Code-Switching Speech Recognition

Souvik Maji Arjun Bhattad
Indian Institute of Technology Jodhpur
{b22cs089, b22ai051}@iitj.ac.in

Abstract

This report introduces an in-depth analysis of code-switching automatic speech recognition (ASR), a challenge in which speakers switch between two or more languages within a single utterance. The real-world significance of code-switching ASR is examined, along with an evaluation of state-of-the-art (SOTA) techniques and models developed to address this challenge. Various existing models are compared, analyzing their strengths, limitations, and evaluation methodologies. Key challenges such as language-specific information fusion, tokenizer scalability, and language bias are discussed, along with open research questions and future directions in the field.

1. Introduction

Speech recognition technology has advanced significantly in recent years, driving virtual assistants, enabling automated transcription, and enhancing countless human–machine interactions. However, one particularly complex and intriguing challenge within this field is code-switching speech recognition—the task of transcribing speech that fluidly alternates between two or more languages within a single utterance or conversation. Code-switching is prevalent in multilingual communities worldwide, making it a crucial area of research in speech and language technologies.

2. Importance in the Modern World

The phenomenon of code-switching is no longer limited to casual conversations; it has become prevalent in formal and institutional settings, including business negotiations, academic discussions, healthcare interactions, and government communications. As global networks continue to expand, multilingual societies have become more common, with speakers frequently alternating between languages for greater expressive power, clarity, and efficiency. Recognizing and adapting to this linguistic behavior is essential for creating **inclusive, effective, and practical speech recognition technologies**.

2.1. Enhancing User Experience Across Diverse Populations

In multilingual regions such as **India, Singapore, Canada, and parts of Latin America and Europe**, people grow up speaking multiple languages. Their natural speech often includes **rapid shifts between languages**, reflecting the **cultural and linguistic diversity** of their communities. Traditional monolingual speech recognition systems fail to capture this complexity, leading to inaccuracies and frustration. By supporting **code-switching ASR**, users can communicate naturally without needing to suppress their multilingual identity, resulting in a **more seamless and authentic experience**.

Figure 1. Multilingual ASR

2.2. Improving Customer Service and Support

Businesses and multinational corporations frequently interact with customers who switch languages mid-conversation. This is especially common in **call centers and customer service interactions**, where users may express themselves in a mix of languages for clarity. If speech recognition technology fails to handle these transitions, **critical details may be lost or misinterpreted**, leading to customer dissatisfaction and increased errors. By integrating **robust code-switching ASR**, businesses can provide **more efficient, accurate, and culturally aware customer support**.

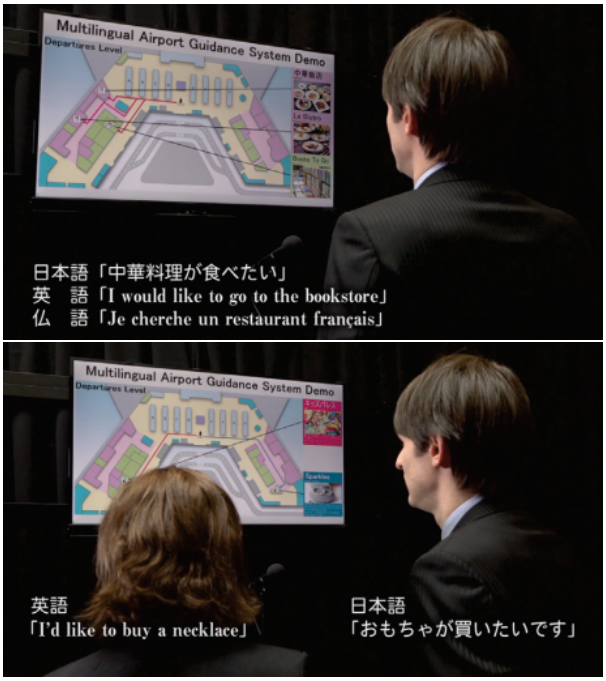

Figure 2. *Example of seamless ASR on code-switching speech (top) and multilingual multi-speaker speech (bottom).*

2.3. Expanding Accessibility and Inclusivity

For individuals who rely on assistive technologies—such as those with **visual impairments or motor disabilities**—the ability to interact with speech recognition systems in a **multilingual** manner is essential. A lack of code-switching support can create unnecessary barriers, **forcing users into artificial monolingual constraints**. Developing **inclusive ASR systems** allows more people to engage with technology **naturally and comfortably**, improving accessibility across diverse user groups.

