On the Impact of Matrix Language Identification for Automatic Speech Recognition of Code-Switched Speech

Anonymous ACL submission

Abstract

Code-switching (CS) is when a speaker alternates between two or more languages within a conversation, even within a single phrase. CS presents significant challenges for automatic speech recognition (ASR) systems due to mixed grammatical structures, accents and in-sentence language changes. One useful method for enhancing ASR performance on CS data is the accurate identification of the token Language Identities (LID). However, the LID of tokens do not explicitly inform ASR models of the dominant language which provides the grammatical structure for the CS utterance. The Matrix Language Frame (MLF) theory provides a syntactic and structural framework for the generation and analysis of CS utterances. It explains the CS process through the interaction of the two languages: the Matrix Language, which provides the grammatical structure for the CS utterance, and the Embedded Language, which is the language that is being inserted into the grammatical frame. This paper investigates the impact of Matrix Language Identity (MLID) analysis from the MLF theory on the effectiveness and accuracy of ASR systems when processing CS speech. The text-derived MLID was predicted from CS audio simultaneously with the ASR and token Language Identity (LID) prediction task, and the whole model was trained in a multi-task learning (MTL) setup. The proposed CS ASR system was compared to other MTL setups and showed a Mixed Error Rate (MER) decrease from 20.2% in an Attention-CTC ASR baseline to 19.7%. It was shown that having predicted MLID as Mandarin leads to an increase of recognised function words, indicating that MLID informs the ASR decoder of the grammatical properties of the utterance.

1 Introduction

Code-switching (CS) refers to the phenomenon in which speakers alternate between two or more languages within a single conversation, utterance, or sentence, both in spoken and written forms. While this linguistic behaviour is a natural and widespread mode of communication in many multilingual communities, it poses significant challenges for natural language processing systems, particularly Automatic Speech Recognition (ASR). One of the primary difficulties in developing effective CS-capable ASR systems lies in the limited availability of high-quality code-switched data, which is substantially scarcer than data for monolingual speech. As a result, ASR models trained on CS data often underperform relative to their monolingual counterparts (Radford et al., 2023).

Nevertheless, the prevalence of code-switching in multilingual societies—such as India, South Africa, and Nigeria—underscores the urgent need for robust and adaptable ASR systems that can handle language mixing effectively (Diwan et al., 2021; Ncoko et al., 2000; Rufai Omar, 1983). Improving ASR performance for CS speech is not only a technical challenge but also a critical step toward creating inclusive language technologies that reflect the linguistic realities of diverse populations.

The Linguistic Matrix Language Frame (MLF) theory (Myers-Scotton, 1993) provides an explanation for CS production and introduces the concept of a main, i.e. dominant language and a secondary, inserted language in CS utterances. These languages are called Matrix Language (ML) and Embedded Language (EL), respectively. MLF theory introduces two methods for ML determination:

1. *The Morpheme Order Principle* - the ML will provide the surface morpheme order for a CS utterance if it consists of singly occurring EL lexemes and any number of ML morphemes;

2. *The System Morpheme Principle* - all system morphemes that have grammatical relations external to their head constituent will come from the ML.

However, there is limited research on the automatic classification of the Matrix Language Identity (MLID). Consequently, due to the lack of research on MLID classification, the impact on the performance of ASR systems has not been studied. This paper aims to fill the gaps outlined above by investigating how MLID can enhance the effectiveness of ASR systems in recognising CS speech.

In this paper, a novel CS ASR system was proposed, which makes use of the auxiliary text-derived MLID prediction in an MTL setup to improve SotA CS ASR performance. Multiple ASR systems were trained to demonstrate the effects of incorporating conventional LID information on both utterance and token levels. The LID-based ASR systems were compared to an ASR system with an auxiliary MLID component. The MLID-based system ASR predictions were then analysed to demonstrate if using an MLID component in a multitask learning setup will increase the odds of recognising function words of the ML.

The remainder of the paper is structured as follows. The next section provides a comprehensive summary of the related literature. Afterwards, a detailed description of the methods used is provided. This is followed by a section on experiments, which provides information on datasets, detailed implementation, experiment descriptions and outcomes. The final content section presents a discussion and analysis of the results. Conclusions summarise and complete the paper.

