *Proc. Interspeech 2018*, pp. 1933–1937, 2018.

[133] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switching language modeling using syntax-aware multi-task learning," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pp. 62–67, July 2018.

[134] K. Chandu, T. Manzini, S. Singh, and A. W. Black, "Language informed modeling of code-switched text," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pp. 92–97, July 2018.

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

[136] Y. Gao, J. Feng, Y. Liu, L. Hou, X. Pan, and Y. Ma, "Code-Switching Sentence Generation by Bert and Generative Adversarial Networks," in *Proc. Interspeech 2019*, pp. 3525–3529, 2019.

[137] B. Samanta, S. Reddy, H. Jagirdar, N. Ganguly, and S. Chakrabarti, "A Deep Generative Model for Code-Switched Text," June 2019.

[138] K. Raghavi Chandu and A. W. Black, "Style Variation as a Vantage Point for Code-Switching," May 2020.

[139] B. Samanta, S. Reddy, H. Jagirdar, N. Ganguly, and S. Chakrabarti, "A deep generative model for code-switched text," *arXiv e-prints*, June 2019.

[140] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting Similarities among Languages for Machine Translation," *arXiv e-prints*, Sept. 2013.

[141] S. Ruder, I. Vulić, and A. Søgaard, "A Survey Of Cross-lingual Word Embedding Models," *arXiv e-prints*, June 2017.

[142] A. Conneau, S. Wu, H. Li, L. Zettlemoyer, and V. Stoyanov, "Emerging cross-lingual structure in pretrained language models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6022–6034, July 2020.

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

[144] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv e-prints*, p. arXiv:1706.03762, June 2017.

[145] C. Hokamp and Q. Liu, "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)*, pp. 1535–1546, July 2017.

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

[147] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 10 2017.

[148] N. Kitaev and D. Klein, "Constituency parsing with a self-attentive encoder," *CoRR*, vol. abs/1805.01052, 2018.

[149] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Speech and Computer*, pp. 198–208, Springer International Publishing, 2018.

[150] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Oriental COCOSDA 2017*, 2017. Submitted.

[151] J. Calvillo, L. Fang, J. Cole, and D. Reitter, "Surprisal predicts code-switching in Chinese-English bilingual text," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Nov. 2020.

[152] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependences in stochastic language modelling," *Comput. Speech Lang.*, vol. 8, pp. 1–38, 1994.

[153] Z. Zeng, Y. Khassanov, V. T. Pham, H. Xu, E. S. Chng, and H. Li, "On the End-to-End Solution to Mandarin-English Code-Switching Speech Recognition," in *Proc. Interspeech 2019*, pp. 2165–2169, 2019.

[154] J. R. NOVAK, N. MINEMATSU, and K. HIROSE, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework," *Natural Language Engineering*, vol. 22, no. 6, p. 907–938, 2016.

[155] S. A. Chowdhury, A. Hussein, A. Abdelali, and A. Ali, "Towards one model to rule all: Multilingual strategy for dialectal code-switching arabic asr," in *Interspeech*, 2021.

[156] S. Thomas, B. Kingsbury, G. Saon, and H.-K. J. Kuo, "Integrating text inputs for training and adapting rnn transducer asr models," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8127–8131, 2022.

[157] J. Thienpondt and K. Demuynck, "Transfer learning for robust low-resource children's speech asr with transformers and source-filter warping," 2022.

[158] A. C S, P. A P, and A. G. Ramakrishnan, "Unsupervised domain adaptation schemes for building asr in low-resource languages," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 342–349, 2021.

[159] L. Milroy and W. Li, "A social network approach to code-switching: The example of a bilingual community in britain," 1995.

[160] C. Myers-Scotton, *Code-Switching*, ch. 13, pp. 217–237. John Wiley & Sons, Ltd, 2017.

[161] C. M. Scotton, "The possibility of code-switching: motivation for maintaining multilingualism," *Anthropological linguistics*, pp. 432–444, 1982.

[162] Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong, "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 243–250, 2021.

[163] K. Li, J. Li, G. Ye, R. Zhao, and Y. Gong, "Towards code-switching asr for end-to-end ctc models," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6076–6080, 2019.

[164] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," 2018.