2.4. Catalyzing Growth in Media and Entertainment

The **media and entertainment industry** increasingly produces content that blends multiple languages, whether in **films, TV shows, music, or digital streaming platforms**. Automatic **subtitling and transcription** for multilingual content become significantly more challenging when **code-switching** occurs. Advanced **ASR systems capable of recognizing mixed-language speech** ensure more accurate subtitles and translations, helping global audiences **fully engage with multilingual content**. This is also crucial for **social media**, where content creators frequently switch between languages to **connect with a broader audience**.

2.5. Supporting Language Preservation and Revitalization

In many communities, **code-switching involves minority or endangered languages**. If speech recognition technology can accurately capture and process **these languages within code-switched speech**, it can aid in **documentation and preservation**. Providing technological support for these languages **encourages their continued use**, ensuring their survival for **future generations** and promoting **cultural heritage preservation**.

2.6. Empowering Education and Cross-Cultural Learning

Bilingual education programs often incorporate **code-switching** as a pedagogical tool to help students grasp complex concepts in a way that aligns with their linguistic background. ASR systems that support **mixed-language input** can enhance **automated note-taking, classroom captions, and real-time translation**, making education more **accessible and engaging**. This allows learners to **fully leverage their multilingual abilities**, leading to **deeper comprehension and greater participation**.

2.7. Advancing Linguistic and Social Research

Beyond practical applications, accurate transcription and analysis of **code-switched speech** provide valuable data for **linguistic, sociological, and anthropological research**. Patterns of **code-switching** reveal insights into **language evolution, cultural exchange, and identity formation**. High-quality **speech datasets** with **code-switching ASR** also contribute to advancements in **Natural Language Processing (NLP)**, improving overall language modeling in AI applications.

2.8. Driving Innovation in AI and NLP

Code-switching lies at the **intersection of multiple AI research challenges**, including **multilingual language modeling, acoustic modeling, and speaker adaptation**. Developing **robust AI models** that perform well on code-switched speech **pushes the boundaries of speech and language processing**, leading to innovations in **machine translation, cross-lingual information retrieval, and multilingual AI assistants**.

3. Related Models and Their Limitations

3.1. Early Approaches to Code-Switching ASR

The development of early code-switching (CS) ASR systems was hindered by the scarcity of high-quality CS speech corpora. Initial methods relied on rule-based and statistical models, which required separate language models and phoneme-level adaptation for each language.

3.1.1. Acoustic and Language Model Adaptation (1997–2015)

Separate ASR models were developed for each language, utilizing frame-level language identification (LID) to switch between models [6, 7]. Weighted Finite-State Transducers (WFSTs) were introduced to improve the integration of multiple ASR systems [8]. Statistical Language Models (SLMs) were trained on limited CS text corpora, but they struggled with syntactic and semantic generalization [9].

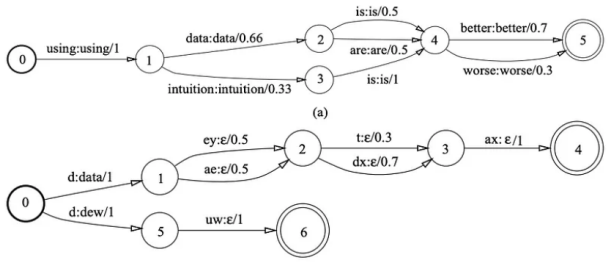

Figure 3. An illustration of WFST.

Limitations:

- Required explicit language segmentation, making real-time ASR impractical.
- Performed poorly on intra-sentential code-switching, as language boundaries were often ambiguous.
- Suffered from high Word Error Rate (WER) due to a lack of CS-specific training data.

3.2. End-to-End Neural Models for CS ASR

With the rise of deep learning, ASR systems transitioned from pipeline-based architectures to end-to-end (E2E) neural models, eliminating the need for separate acoustic and language models.

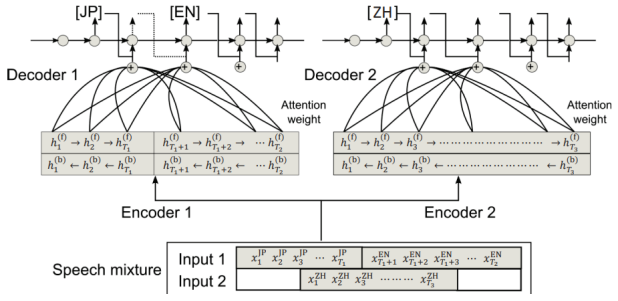

Figure 4. Example Neural Model for CS ASR

3.2.1. End-to-End Multilingual ASR (2015–2021)

Connectionist Temporal Classification (CTC) and Recurrent Neural Network Transducer (RNNT)-based models replaced traditional hybrid ASR architectures [10, 11]. Transformer-based ASR models introduced self-attention mechanisms to better capture long-range dependencies and context switching [12]. Multilingual ASR models were trained on hundreds of languages using joint phoneme-based tokenization, allowing models to generalize better across language pairs [13]. Language ID-based tokenization incorporated special tokens (e.g., [EN], [ES]) to signal language transitions during decoding [12, 14].