2 Related work

Following the ideas of the linguistic theory, one can state that CS is an outcome of **compositions of two models** and it is not a manifestation of a new language. This way, according to linguistic theory, multilingual data should be sufficient for building an ASR model that can recognise CS. This encouraged the emergence of LID-based approaches in multilingual and CS ASR. Initial attempts at multilingual ASR (Ma et al., 2002) explored multilingual speech recognition conditioned on explicit utterance LID, setting the stage for later integration of LID into acoustic modelling. Similarly, Zhang et al. 2014 proposed training stacked bottleneck features conditioned on utterance LIDs, which improved multilingual ASR performance across multiple languages. However, not a single multilingual ASR system was able to surpass the quality of an ASR trained on CS data (Khassanov et al., 2019; Li et al., 2019; Shan et al., 2019). This is due to the multilingual ASR systems not being introduced to additional information that might help with CS detection in an audio stream, for example, sequential and grammatical information only available in real CS data. White et al. 2008 investigated acoustic models for CS, emphasising the importance of EL as opposed to ML acoustic modelling when handling EL insertions. However, Yılmaz et al. 2016 later explored bilingual deep neural networks for Frisian-Dutch CS speech, showing the feasibility of shared representations for closely related languages. LID of a whole CS utterance in an ASR pipeline may be performed when CS is regarded as a separate language (Mary N J et al., 2020), in this case, the MTL component performs both LID and CS detection. If a multilingual ASR system uses a conventional LID component as an auxiliary task without a separate CS language, then LID prompts the CS recognition output to be monolingual (Toshniwal et al., 2017). LID of a CS utterance is ill-defined since a CS utterance is a mix of two languages and cannot be a separate language according to the linguistic theory (Myers-Scotton, 1993). Consequently, there have been no attempts to classify CS utterances based on the dominant language in the utterance in such a way that it would improve CS ASR quality.

The emergence of end-to-end (E2E) models prompted a shift towards token-level and frame-level handling of CS in ASR. Watanabe et al. 2018 introduced a language-independent E2E ASR architecture capable of joint frame-level LID and ASR in simulated CS data. Luo et al. 2018 have introduced advanced E2E models specifically for CS that can handle spontaneous switches by introducing in-prompt switch tokens among the predicted ASR tokens. Other works, where token (e.g. word) LIDs were utilised to improve CS ASR recognition include Zeng et al. 2019; Seki et al. 2019; Mary N J et al. 2020; Liu et al. 2021b, 2023; Wang and Li 2023. However, token LIDs do not consider the distribution of tokens based on their explicit grammatical role. One could argue that such a linguistically informed token LID model can be implicitly learnt from text-derived LID tokens and corresponding audio, but for this to be true large amounts of CS ASR data are required, which cannot be achieved as of today.

The above demonstrates that the impact of acoustic MLID classification on the performance of ASR systems has not been studied. Although knowing

the MLID of an utterance increases the likelihood of recognising ML system morphemes (Myers-Scotton, 2002), this information has not been explicitly leveraged to improve code-switching ASR performance. Other ideas from the MLF theory have been used in CS Language Modelling (LM) for ASR, for example text augmentation (Yilmaz et al., 2018; Lee et al., 2019), using additional grammatical information during LM construction (Adel et al., 2015; Soto and Hirschberg, 2019) or including a separate MLF-inspired loss function in an E2E model during monolingual training (Chang et al., 2019; Lee and Li, 2020). This paper aims to fill the gaps outlined above by investigating how MLID can enhance the effectiveness of ASR systems in recognising CS speech.

3 CS ASR using MLF theory

3.1 P1.2: The Morpheme Order Principle

The original definition of the Morpheme Order Principle (described in Section 1) includes two separate features of the ML. Therefore, one can derive two separate principles for ML determination based on the Morpheme Order Principle:

> P1.1) ML provides context for singly occurring words,
>
> P1.2) ML provides the morpheme order for the utterance.

In this paper the second part of the Morpheme Order Principle namely P1.2 is implemented for multiple languages which states that the morpheme order comes from the ML. For example, in "你觉得我们speak clear enough 吗" a translation of the auxiliary Mandarin verb 吗 will never appear at the end of an utterance in English, signifying that Mandarin is ML in this utterance.

P1.2 can be defined formally for several languages as follows. Assume that the languages $(L_1, ..., L_N) \subset L$ are present in a CS utterance $\mathbf{y}$, then $\mathbf{y}$ can be translated into monolingual utterances $(\hat{\mathbf{y}}_{L_1}, ..., \hat{\mathbf{y}}_{L_N})$ by translating the words to words of the other constituent language. $(\hat{\mathbf{y}}_{L_1}, ..., \hat{\mathbf{y}}_{L_N})$ are obtained from the original utterance $\mathbf{y}$. Consider a probability of an utterance $P(\mathbf{y}|L)$ given the language L, then to determine the more likely transcription having languages $(L_1, ..., L_N)$ one can define the following decision function:

$$\text{ML} = \underset{L \in \{L_1, ..., L_N\}}{\arg\max} \; P(\hat{\mathbf{y}}_L|L) \tag{5}$$

3.2 MTL ASR with MLID

Given a recording of CS utterance $\mathbf{X}$ with a corresponding transcription $\mathbf{y}$, then an E2E model may be used to approximate a speech recognition process: $P(\mathbf{y}|\mathbf{X}, \Theta)$, where Θ are model parameters.