[165] S. Murthy, D. Sitaram, and S. Sitaram, "Effect of tts generated audio on oov detection and word error rate in asr for low-resource languages," pp. 1026–1030, 09 2018.

[166] C. Peyser, H. Zhang, T. N. Sainath, and Z. Wu, "Improving performance of end-to-end asr on numeric sequences," *arXiv preprint arXiv:1907.01372*, 2019.

[167] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, "Speech recognition with augmented synthesized speech," in *ASRU*, 2019.

[168] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *ICASSP*, 2020.

[169] R. Zhao, J. Xue, J. Li, W. Wei, L. He, and Y. Gong, "On addressing practical challenges for rnn-transducer," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 526–533, IEEE, 2021.

[170] T. K. Lam, M. Ohta, S. Schamoni, and S. Riezler, "On-the-Fly Aligned Data Augmentation for Sequence-to-Sequence ASR," in *Proc. Interspeech 2021*, pp. 1299–1303, 2021.

[171] A. D. McCarthy, L. Puzon, and J. Pino, "Skinaugment: Auto-encoding speaker conversions for automatic speech translation," in *ICASSP*, 2020.

[172] T. K. Lam, S. Schamoni, and S. Riezler, "Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation," 2022.

[173] Y. Long, Y. Li, Q. Zhang, S. Wei, H. Ye, and J. Yang, "Acoustic data augmentation for mandarin-english code-switching speech recognition," *Applied Acoustics*, vol. 161, p. 107175, 2020.

[174] C. Du, H. Li, Y. Lu, L. Wang, and Y. Qian, "Data augmentation for end-to-end code-switching speech recognition," in *SLT*, 2021.

[175] Y. Sharma, B. Abraham, K. Taneja, and P. Jyothi, "Improving low resource code-switched asr using augmented code-switched TTS," *Interspeech*, 2020.

[176] K. Taneja, S. Guha, P. Jyothi, and B. Abraham, "Exploiting Monolingual Speech Corpora for Code-Mixed Speech Recognition," in *Interspeech*, 2019.

[177] H. Seki, S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," in *ICASSP*, 2018.

[178] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," 2017.

[179] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," 2015.

[180] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Component fusion: Learning replaceable language model component for end-to-end speech recognition system," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5361–5635, 2019.

[181] W. Michel, R. Schlüter, and H. Ney, "Early stage lm integration using local and global log-linear combination," 2020.

[182] R. A. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[183] R. A. Khan and J. Chitode, "Concatenative speech synthesis: A review," *International Journal of Computer Applications*, vol. 136, no. 3, pp. 1–6, 2016.

[184] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," 2015.

[185] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," 2018.

[186] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[187] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[188] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," 2015.

[189] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[190] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on espnet toolkit boosted by conformer," 2020.

[191] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

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

[193] M. P. Lewis, *Ethnologue: Languages of the world*. SIL international, 2009.

[194] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 265–271, 2017.

[195] B. Li, R. Pang, T. N. Sainath, A. Gulati, Y. Zhang, J. Qin, P. Haghani, W. R. Huang, M. Ma, and J. Bai, "Scaling end-to-end models for large-scale multilingual asr," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1011–1018, IEEE, 2021.

[196] Y. Lu, M. Huang, X. Qu, P. Wei, and Z. Ma, "Language adaptive cross-lingual speech representation learning with sparse sharing sub-networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6882–6886, IEEE, 2022.

[197] A. Bapna, C. Cherry, Y. Zhang, Y. Jia, M. Johnson, Y. Cheng, S. Khanuja, J. Riesa, and A. Conneau, "mslam: Massively multilingual joint pre-training for speech and text," *arXiv preprint arXiv:2202.01374*, 2022.

[198] X. Li, F. Metze, D. R. Mortensen, A. W. Black, and S. Watanabe, "Asr2k: Speech recognition for around 2000 languages without audio," *INTERPSPEECH 2022*, 2022.

[199] J. Bai, B. Li, Y. Zhang, A. Bapna, N. Siddhartha, K. Chai Sim, and T. N. Sainath, "Joint unsupervised and supervised training for multilingual asr," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[200] L. Zhou, J. Li, E. Sun, and S. Liu, "A configurable multilingual model is all you need to recognize all languages," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6422–6426, IEEE, 2022.