Limitations:

- Poor generalization to unseen language pairs, as most models were trained primarily on large monolingual datasets.
- Suboptimal LID performance, often requiring external post-processing to determine token-level language identity.

3.3. Synthetic Data Augmentation for Code-Switching

Due to the scarcity of code-switching data, researchers developed synthetic data augmentation techniques to improve model training.

3.3.1. Text-Based Code-Switching Data Synthesis

Parallel translation models generated synthetic CS text by randomly switching words in bilingual sentences [15]. Neural Machine Translation (NMT)-based synthesis introduced code-switching transitions within bilingual corpora [16]. Probabilistic code-switching models simulated more naturalistic CS patterns [17].

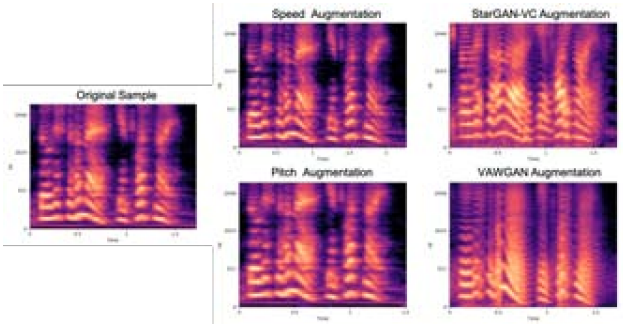

Figure 5. Data Augmentation for ASR.

Limitations:

- Generated unnatural code-switching patterns that did not align with real-world linguistic structures.
- Required manual selection of language pairs, limiting scalability and generalization.

3.3.2. Speech-Based Synthetic Code-Switching

Segment-level speech concatenation combined monolingual speech segments and applied Voice Activity Detection (VAD) for smoothing [18]. Phoneme-based CS synthesis modified individual phonemes to create more realistic CS speech samples [19].

Limitations:

- Introduced synthetic artifacts such as abrupt amplitude changes and inconsistent background noise.
- Used fixed switch points, limiting the model's ability to capture natural code-switching dynamics.

4. State-of-the-Art Model

The state-of-the-art (SotA) approach to Code-Switching ASR is based on the work of Dhawan et al. (2023) [1]. Their model leverages a Conformer-RNNT architecture and an innovative Concatenated Tokenizer , improving both ASR accuracy and language identification in code-switching settings.

4.1. Strengths of Cutting-Edge Models in Code-Switching ASR

4.1.1. Model Architecture Specifications

Conformer-RNNT Model The primary ASR model used in this study is the Conformer-RNNT Large, which integrates:

- **Transformers**: Capture long-range dependencies in speech.
- **Convolutional Neural Networks (CNNs)**: Efficiently model local patterns in audio.
- **Recurrent Neural Transducers (RNNTs)**: Synchronize input speech with output sequences in a streaming-friendly manner.

No External Language Model (LM) The model operates without an external LM, relying solely on the acoustic and linguistic information learned during training. This makes tokenization a crucial factor in achieving high accuracy.

Tokenization Strategies Tokenization significantly impacts ASR performance:

- **Monolingual Tokenizers**: Designed for a single language using sub-word units (e.g., SentencePiece).
- **Aggregated Tokenizers**: Trained on a multilingual corpus but sacrifice language-specific information.
- **Concatenated Tokenizers**:
  - Utilize existing monolingual tokenizers while maintaining non-overlapping, distinct token ID spaces.
  - Example: English tokens (0-1023), Spanish tokens (1024-...).
  - Enable both transcription and automatic language identification at the token level.

4.2. Concatenated Tokenizer in Depth

4.2.1. Key Features

- **Distinct Token ID Spaces**: Preserve language identity at the token level.

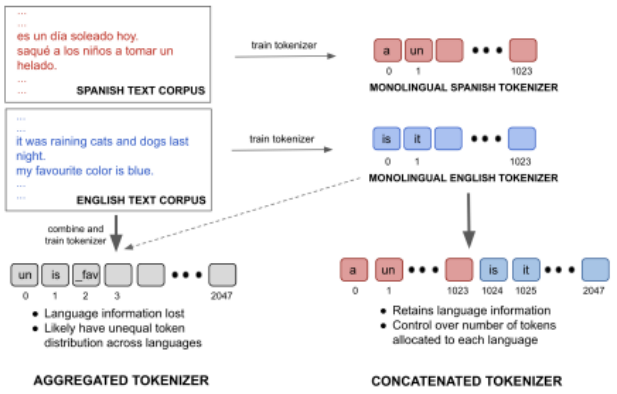

Figure 6. Aggregated vs Concatenated tokenization approaches for a bilingual English-Spanish example. Spanish text and tokenizer is represented in red, English text and tokenizer in blue.