Auxiliary tasks may be added to the ASR model, making it a Multi-Task Learning (MTL) training pipeline. LID conditioning in a component-based system (Liu et al., 2021a) or MTL with LID prediction (Chen et al., 2023) is commonly used in multilingual ASR since the probability of an utter-

$$\Theta^*_{\text{joint}}, \Theta^*_{\text{ASR}}, \Theta^*_{\text{LID}} = \underset{\Theta_{\text{joint}}, \Theta_{\text{ASR}}, \Theta_{\text{LID}}}{\arg\max} \sum_{i=1}^{|D|} (\log P(y_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{ASR}}) + \log P(L_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{LID}})) \tag{1}$$

$$\Theta^*_{\text{joint}}, \Theta^*_{\text{ASR}}, \Theta^*_{\text{tLID}} = \underset{\Theta_{\text{joint}}, \Theta_{\text{ASR}}, \Theta_{\text{tLID}}}{\arg\max} \sum_{i=1}^{|D|} (\log P(y_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{ASR}}) + \log P(l_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{tLID}})) \tag{2}$$

$$\begin{aligned}\Theta^*_{\text{joint}}, \Theta^*_{\text{ASR}}, \Theta^*_{\text{MLID}}, \Theta^*_{\text{tLID}} = \underset{\Theta_{\text{joint}}, \Theta_{\text{ASR}}, \Theta_{\text{MLID}}, \Theta_{\text{tLID}}}{\arg\max} \sum_{i=1}^{|D|} (&\log P(y_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{ASR}}) \\ &+ \log P(L_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{MLID}}) + \log P(l_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{tLID}}))\end{aligned} \tag{3}$$

$$\begin{aligned}\Theta^*_{\text{joint}}, \Theta^*_{\text{ASR}}, \Theta^*_{\text{MLID}}, \Theta^*_{\text{tLID}} = \underset{\Theta_{\text{joint}}, \Theta_{\text{ASR}}, \Theta_{\text{MLID}}, \Theta_{\text{tLID}}}{\arg\max} \sum_{i=1}^{|D|} (&\log P(y_i|\mathbf{X}_i, \hat{L}_i, \hat{l}_i, \Theta_{\text{joint}}, \Theta_{\text{ASR}}) \\ &+ \log P(L_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{MLID}}) + \log P(l_i|\mathbf{X}_i, \Theta_{\text{joint}}, \Theta_{\text{tLID}}))\end{aligned} \tag{4}$$

ance is different depending on the language spoken in the utterance $P(\mathbf{y}|L_1) \neq P(\mathbf{y}|L_2)$. If D is the training dataset, a multilingual ASR system trained with an auxiliary LID task has the joint objective as shown in Equation 1. In Equation 1 Θ_{ASR} are the parameters of the model which produces the ASR output, Θ_{LID} are the parameters used to produce the LID output, and Θ_{joint} are the parameters contributing to the production of both ASR and LID output.

Similarly to the MTL with utterance LID, MTL with token LID prediction (Liu et al., 2023) may be used in CS ASR. Having ASR as the main task and token LID as an auxiliary task, an MTL objective may be defined in Equation 2 where l_i are the token LID tags and Θ_{tLID} are the parameters only contributing to the token LID output.

Consequently, an MLID classifier can be used to further enhance CS ASR recognition quality in a low-resource setting. There is no existing ML labelled data for CS speech recordings, but MLID labels can be obtained from target transcriptions using the implementations of the principles for ML determination from the MLF theory (Iakovenko and Hain, 2024). Having determined the MLID for the data the following MTL objective may be formulated and displayed in Equation 3, where Θ_{MLID} are the parameters deriving the MLID output.

Suppose $\hat{l}_i$ and $\hat{L}_i$ are the predicted token LID sequence and MLID for audio $\mathbf{X}_i$, then the ASR probability may be conditioned by the predicted MLID and token LID. The final objective is shown in Equation 4.