[201] C. Zhang, B. Li, T. Sainath, T. Strohman, S. Mavandadi, S.-y. Chang, and P. Haghani, "Streaming end-to-end multilingual speech recognition with joint language identification," *INTERSPEECH 2022*, 2022.

[202] H. Gonen and Y. Goldberg, "Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training," *Proc. EMNLP*, 2018.

[203] K. Li, J. Li, G. Ye, R. Zhao, and Y. Gong, "Towards code-switching asr for end-to-end ctc models," in *Proc. ICASSP*, 2019.

[204] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Component fusion: Learning replaceable language model component for end-to-end speech recognition system," in *Proc. ICASSP*, 2019.

[205] S. Shah, B. Abraham, S. Sitaram, V. Joshi, *et al.*, "Learning to recognize code-switched speech without forgetting monolingual speech recognition," *arXiv preprint arXiv:2006.00782*, 2020.

[206] Y. Lu, M. Huang, H. Li, J. Guo, and Y. Qian, "Bi-encoder transformer network for mandarin-english code-switching speech recognition using mixture of experts.," in *Proc. Interspeech*, 2020.

[207] X. Zhou, E. Yılmaz, Y. Long, Y. Li, and H. Li, "Multi-encoder-decoder transformer for code-switching speech recognition," *Interspeech*, 2020.

[208] S.-P. Chuang, T.-W. Sung, and H.-y. Lee, "Training code-switching language model with monolingual data," in *Proc. ICASSP*, 2020.

[209] S. Dalmia, Y. Liu, S. Ronanki, and K. Kirchhoff, "Transformer-transducers for code-switched speech recognition," in *Proc. ICASSP*, 2021.

[210] S. Zhang, J. Yi, Z. Tian, Y. Bai, J. Tao, *et al.*, "Decoupling pronunciation and language for end-to-end code-switching asr," in *Proc. ICASSP*, 2021.

[211] G. Liu and L. Cao, "Code-switch speech rescoring with monolingual data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[212] B. Yan, C. Zhang, M. Yu, S.-X. Zhang, S. Dalmia, D. Berrebbi, C. Weng, S. Watanabe, and D. Yu, "Joint modeling of code-switched and monolingual asr via conditional factorization," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6412–6416, 2022.

[213] J. Tian, J. Yu, C. Zhang, C. Weng, Y. Zou, and D. Yu, "Lae: Language-aware encoder for monolingual and multilingual asr," *INTERSPEECH 2022*, 2022.

[214] T. Song, Q. Xu, M. Ge, L. Wang, H. Shi, Y. Lv, Y. Lin, and J. Dang, "Language-specific characteristic assistance for code-switching speech recognition," *INTERSPEECH 2022*, 2022.

[215] A. Graves, "Sequence Transduction with Recurrent Neural Networks," in *Proc. ICML*, 2012.

[216] H. Liu, L. P. García-Perera, X. Zhang, J. Dauwels, A. W. Khong, S. Khudanpur, and S. J. Styles, "End-to-end language diarization for bilingual code-switching speech.," in *Interspeech*, 2021.

[217] K. Knight and J. Graehl, "Machine transliteration," *Computational Linguistics*, vol. 24, no. 4, pp. 599–612, 1998.

[218] P. Jyothi and M. Hasegawa-Johnson, "Transcribing continuous speech using mismatched crowdsourcing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[219] S. Thomas, K. Audhkhasi, and B. Kingsbury, "Transliteration based data augmentation for training multilingual asr acoustic models in low resource settings.," in *INTERSPEECH*, pp. 4736–4740, 2020.

[220] J. Billa, "Leveraging Non-Target Language Resources to Improve ASR Performance in a Target Language," in *Proc. Interspeech 2021*, pp. 2581–2585, 2021.

[221] L. Lugosch, T. Likhomanenko, G. Synnaeve, and R. Collobert, "Pseudo-labeling for massively multilingual speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7687–7691, IEEE, 2022.

[222] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *International Conference on Speech and Computer*, pp. 267–278, Springer, 2020.

[223] Z. Meng, N. Kanda, Y. Gaur, S. Parthasarathy, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, "Internal language model training for domain-adaptive end-to-end speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7338–7342, IEEE, 2021.

[224] W. Zhou, Z. Zheng, R. Schlüter, and H. Ney, "On language model integration for rnn transducer based speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8407–8411, IEEE, 2022.