- **Token-Level Language Identification (LID)**: Allows automatic language detection per token.

4.2.2. Advantages

- **Explicit Language Representation**: Enhances performance in code-switching scenarios.
- **Real-Time LID**: Enables efficient, on-the-fly language identification.
- **Flexibility**: Allows suppression or addition of languages during inference.
- **Comparable ASR Accuracy**: Performs on par with aggregated tokenizers while incorporating LID.
- **Multilingual Expansion**: Facilitates the addition of new languages using weight surgery, reducing training time.

4.2.3. Comparison with LID Symbols

Unlike models that use special LID tokens to mark monolingual spans, the concatenated tokenizer provides continuous LID tracking at the token level.

4.3. Synthetic Data Generation

4.3.1. Data Sources

This approach generates code-switching speech using monolingual corpora.

4.3.2. Configurable Parameters

- **Max/Min Duration**: Controls the length of speech segments.
- **Silence Durations**: Adjusts the timing of pauses before, between, and after speech segments.
- **Sampling Frequency**: Alters the probability distribution of different languages.
- **Amplitude Normalization & Scaling**: Ensures consistent peak amplitudes across samples.

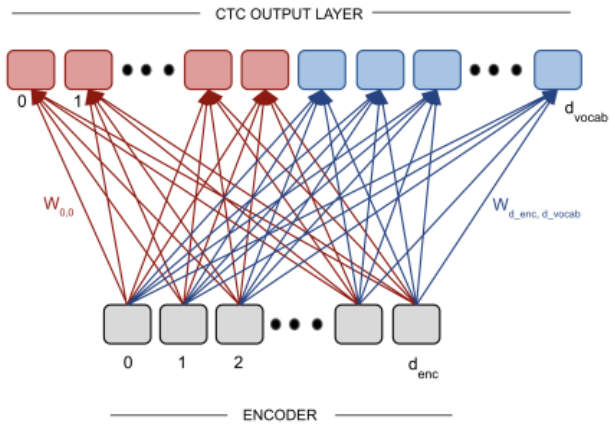

Figure 7. Diagram illustrating the benefits of concatenated tokenizers for easy addition/suppression of languages in multilingual ASR models. For simplicity, a single output step of a bilingual ASR model is shown with a CTC decoder consisting of one feed-forward fully connected layer (FC) with weights W that maps encoder representation to token logits. The concatenated tokenizer has two languages marked by red and blue.Due to the non-overlapping token mappings for different languages in the concatenated tokenizer, the FC weights can easily be separated and modified independently.

4.3.3. Data Generation Techniques

- **Offline**: Pre-generates training datasets.
- **Online**: Dynamically creates training data, enabling flexible experimentation.

4.4. Language Identification (LID) Task

4.4.1. Task Description

- Determines the language spoken in an audio sample.
- Essential for code-switching ASR, as it identifies language transitions within speech.

4.4.2. Role of Concatenated Tokenizers

- Token IDs inherently encode language identity, enabling utterance-level LID.
- The most frequently occurring language in an utterance is classified as its primary language.

4.4.3. Adaptability to Unseen Data

- Evaluated on the **FLEURS dataset** (not seen during training) [21].
- Achieved over **98% accuracy**, outperforming previous LID models.

4.5. Model Strengths

4.5.1. Enhanced Code-Switching Recognition

- The concatenated tokenizer preserves language identity, improving transcription accuracy in code-switching settings.