4 Experiments

4.1 Datasets

The experiments are carried out on a dataset of spontaneous Singaporean speech SEAME (Lyu et al., 2010). The SEAME dataset was collected in Singapore and Malaysia, with data recorded by Nanyang Technological University and Universiti Sains Malaysia, respectively. The recordings encompass two distinct speaking styles: conversational and interview-based speech. In the conversational sessions, the speech of each participant was recorded separately, covering a range of informal topics such as hobbies, friendships, and daily activities. In contrast, the interview recordings include only the responses of individual interviewees. The speakers were between 19 and 33 years of age with a nearly equal gender distribution (49.7% female, 50.3% male). In total, the dataset comprises 156 distinct speakers, with 36.8% from Malaysia and the remainder from Singapore. The dataset includes 192 hours of audio recordings, 110037 utterances and 1449737 transcribed words. The dataset is split into train, validation, and two test sets devman and devsge commonly used for CS ASR benchmarking (Zeng et al., 2019). The summary of the data splits is shown in Table 1.

4.2 Model

The ASR model employs a Conformer encoder (Gulati et al., 2020) with 12 blocks and a Transformer decoder (Vaswani et al., 2017) comprising 6 blocks. Each block contains 4 attention heads and 2048-dimensional feedforward layers. The system is trained using a joint loss combining attention-based cross-entropy and CTC loss (Kim et al., 2017), along with an additional intermediate CTC loss. Following (Chen et al., 2023), the weights for the CTC and intermediate CTC losses are both set to 0.3. Auxiliary tasks, such as token LID, utterance LID and utterance MLID, are introduced by the intermediate CTC losses. English tokens are segmented using 3,000 Byte Pair Encoding (BPE) units, while Mandarin tokens are represented at the character level, comprising 2,622 units. The model is trained with the Adam optimiser (Kingma and Ba, 2014), beginning with a learning rate of 0.001 and employing 25,000 warm-up steps. Training is conducted over 60 epochs, and the final model is obtained by averaging the parameters of the top 10 checkpoints based on validation performance. All experiments are conducted using the ESPnet toolkit (Watanabe et al., 2018) on four NVIDIA RTX 3060 GPUs over a period of three days.

When the above model is trained using an MTL setup, additional output layers are defined after the 5th and 6th encoder blocks. The auxiliary output layers perform tasks of either predicting a sequence of token LIDs or predicting a single LID for the whole utterance. The implemented setups are summarised in Figure 1. b), c) and d) setups from the figure will be later distinguished from others using a **-mtl* suffix.

An alternative way of including an utterance LID during training is appending the LID at the beginning of the utterance transcription (Table 3 lines 2 and 3). This setup will be later referred to using a **-prompt* suffix.

Table 1: Dataset splits used for CS ASR training.

	unit	train	validation	devman	devsge
Total size	hours	96.4	5	7.49	3.93
	utterances	89364	4704	6531	5321
Speakers	n speakers	134	134	10	10
Average utterance length	words	13.9	13.8	14.8	10.2
	seconds	3.9	3.9	4.1	2.7

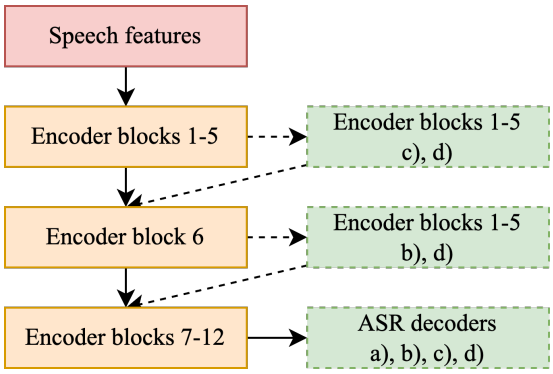

Figure 1: The outline of the ASR models used for experiments. The decoders vary in different implementations: a) The baseline implementation includes only the ASR decoder; b) An token LID decoder is defined alongside the ASR decoders where apart from recognising a sequence of tokens the model is trained to predict a sequence of token LIDs; c) An LID/MLID decoder which performs classification of the entire utterance is performed simultaneously with the main ASR task; d) All decoders are used to perform ASR, token LID and LID/MLID tasks simultaneously. The dotted arrow from the additional decoders to the encoders signifies the concatenation of the outputs of the encoder with the hidden representations.

4.3 Evaluation

This work employs several evaluation methods to quantify differences in recognition performance. All methods are based on the Levenshtein distance between token sequences, normalised by the number of tokens in the reference. However, each method applies a different tokenisation strategy, thereby emphasising different types of discrepancies between predicted and reference utterances. A summary of these evaluation methods and their respective tokenisations is provided in Table 2.