[225] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[226] Z. Zeng, Y. Khassanov, H. X. Van Tung Pham, E. S. Chng, and H. Li, "On the end-to-end solution to mandarin-english code-switching speech recognition," 2019.

[227] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *Proc. ACL*, 2015.

[228] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech*, 2020.

[229] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *Proc. ICASSP*, 2021.

[230] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, IEEE, 2017.

[231] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an automatic speech recognition dedicated corpus," in *Proc. LREC*, 2012.

[232] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *JSTSP*, 2017.

[233] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NIPS*, vol. 33, 2020.

[234] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *CoRR, abs/2006.13979*, 2020.

[235] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[236] S. Sitaram, K. Chandu, S. Rallabandi, and A. Black, "A survey of code-switched speech and language processing," *arXiv preprint arXiv:1904.00784*, 2019.

[237] J.-H. Lee, C.-W. Lee, J.-S. Choi, J.-H. Chang, W. K. Seong, and J. Lee, "CTRL: Continual Representation Learning to Transfer Information of Pre-trained for WAV2VEC 2.0," *Interspeech*, 2022.

[238] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, 2002.

[239] V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Semi-supervised training of acoustic models using lattice-free MMI," in *ICASSP*, 2018.

[240] E. van der Westhuizen and T. Niesler, "A first South African corpus of multilingual code-switched soap opera speech.," in *LREC*, 2018.

[241] E. Yılmaz, A. Biswas, E. van der Westhuizen, F. de Wet, and T. Niesler, "Building a Unified Code-Switching ASR System for South African Languages," in *Interspeech*, 2018.

[242] A. Biswas, E. Yilmaz, F. De Wet, E. Van der westhuizen, and T. Niesler, "Semi-supervised Development of ASR Systems for Multilingual Code-switched Speech in Under-resourced Languages," in *LREC*, 2020.

[243] N. Wilkinson, A. Biswas, E. Yilmaz, F. De Wet, E. Van der westhuizen, and T. Niesler, "Semi-supervised Acoustic Modelling for Five-lingual Code-switched ASR using Automatically-segmented Soap Opera Speech," in *SLTU & CCURL*, 2020.

[244] P. Bell, M. J. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester, *et al.*, "The MGB challenge: Evaluating multi-genre broadcast media recognition," in *ASRU*, 2015.

[245] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013.

[246] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP*, 2013.

[247] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *SLT*, 2012.

[248] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[249] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[250] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[251] A. Vyas, S. Madikeri, and H. Bourlard, "Comparing CTC and LF-MMI for out-of-domain adaptation of wav2vec 2.0 acoustic model," in *Interspeech*, 2021.

[252] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," in *Interspeech*, 2022.

[253] K. D. N, P. Wang, and B. Bozza, "Using Large Self-Supervised Models for Low-Resource Speech Recognition," in *Interspeech*, 2021.

[254] M. Wiesner, D. Raj, and S. Khudanpur, "Injecting Text and Cross-Lingual Supervision in Few-Shot Learning from Self-Supervised Models," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8597–8601, May 2022. ISSN: 2379-190X.

[255] S. Kessler, B. Thomas, and S. Karout, "An Adapter Based Pre-Training for Efficient and Scalable Self-Supervised Speech Representation Learning," in *ICASSP*, 2022.

[256] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, May 2019.

[257] H. Y. Chan and P. Woodland, "Improving broadcast news transcription by lightly supervised discriminative training," in *ICASSP*, 2004.

[258] T. Fraga-Silva, J.-L. Gauvain, and L. Lamel, "Lattice-based unsupervised acoustic model training," in *ICASSP*, 2011.

[259] E. Wallington, B. Kershenbaum, P. Bell, and O. Klejch, "On the learning dynamics of semi-supervised training for ASR," in *Interspeech*, 2021.

[260] I. Caswell, T. Breiner, D. van Esch, and A. Bapna, "Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus," in *ACL*, 2020.

[261] T. Reitmaier, E. Wallington, D. Kalarikalayil Raju, O. Klejch, J. Pearson, M. Jones, P. Bell, and S. Robinson, "Opportunities and challenges of automatic speech recognition systems for low-resource language speakers," in *CHI Conference on Human Factors in Computing Systems*, 2022.