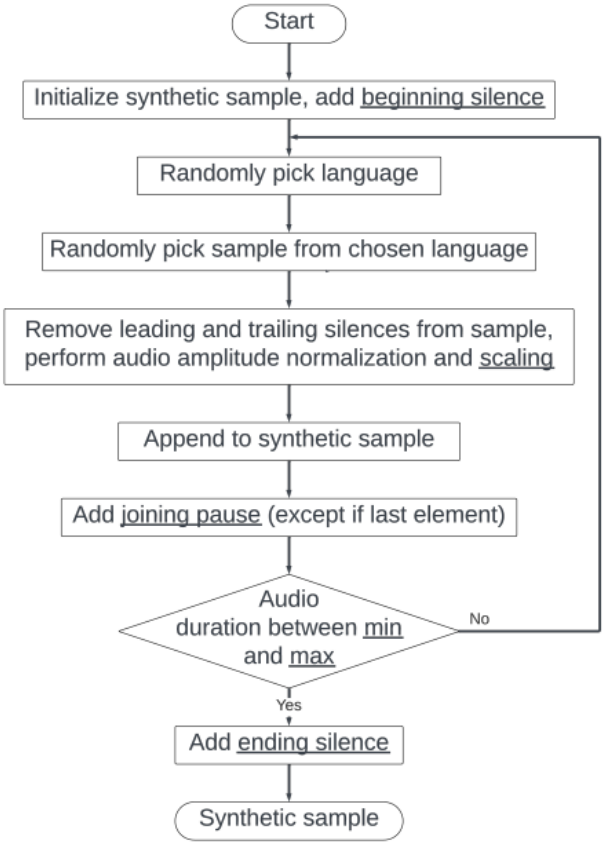

Figure 8. Flowchart of the synthetic CS sample generation process for two languages. The controllable hyperparameters have been underlined. The process can be used for both online synthetic data generation in the dataloader or offline creation of synthetic speech corpus.

- This model achieves **state-of-the-art (SotA) performance** on real-world **Miami Bangor (EN-ES) and MUCS (EN-HI) datasets** [1].

4.5.2. Scalable Synthetic Data Utilization

- Generates realistic code-switching speech from monolingual sources.
- Reduces reliance on manually annotated code-switching datasets.
- Enables dynamic dataset generation, reducing dependency on static corpora.

4.5.3. Improved Spoken Language Identification

- Token-level LID eliminates the need for a separate LID model.
- Outperforms previous models, achieving **98%+ accuracy on FLEURS** [21].

4.5.4. Adaptability to Diverse Training Strategies

A single model can support:

- **Monolingual ASR**
- **Bilingual ASR**
- **Code-Switching ASR**
- **Pretrained bilingual checkpoints** accelerate code-switching model training.

4.5.5. Balanced Performance Optimization

- Unlike earlier code-switching ASR models, this approach maintains ASR accuracy across languages.
- The concatenated tokenizer matches the performance of aggregated tokenizers while integrating LID functionality.

4.5.6. Open-Source Availability and Reproducibility

- The methods are **open-source** via the **NeMo Toolkit** [20], fostering **community collaboration and adoption**.

5. Experimental Results and Evaluation Metrics

5.1. Experimental Setup

5.1.1. Datasets

- **English Corpus:** LibriSpeech (960 hours) [3].
- **Spanish Corpus:** Common Voice 7.0 [26], Multilingual LibriSpeech, VoxPopuli, Fisher (1,300 hours) [24].
- **Hindi Corpus:** ULCA Dataset (2,250 hours), MUCS 2021 [23].
- **Synthetic CS Data:** 10,000 hours generated using the proposed synthetic data pipeline for English-Hindi and English-Spanish.

5.1.2. Models and Training

- **Base Model:** Conformer-RNNT Large (120M parameters, no external LM).
- **Training:** 200 epochs with AdamW optimizer, Noam scheduler, 0.0015 peak learning rate, and warmup steps.
- **Tokenizer Variants:** Comparison between Concatenated Tokenizer and Aggregate Tokenizer in monolingual, bilingual, and CS models.

5.2. Experimental Results

5.2.1. Monolingual and Bilingual ASR Performance

The performance of monolingual and bilingual ASR models is evaluated using Word Error Rate (WER) on English, Spanish, and Hindi test sets. As shown in Table 1, bilingual models perform comparably to monolingual ASR while offering multilingual capabilities.

The results indicate that the Aggregate Tokenizer achieves slightly lower WER than the Concatenated Tokenizer . However, the Concatenated Tokenizer retains explicit language separation at the token level, making it useful for tasks involving language identification (LID) .

5.2.2. Code-Switching ASR Performance

The code-switching ASR performance is measured on both synthetic CS speech and real-world datasets , including Miami Bangor (English-Spanish) [4] and MUCS 2021 (English-Hindi) [23] . The results in Table 2 show that CS models trained on synthetic CS speech generalize well to real-world CS speech.

Although the Concatenated Tokenizer performs similarly to the Aggregate Tokenizer , it introduces explicit token-level LID , which enhances adaptability in multilingual ASR applications.

5.2.3. Spoken Language Identification (LID) Accuracy

The performance of spoken language identification (LID) is evaluated on the FLEURS dataset [21], as shown in Table 3 . The Concatenated Tokenizer provides token-level LID with over 98% accuracy , eliminating the need for a separate LID model.