In addition to Levenshtein-based evaluation metrics, Sentence Error Rate (SER) is used to quantify the proportion of sentences that contain recognition errors; it is the complement of sentence accuracy. The quality of utterance classification is assessed using accuracy, defined as the proportion of utterances whose predicted class matches the ground truth.

Table 2: Evaluation methods summary. "-" means that the metric is not estimated for the language in the experiments.

Error rate type	Acronym	Tokens English	Tokens Mandarin
Word	WER	words	-
Character	CER	letters	characters
Mixed	MER	words	characters

For LID evaluation, Token Error Rate (TER) is used to measure the accuracy of token-level LID predictions produced by the additional decoder component (denoted as d in Figure 1). At the utterance level, two types of SER are considered:

1. The proportion of utterances in which the entire sequence of token-level LID labels is incorrectly predicted, and

2. The proportion of utterances in which the predicted utterance-level LID or MLID, generated by component c) in Figure 1, does not match the ground truth.

4.4 Auxiliary task definitions

In a multi-task learning (MTL) framework, auxiliary tasks are secondary objectives learned in parallel with the primary task. In this study, the primary task is automatic speech recognition (ASR), while the auxiliary tasks include token-level language identification (LID), utterance-level LID, and matrix language identification (MLID). Auxiliary tasks are particularly beneficial in scenarios with limited training data, as they help guide the optimisation process. Specifically, they provide additional supervision signals that steer the model's gradients toward more generalizable solutions and help avoid convergence to suboptimal local minima.

This subsection outlines the procedures used to obtain labels for the auxiliary tasks. All label generation methods are either derived from the textual transcriptions, such as those for token LID and MLID, or sourced directly from the SEAME corpus, as in the case of utterance-level LID.

4.4.1 Token LID

Following Figure 1, an token LID decoder may be defined which predicts a sequence of token LIDs for an utterance. The LID of the tokens is determined based on the script the words are written in similar to Wang and Li 2023. An example of the sequence of tokens is given in Table 3 in line 4.

4.4.2 LID

Apart from a token LID component, an additional LID task may be defined which predicts a monolingual language ID of an utterance from the SEAME dataset. CS between utterances (inter-sentential CS) is common for CS: 48% of isolated SEAME utterances are monolingual, so for those LID can be defined. The remaining CS utterances may be regarded as a separate language by introducing a separate LID "<cs>". A demonstration of such classification is given in Table 3 in line 5.

Table 3: Example target labels for a sentence "okay kay 让我拿出我的calculator". a), b), c) and d) refer to the model structure specified in Figure 1.

Decoder type	**Decoder output**
ASR (a,b,c,d)	okay kay 让 我 拿 出 我 的calculator
ASR with LID prompt (a)	<cs> okay kay 让我拿出我的calculator
ASR with MLID prompt (a)	<zh> okay kay 让我拿出我的calculator
token LID (b,d)	<en> <en> <zh> <zh> <zh> <zh> <zh> <zh> <en>
LID (c,d)	<cs>
MLID (c,d)	<zh>

4.4.3 MLID

Assume that in a monolingual utterance the MLID is equal to LID, this way all utterances in a CS dataset may be classified as one of the two mixed languages ("<zh>" or "<en>"). MLID is derived from the two principles for ML determination and uses the following rules for accurately identifying the MLID from text:

- P1.1) ML is the language that provides the context for the singleton insertions from the EL;
- P1.2) ML provides the word order for the CS utterance;
- P2) ML provides the system morphemes for the CS utterance.

P1.2 is described in Section 3 and the other two rules (P1.1 and P2) are explained in greater detail in Iakovenko and Hain 2024. To provide the most accuracy and coverage of the principles, they are applied jointly in the following order of priority: P1.1 > P2 > P1.2.

4.5 Token LID, LID and MLID in CS ASR

The summary of the results is presented in Table 4. *no-lid* is the baseline implementation with no additional LID losses, *lid-** are the implementations with token LID and monolingual utterance LIDs following Liu et al. 2023, *ml-** are the implementations with token LID and utterance MLIDs. MER is primarily discussed in this section since it is a preferred metric in English/Mandarin CS ASR research (Vu et al., 2012).