[262] A. Carmantini, P. Bell, and S. Renals, "Untranscribed web audio for low resource speech recognition.," in *Interspeech*, 2019.

[263] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli, "Self-training and pre-training are complementary for speech recognition," in *ICASSP*, 2021.

[264] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *ASRU*, 2011.

[265] E. Barnard, M. H. Davel, C. van Heerden, F. De Wet, and J. Badenhorst, "The nchlt speech corpus of the south african languages," in *SLTU*, 2014.

[266] S.-W. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Interspeech*, 2021.

[267] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for asr," in *ICASSP*, 2018.

[268] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *NAACL-HLT: Demonstrations*, 2019.

[269] O. Klejch, E. Wallington, and P. Bell, "The CSTR System for Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages," in *Interspeech*, 2021.

[270] J. J. van Vüren and T. Niesler, "Code-Switched Language Modelling Using a Code Predictive LSTM in Under-Resourced South African Languages," in *SLT*, 2022.

[271] S. Sitaram, K. R. Chandu, S. K. Rallabandi, and A. W. Black, "A survey of code-switched speech and language processing," *arXiv preprint arXiv:1904.00784*, 2019.

[272] B. M. L. Srivastava and S. Sitaram, "Homophone identification and merging for code-switched speech recognition," in *Proc. of Interspeech*, 2018.

[273] J. Emond, B. Ramabhadran, B. Roark, P. Moreno, and M. Ma, "Transliteration based approaches to improve code-switched speech recognition performance," in *Proc. of IEEE Spoken Language Technology Workshop*, 2018.

[274] S. A. Chowdhury, A. Hussein, A. Abdelali, and A. Ali, "Towards one model to rule all: Multilingual strategy for dialectal code-switching Arabic ASR," in *Proc. of Interspeech*, 2021.

[275] A. Ali, W. Magdy, and S. Renals, "Multi-reference evaluation for dialectal speech recognition system: A study for Egyptian ASR," in *Proc. of the Workshop on Arabic Natural Language Processing*, 2015.

[276] A. Ali, S. Khalifa, and N. Habash, "Towards variability resistant dialectal speech evaluation," in *Proc. of Interspeech*, 2019.

[277] N. Habash, M. Diab, and O. Rambow, "Conventional orthography for dialectal Arabic," in *Proc. of the International Conference on Language Resources and Evaluation*, 2012.

[278] S. Kim, D. Le, W. Zheng, T. Singh, A. Arora, X. Zhai, C. Fuegen, O. Kalinli, and M. L. Seltzer, "Evaluating user perception of speech recognition system quality with semantic distance metric," in *Proc. of Interspeech*, 2022.

[279] N. Habash, A. Soudi, and T. Buckwalter, "On Arabic Transliteration," in *Arabic Computational Morphology: Knowledge-based and Empirical Methods* (A. van den Bosch and A. Soudi, eds.), Springer, Netherlands, 2007.

[280] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. of Interspeech*, 2020.

[281] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2018.

[282] A. C. Morris, V. Maier, and P. Green, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition," in *Proc. of the International Conference on Spoken Language Processing*, 2004.

[283] A. Morris, "An information theoretic measure of sequence recognition performance," tech. rep., IDIAP, 2002.

[284] N. Durrani, H. Sajjad, H. Hoang, and P. Koehn, "Integrating an unsupervised transliteration model into statistical machine translation," in *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics*, 2014.

[285] D. R. Mortensen, S. Dalmia, and P. Littell, "Epitran: Precision G2P for many languages," in *Proc. of the International Conference on Language Resources and Evaluation*, 2018.

[286] S. Kim, A. Arora, D. Le, C.-F. Yeh, C. Fuegen, O. Kalinli, and M. L. Seltzer, "Semantic distance: A new metric for ASR performance analysis towards spoken language understanding," in *Proc. of Interspeech*, 2021.

[287] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[288] H. Xu, B. Van Durme, and K. Murray, "Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2021.

[289] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, "The interplay of variant, size, and task type in Arabic pre-trained language models," in *Proc. of the Arabic Natural Language Processing Workshop*, 2021.

[290] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for Arabic language understanding," in *Proc. of the Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020.

[291] B. Gambäck and A. Das, "Comparing the level of code-switching in corpora," in *Proc. of the International Conference on Language Resources and Evaluation*, 2016.