5.3. Evaluation Metrics

5.3.1. Word Error Rate (WER)

$$WER = \frac{S + D + I}{N} \tag{1}$$

where:

- S = Substitutions
- D = Deletions
- I = Insertions
- N = Total words

Lower WER indicates better ASR performance.

5.3.2. Code-Switching Evaluation Metrics

- **Code-Switching Point Detection Accuracy (CSPA):** Measures the correct identification of language transition points.
- **Code-Switching Performance Gap (CS-PG):**

$$CS - PG = \frac{WER_{CS} - WER_{mono}}{WER_{mono}} \times 100 \tag{2}$$

5.3.3. Spoken Language Identification (LID) Metrics

- **LID Accuracy:** Percentage of correctly classified tokens.
- **F1-Score & Confusion Matrix:** Used to evaluate misclassification between languages.

6. Limitations

- **Scalability & Vocabulary Growth**: The concatenated tokenizer expands its vocabulary linearly with each new language, increasing memory demands and limiting scalability beyond bilingual models.
- **Static Token Allocation**: Predefined token spaces may not be optimal for imbalanced datasets, reducing ASR accuracy for underrepresented languages.
- **Language Representation Fusion**: Basic fusion techniques fail to capture complex cross-lingual dependencies, negatively impacting code-switching accuracy.

Table 1. Monolingual evaluation set results for the English-Spanish and English-Hindi models. We present WER(%) (lower is better) for multilingual (ml) and code-switched (cs) models trained with concatenated (con) and aggregate (agg) tokenizers vs monolingual baselines. The use of the concatenated tokenizer does not hurt model performance while adding the ability to predict LID for each token.

(a) (a) English-Spanish results on the monolingual English Librispeech test-other and Spanish Fisher test sets.

Model	Tokenizer	English LS test-other	Spanish Fisher-test
en	mono	5.29	98.37
es	mono	85.68	16.14
ml	agg	5.00	16.37
ml	con	5.14	16.72
cs	agg	5.38	16.35
cs	con	5.28	16.42

(b) (b) English-Hindi results on the monolingual English Librispeech test-other and Hindi ULCA eval sets.

Model	Tokenizer	English LS test-other	Hindi ULCA
en	mono	5.29	100
hi	mono	100	10.53
ml	agg	5.00	10.78
ml	con	5.14	10.73
cs	agg	5.42	11.35
cs	con	5.29	11.64

Table 2. Performance comparison of the code-switched (cs) English-Spanish and English-Hindi models trained with concatenated (con) and aggregate (agg) tokenizers on both synthetic and real-world blind CS evaluation datasets. The performance of the multilingual (ml) models has also been reported as a benchmark. We observe that cs models significantly outperform ml models, highlighting the advantage of using the proposed synthetic CS data for training.

(a) (a) Code-switched English-Spanish models: WER(%) on synthetic and Miami-Bangor CS evaluation sets.

Model	Tokenizer	synth	Miami
cs	agg	5.51	50.0 [4]
cs	con	5.50	53.3
ml	agg	16.52	58.78
ml	con	24.08	63.54

(b) (b) Code-switched English-Hindi models: WER(%) on synthetic and MUCS CS evaluation sets.

Model	Tokenizer	synth	MUCS
cs	agg	6.55	30.3 [23]
cs	con	6.57	28.78
ml	agg	35.70	62.18
ml	con	53.01	100.0

- **Limited LID Integration**: Token-based language identification (LID) struggles to track dynamic language shifts in natural speech, affecting real-time performance.
- **Dataset Scarcity**: Current benchmarks lack spontaneous code-switching speech, hindering model generalization to real-world scenarios.

7. Open Challenges and Potential Solutions

7.1. Scalability and Vocabulary Enlargement

- **Problem:** As more languages are added, vocabulary size grows proportionally, increasing memory and computational demands. This limits scalability beyond bilingual or trilingual setups.

Table 3. Spoken Language Identification (LID) performance.

Language	# of samples	LID Accuracy (%)
English	647	98 (FLEURS) [21]
Spanish	908	100 (FLEURS)
Hindi	418	99 (FLEURS)

- **Challenge:** More languages require significantly more parameters, leading to longer training times and slower inference.
- **Potential Solution:** Implement compact multilingual tokenization techniques, such as subword sharing or adaptive token mapping, to control vocabulary expansion.

7.2. Static Token Allocation

- **Problem:** Current tokenizers allocate fixed token spaces per language without adjusting dynamically based on language complexity or training data distribution.
- **Challenge:** Poor token allocation can result in insufficient token availability for underrepresented languages, weakening recognition performance.
- **Potential Solution:** Use adaptive token space allocation, dynamically adjusting token distribution based on language frequency and complexity.