The table shows that any way of incorporating LID information is beneficial for ASR training which is known from previous research (Section 2): *no-lid* getting the lowest total MER of 20.2%. Introducing the utterance LID or MLID in prompt (*lid-utt-prompt* and *ml-utt-prompt* respectively) does improve ASR performance in comparison to the baseline (20% MER for both implementations) but using MLID in an MTL setup (*ml-utt-mtl*) is more effective achieving 19.8% MER. Interestingly, the ASR quality is the same for setups when using token LID (*lid-tok-mtl*) and MLID (*ml-utt-mtl*) as auxiliary tasks, despite token LID providing more detailed information about the target text. Finally, the performance may be further improved by training the ASR system with both MLID and token LID auxiliary tasks (*ml-utt-tok-mtl*) yielding the best result of 19.7% MER. Although the changes in performance are subtle, they are statistically significant: p-value test for CS utterances from devman and devsge reveals a significant difference at the level of p=0.05 between *lid-utt-tok-mtl* and *ml-utt-tok-mtl*.

Table 4: Model performance for all implementations. a), b), c) and d) refer to the model structure specified in Figure 1. Metric definitions are given in Section 4.3 and 2. *no-lid* is the baseline implementation with no additional LID losses, *lid-** are the implementations with token LID and monolingual utterance LIDs, *ml-** are the implementations with token LID and utterance MLIDs.

Metric	MER			SER		CER	WER
Subset / Model	devman	devsge	all	devman	devsge	Mandarin	English
a) *no-lid*	16.8	23.5	20.2 75.3	74.7	18.6	29.2	
a) *lid-utt-prompt*	16.6	23.4	20	75.4	74.4	18.4	28.8
b) *lid-tok-mtl*	16.4	23.1	19.8	75.5	73.8	18.2	28.7
c) *lid-utt-mtl*	16.5	23.5	20	75.4	74.8	18.3	29
d) *lid-utt-tok-mtl*	16.5	23.2	19.9	75.1	**73.6**	**17.9**	28.9
a) *ml-utt-prompt*	16.6	23.3	20	75.5	74.5	18.1	29
c) *ml-utt-mtl*	**16.3**	23.2	19.8	**75**	74.1	18	28.7
d) *ml-utt-tok-mtl*	16.4	**22.9**	**19.7**	**75**	73.8	18	**28.6**

4.6 Language disambiguation performance

The performance of token LID, utterance LID and MLID tasks is also measured in the *lid-utt-tok-mtl* and *ml-utt-tok-mtl* setups. Token LID is assessed using TER and achieves 9.7% in *lid-utt-tok-mtl* and 9.8% in the *ml-utt-tok-mtl* setup which are almost identical values. Having calculated the token-level errors it is possible to also compute SER of the token LID task similarly to ASR, leading to 68.5% SER for *lid-utt-tok-mtl* and 67.8% SER for *lid-utt-tok-mtl*. Token LID TER and SER are similar in both setups which allows us to fairly assess the impact of the utterance LIDs and MLIDs on the ASR task. If token LID module is used to calculate utterance LID based on the recognised tokens the overall LID accuracy is 90.8%.

For the separate utterance classification decoders accuracy of LID prediction is 86.1% in the *lid-utt-tok-mtl* setup and the accuracy of MLID prediction is 71.2% in the *ml-utt-tok-mtl* setup. Despite the high SER values solving these classification tasks alongside ASR and token LID brings improvement to the ASR performance thus also highlighting the necessity for further research in this area.

5 Discussion

Given the results above, it can be concluded that MLID is the most advantageous approach for classifying CS utterances in CS ASR systems. This is primarily because MLID provides more nuanced information than monolingual LID. While monolingual LID merely indicates that an utterance belongs to a single language, MLID additionally signals that the grammatical structure of the utterance aligns with the ML. This grammatical alignment imposes constraints on the types of EL tokens that are likely to occur. For instance, EL function words are less likely to be predicted. This property makes it possible to jointly model utterance-level classification and token-level LID, leading to improvements in overall CS ASR performance.

To explore the impact of MLID on function word recognition, POS tags were assigned to the ASR outputs using a monolingual POS tagging approach described in (AlGhamdi et al., 2016). POS tagging was applied to outputs from two configurations: *lid-utt-tok-mtl* and *ml-utt-tok-mtl*, using the SpaCy toolkit. Following the definition in (Bullock et al., 2018), function words were identified as those tagged as auxiliaries, determiners, coordinating conjunctions, or subordinating conjunctions. The number of function words was then counted for each configuration and grouped by the MLIDs assigned by the MLID decoder in the *ml-utt-tok-mtl* setup (see Table 5).

Analysis of the results shows that for utterances labelled with English MLID, the proportion of English function words remained relatively stable, with a slight decrease from 18.15% (3354 words) in the *lid-utt-tok-mtl* setup to 18.13% (3352 words) in the *ml-utt-tok-mtl* setup. In contrast, utterances with Mandarin MLID exhibited a modest increase in the probability of Mandarin function word usage: from 18.48% (3416 words) to 18.66% (3450 words). This suggests that incorporating MLID not only improves classification but may also better preserve function word usage patterns in line with the ML.