7.3. Limited Integration of Language-Specific Representations

- **Problem:** Existing fusion techniques (e.g., concatenation, summation) fail to effectively capture complex cross-lingual dependencies.
- **Challenge:** The model struggles to transfer contextual knowledge efficiently between different languages.
- **Potential Solution:** Implement advanced fusion mechanisms, such as cross-attention-based mixture-of-experts (MoE) fusion, as explored in models like CAMEL [2].

7.4. Inefficient Use of Language Bias

- **Problem:** Token ID ranges are used for language identification (LID), but this approach does not adequately handle complex language shifts.
- **Challenge:** The lack of explicit contextual language information increases errors in ambiguous code-switching scenarios.
- **Potential Solution:** Utilize language diarization decoders (LD decoders), as seen in CAMEL, to incorporate language awareness at the embedding level [2].

7.5. Data Augmentation and Code-Switching Generalization

- **Problem:** ASR models for code-switching lack diverse, large-scale datasets, leading to poor generalization.
- **Challenge:** While synthetic data augmentation helps, current methods fail to fully capture natural code-switching patterns.
- **Potential Solution:** Research into speech collage-based augmentation and low-resource unsupervised learning methods, such as those used in Speech Collage.

7.6. Evaluation Challenges in Code-Switching ASR

- **Problem:** Current ASR benchmarks do not accurately reflect real-world code-switching scenarios.
- **Challenge:** Many datasets (e.g., Miami Bangor, SEAME) emphasize read speech rather than spontaneous speech.
- **Potential Solution:** Develop more realistic evaluation datasets, such as DECM (a German-English CS dataset), to better benchmark spontaneous code-switching speech [5].

7.7. Cross-Lingual Transfer Learning and Generalization

- **Problem:** Most state-of-the-art ASR models are trained on specific language pairs, making it difficult to generalize to new languages.
- **Challenge:** Adding new languages typically requires retraining the entire model.
- **Potential Solution:** Implement cross-lingual transfer learning techniques, such as ”weight surgery,” to enable more efficient multilingual adaptation.

7.8. Handling Code-Mixing in ASR

- **Problem:** Code-mixing creates inconsistencies in phonetics, syntax, and semantics, making it difficult for ASR models to process intra-word and intra-phrase language shifts.
- **Challenge:** Standard tokenization methods fail to address pronunciation variations and grammatical inconsistencies in mixed-language utterances.
- **Potential Solution:** Explore phoneme-aware fusion models and context-aware tokenization strategies.

7.9. Robustness to Noise and Accents

- **Problem:** Code-switching frequently occurs in noisy environments and among diverse accents, negatively affecting ASR performance.
- **Challenge:** Many ASR models are trained on clean, scripted datasets, making them less resilient to real-world speech variations.
- **Potential Solution:** Implement noise-augmented training, accent-aware speech models, and self-supervised learning on noisy code-switching data.

8. Future Research Directions

To address these limitations, future research should focus on:

- **Compact Multilingual Tokenization**: Developing efficient tokenization strategies to prevent vocabulary size explosion in multilingual models [1].
- **Enhanced Cross-Lingual Fusion**: Implementing techniques such as gated cross-attention mechanisms to improve contextual understanding in code-switching ASR [2].

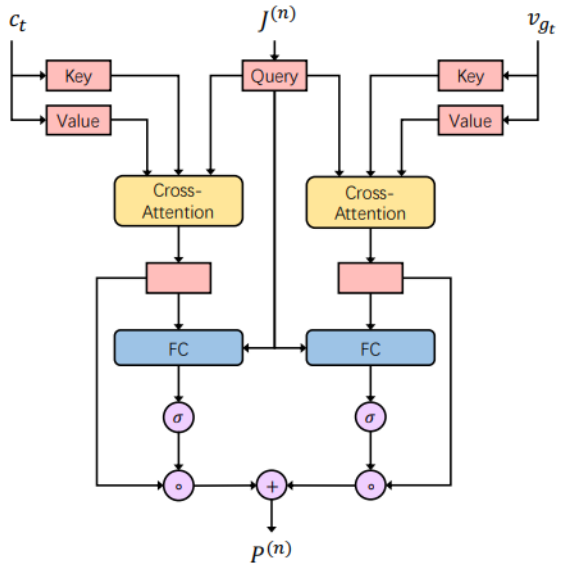

Figure 9. An illustration of gated cross-attention in the decoder.

- **Self-Supervised Learning for Low-Resource ASR**: Leveraging self-supervised and semi-supervised learning to enhance recognition performance in low-resource code-switching scenarios [8, 9].
- **Comprehensive Evaluation Datasets**: Expanding benchmark datasets to include spontaneous code-switching speech, such as the DECM dataset [5].
- **Language-Aware Decoding**: Exploring advanced language diarization decoders (LD decoders) to improve real-time language tracking in ASR systems [6, 7].

References

[1] K. Dhawan, D. Rekesh, and B. Ginsburg, "Unified Model for Code-Switching Speech Recognition and Language Identification Based on Concatenated Tokenizer," *Proc. 6th Workshop on Computational Approaches to Linguistic Code-Switching*, 2023. 4, 5, 9

[2] H. Wang, X. Wan, N. Zheng, K. Liu, H. Zhou, G. Li, and L. Xie, "CAMEL: Cross-Attention Enhanced Mixture-of-Experts and Language Bias for Code-Switching Speech Recognition," *arXiv preprint arXiv:2412.12760*, 2025. 8, 9

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," *Proc. ICASSP*, 2015. 6

[4] M. Deuchar, P. Davies, J. R. Herring, M. C. Parafita Couto, and D. Carter, "Building bilingual corpora: The Miami Bangor Corpus," *Advances in the Study of Bilingualism*, 2014. 6, 7

[5] E. Y. Ugan, N. Pham, and A. Waibel, "DECM: Evaluating Bilingual ASR Performance on a Code-switching/Mixing Benchmark," *Proc. LREC-COLING*, 2024. 8, 9

[6] F. Weng, H. Bratt, L. Neumeyer, and A. Stolcke, "A study of multilingual speech recognition," *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, 1997. 3, 9

[7] A. Waibel, H. Soltau, T. Schultz, T. Schaaf, and F. Metze, "Multilingual speech recognition," *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, 2000. 3, 9

[8] A. Ali, S. A. Chowdhury, A. Hussein, and Y. Hifny, "Arabic code-switching speech recognition using monolingual data," *Proc. Interspeech*, 2021. 3, 9

[9] 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. 3, 9

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labeling unsegmented sequence data with recurrent neural networks," *Proc. ICML*, 2006. 3

[11] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2013. 3

[12] A. Radford, J. Wook Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022. 3

[13] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, et al., "Scaling speech technology to 1,000+ languages," *arXiv preprint arXiv:2305.13516*, 2023. 3

[14] H. Seki, T. Hori, S. Watanabe, J. R. Le Roux, and J. R. Hershey, "End-to-end multilingual multi-speaker speech recognition," *Proc. Interspeech*, 2019. 3

[15] G. I. Winata, A. Madotto, C.-S. Wu, and P. Fung, "Code-switched language models using neural-based synthetic data from parallel sentences," *arXiv preprint arXiv:1909.08582*, 2019. 3

[16] D. Gupta, A. Ekbal, and P. Bhattacharyya, "A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning," *Findings of the Association for Computational Linguistics (EMNLP)*, 2020. 3

[17] I. Tarunesh, S. Kumar, and P. Jyothi, "From machine translation to code-switching: Generat-

ing high-quality code-switched text," *arXiv preprint arXiv:2107.06483*, 2021. 3

[18] 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," *Proc. ICASSP*, 2018. 3

[19] O. Weller, M. Sperber, T. Pires, H. Setiawan, C. Gollan, D. Telaar, and M. Paulik, "End-to-end speech translation for code-switched speech," *Proc. ACL*, 2022. 3

[20] NVIDIA NeMo Toolkit, "NeMo: Speech and Language Processing Toolkit," *https://github.com/NVIDIA/NeMo*, 2023. 6

[21] A. Conneau et al., "FLEURS: Few-shot learning evaluation of universal representations of speech," *Proc. SLT*, 2023. 5, 6, 8

[22] ULCA, "ULCA Speech Dataset: A multilingual corpus for ASR research," *Govt. of India, Bhashini Initiative*, 2022. Available: https://ulca.in

[23] K. Diwan, R. Kumar, A. Singh, et al., "Multilingual and code-switching ASR challenges for low-resource Indian languages: The MUCS 2021 benchmark," *Proc. Interspeech*, 2021. 6, 7

[24] N. Wang, L. Wu, J. Zhang, et al., "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning, and interpretation," *Proc. NeurIPS*, 2021. 6

[25] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: A resource for the next generations of speech-to-text," *Proc. LREC*, 2004.

[26] A. Ardila, M. Branson, K. Davis, et al., "Common Voice: A massively-multilingual speech corpus," *Proc. LREC*, 2020. 6