Table 5: Predicted function words counts: absolute and relative. Relative is normalised by the number of words in the dataset.

	MLID = English		MLID = Mandarin	
Model	Absolute count	Relative count	Absolute count	Relative count
lid-utt-tok-mtl	3354	18.15%	3416	18.48%
ml-utt-tok-mtl	3352	18.13%	3450	18.66%

6 Conclusion

This paper presented a novel approach to CS ASR by introducing the concept of MLID into an MTL framework. To show the effectiveness of the method, the MLID-based approach was compared to several LID-based CS ASR approaches used in SotA CS ASR models. Drawing from the MLF theory, additionally determining the grammatical structure of CS utterances was shown to improve ASR accuracy. In particular, incorporating MLID as an auxiliary task led to a measurable reduction in MER compared to strong LID-based baselines, indicating the value of distinguishing the dominant grammatical language in CS speech. Using text-derived MLID as an auxiliary task in an MTL ASR setup led to a MER decrease from 20.2% in an Attention-CTC ASR baseline to 19.7%. The comparison to the standard CS ASR frameworks revealed that performing MLID in the MTL CS ASR setup is just as effective as performing token LID. Furthermore, qualitative analysis revealed that MLID prediction facilitates the recognition of function words from the ML, suggesting that MLID helps the model better capture the syntactic structure of utterances. These findings show that linguistic theories such as MLF can be meaningfully integrated into E2E ASR systems to address the unique challenges posed by CS.

While the above method improves CS ASR performance, it requires accurately identified MLIDs. Therefore, future work will focus on incorporating grammatical information directly through setting the backpropagation weights dependent on grammatical function of the word or morpheme. Furthermore, the MLID-informed ASR systems should be extended to other language pairs in the future, such as South African or Indian languages.

Limitations

The limitation of the following approach, as highlighted in the conclusion, is that it was only tested for a single language pair. Given that the MLF theory does not accurately model all types of CS, it could encounter difficulties when processing and training on African or Indian data. Furthermore, the approach requires finely defined MLIDs using accurate textual MLID approaches or manual annotation, which may be difficult to obtain. Finally, the changes in the recognition performance, although statistically significant, are marginal, which may not be useful in certain applications.

References

Heike Adel, Ngoc Thang Vu, Katrin Kirchhoff, Dominic Telaar, and Tanja Schultz. 2015. Syntactic and semantic features for code-switching factored language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:431–440.

Fahad AlGhamdi, Giovanni Molina, Mona Diab, Thamar Solorio, Abdelati Hawwari, Victor Soto, and Julia Hirschberg. 2016. Part of speech tagging for code switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107, Austin, Texas. Association for Computational Linguistics.

Barbara Bullock, Wally Guzmán, Jacqueline Serigos, Vivek Sharath, and Almeida Jacqueline Toribio. 2018. Predicting the presence of a matrix language in code-switching. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 68–75, Melbourne, Australia. Association for Computational Linguistics.

Ching-Ting Chang, Shun-Po Chuang, and Hung yi Lee. 2019. Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In *INTERSPEECH*.

William Chen, Brian Yan, Jiatong Shi, Yifan Peng, Soumi Maiti, and Shinji Watanabe. 2023. Improving massively multilingual asr with auxiliary ctc objectives. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Anuj Diwan, Rakesh Vaideeswaran, Sanket Shah, Ankita Singh, Srinivasa Raghavan, Shreya Khare, Vinit Unni, Saurabh Vyas, Akash Rajpuria, Chiranjeevi Yarra, Ashish Mittal, Prasanta Ghosh, Preethi Jyothi, Kalika Bali, Vivek Seshadri, Sunayana

Sitaram, Samarth Bharadwaj, Jai Nanavati, Raoul Nanavati, and Karthik Sankaranarayanan. 2021. Mucs 2021: Multilingual and code-switching asr challenges for low resource indian languages. pages 2446–2450.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

Olga Iakovenko and Thomas Hain. 2024. Methods of automatic matrix language determination for code-switched speech. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5791–5800, Miami, Florida, USA. Association for Computational Linguistics.

Yerbolat Khassanov, Haihua Xu, Van Tung Pham, Zhiping Zeng, Eng Siong Chng, Chongjia Ni, and Bin Ma. 2019. Constrained Output Embeddings for End-to-End Code-Switching Speech Recognition with Only Monolingual Data. In *Proc. Interspeech 2019*, pages 2160–2164.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Grandee Lee and Haizhou Li. 2020. Modeling code-switch languages using bilingual parallel corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 860–870.

Grandee Lee, Xianghu Yue, and Haizhou Li. 2019. Linguistically motivated parallel data augmentation for code-switch language modeling. In *INTERSPEECH*.

Ke Li, Jinyu Li, Guoli Ye, Rui Zhao, and Yifan Gong. 2019. Towards code-switching asr for end-to-end ctc models. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6076–6080.

Danyang Liu, Ji Xu, Pengyuan Zhang, and Yonghong Yan. 2021a. A unified system for multilingual speech recognition and language identification. *Speech Communication*, 127:17–28.

Hexin Liu, Leibny Paola Garcia Perera, Xinyi Zhang, Justin Dauwels, Andy WH Khong, Sanjeev Khudanpur, and Suzy J Styles. 2021b. End-to-end language diarization for bilingual code-switching speech. In *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, pages 866–870. International Speech Communication Association.

Hexin Liu, Haihua Xu, Leibny Paola Garcia, Andy W. H. Khong, Yi He, and Sanjeev Khudanpur. 2023. Reducing language confusion for code-switching speech recognition with token-level language diarization. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Ne Luo, Dongwei Jiang, Shuaijiang Zhao, Caixia Gong, Wei Zou, and Xiangang Li. 2018. Towards end-to-end code-switching speech recognition. *arXiv preprint arXiv:1810.13091*.

Dau-Cheng Lyu, Tien Ping Tan, Chng Eng Siong, and Haizhou Li. 2010. Seame: a mandarin-english code-switching speech corpus in south-east asia. In *INTERSPEECH*.

Bin Ma, Cuntai Guan, Haizhou Li, and Chin-Hui Lee. 2002. Multilingual speech recognition with language identification. In *INTERSPEECH*, pages 505–508.

Metilda Sagaya Mary N J, Vishwas M. Shetty, and S. Umesh. 2020. Investigation of methods to improve the recognition performance of tamil-english code-switched data in transformer framework. *ICASSP 2020*, pages 7889–7893.

C. Myers-Scotton. 1993. *Duelling Languages: Grammatical Structure in Codeswitching*. Clarendon Press.

Carol Myers-Scotton. 2002. *Contact linguistics: Bilingual encounters and grammatical outcomes*. OUP.

SOS Ncoko, Ruksana Osman, and Kate Cockcroft. 2000. Codeswitching among multilingual learners in primary schools in south africa: An exploratory study. *International Journal of Bilingual Education and Bilingualism*, 3(4):225–241.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Madaki Rufai Omar. 1983. *A linguistic and pragmatic analysis of Hausa-English code-switching (Nigeria)*. University of Michigan.

Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Jonathan Le Roux, and John R. Hershey. 2019. End-to-end multilingual multi-speaker speech recognition. In *INTERSPEECH*.

Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie. 2019. Investigating end-to-end speech recognition for mandarin-english code-switching. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6056–6060.

Víctor Soto and Julia Hirschberg. 2019. Improving code-switched language modeling performance using cognate features. In *INTERSPEECH*.

Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro J. Moreno, Eugene Weinstein, and Kanishka Rao. 2017. Multilingual speech recognition with a single end-to-end model. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4904–4908.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A first speech recognition system for mandarin-english code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892.

Qinyi Wang and Haizhou Li. 2023. Text-derived language identity incorporation for end-to-end code-switching speech recognition. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 33–42, Singapore. Association for Computational Linguistics.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, and 1 others. 2018. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*.

Christopher M White, Sanjeev Khudanpur, and James K Baker. 2008. An investigation of acoustic models for multilingual code-switching. In *INTERSPEECH*, pages 2691–2694.

Emre Yilmaz, Astik Biswas, Ewald van der Westhuizen, Febe de Wet, and Thomas R. Niesler. 2018. Building a unified code-switching asr system for south african languages. In *Interspeech*.

Emre Yılmaz, Henk van den Heuvel, and David Van Leeuwen. 2016. Investigating bilingual deep neural networks for automatic recognition of code-switching frisian speech. *Procedia Computer Science*, 81:159–166.

Zhiping Zeng, Yerbolat Khassanov, Van Tung Pham, Haihua Xu, Eng Siong Chng, and Haizhou Li. 2019. On the End-to-End Solution to Mandarin-English Code-Switching Speech Recognition. In *Proc. Interspeech 2019*, pages 2165–2169.

Yu Zhang, Ekapol Chuangsuwanich, and James Glass. 2014. Language id-based training of multilingual stacked bottleneck features. In *Proc. Interspeech*, pages 1–5. Citeseer.