MaskLID: Code-Switching Language Identification through Iterative Masking

Amir Hossein Kargaran, François Yvon, Hinrich Schütze

MaskLID: Code-Switching Language Identification through Iterative Masking

Amir Hossein Kargaran[♣], **François Yvon**[♠] **and Hinrich Schütze**[♣]
[♣]LMU Munich & Munich Center for Machine Learning, Munich, Germany
[♠]Sorbonne Université & CNRS, ISIR, Paris, France
amir@cis.lmu.de

Abstract

We present MaskLID, a simple, yet effective, code-switching (CS) language identification (LID) method. MaskLID does not require any training and is designed to complement current high-performance sentence-level LIDs. Sentence-level LIDs are classifiers trained on monolingual texts to provide single labels, typically using a softmax layer to turn scores into probabilities. However, in cases where a sentence is composed in both L1 and L2 languages, the LID classifier often only returns the dominant label L1. To address this limitation, MaskLID employs a strategy to mask text features associated with L1, allowing the LID to classify the text as L2 in the next round. This method uses the LID itself to identify the features that require masking and does not rely on any external resource. In this work, we explore the use of MaskLID for two open-source LIDs (GlotLID and OpenLID), that are both based on the FastText architecture. Code and demo are available at github.com/cisnlp/MaskLID.

1 Introduction

Code-switching (CS), the juxtaposition of two or more languages within a single discourse (Gumperz, 1982), is prevalent in both written and spoken communication (Sitaram et al., 2019; Doğruöz et al., 2021). While CS has traditionally been explored as a speech phenomenon (Milroy and Muysken, 1995; Auer, 2013), the increasing prevalence of CS in digital communication, such as SMS and social media platforms (Das and Gambäck, 2013; Bali et al., 2014), requires the development of techniques to also analyze CS in written texts. There is however a lack of CS data for researchers, making it difficult to study CS and to effectively train CS-aware models. This shortage affects many NLP applications dealing with CS scenarios (Solorio et al., 2021; Winata et al., 2023). A first step towards the collection of high-quality corpora of CS texts is thus to identify samples of CS in running texts.

Previous works on CS language identification (LID) have mainly focused on building *word-level* LIDs for code-switching between specific pairs of languages, and are often limited to recognize only two languages (Solorio et al., 2014; Nguyen and Doğruöz, 2013; Elfardy et al., 2013; Barman et al., 2014). However, such approaches are not realistic on a larger scale, especially considering that texts on the web typically lack prior information about the languages that are actually being used.

More recently, Burchell et al. (2024) have investigated the use of high-quality LID at the *sentence-level* to detect instances of CS. They propose to reformulate CS LID as a *sentence-level* task and to associate each segment with *a set of language labels*. Their investigation reveals the difficulty of achieving effective CS LID with existing LID models. Furthermore, their findings indicate that such LIDs predominantly predict only one of the languages occurring in CS sentences.

In this work, we continue this line of research and introduce MaskLID, a method that also uses high-quality sentence-level LID to identify CS segments. By masking the presence of the text features associated with the dominant language, MaskLID improves the ability to recognize additional language(s) as well. We explain in detail how MaskLID works in cooperation with two existing LIDs that are based on the FastText (Bojanowski et al., 2017) architecture in Section 3. As we discuss, our method can identify arbitrary pairs of languages, and is also able to detect mixtures of more than two languages in the same segment. Being based on FastText, it is also extremely fast. This two properties make MaskLID well suited to mine large web corpora for examples of real-world CS segments, that can then serve as valuable training data for applications designed to handle CS inputs. We evaluate MaskLID on two test datasets contain-

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 459–469

ing both CS and monolingual data, showing the benefits of using MaskLID (see Section 4).

2 One Sentence, Multiple Languages

2.1 Code-switching, Code-mixing

Code-switching (CS) can be defined as the alternate use of two languages within the same utterance and can happen either between sentences (inter-sentential CS) or within a sentence (intra-sentential CS or *code-mixing*) (Gumperz, 1982). While loanwords are often seen as a simple form of CS, their assimilation into a foreign linguistic system sometimes yields a mixed use of languages *within a single word*. For the purpose of this work, we mostly focus on inter-sentential CS and use the terms code-switching and code-mixing interchangeably, even though our approach could in fact apply to longer chunks of texts. From an abstract perspective, the main trait of CS is thus the juxtaposition of two (or more) languages within a single segments, a view that is also adopted in e.g. from Bali et al. (2014). From this perspective, CS ID can be formulated as identifying more than one language ID in a given text segment. We also use the fact that mixing does not take place randomly (Myers-Scotton, 1997), and that one language plays a dominant role and provides the linguistic structure into which inserts from other languages can take place.

In the next paragraph, we discuss two previous approaches that share this view and which serve as the foundation of MaskLID. For other related works, refer to Appendix A.

2.2 Detecting CS with Lexical Anchors

Our work is most closely related to the research of Mendels et al. (2018). They propose a method to identify CS data in sentences written in two languages L1 and L2. Their approach first requires a language identifier that is able to label the majority language of a document as language L1, even when the document also contains words that belong to L2. This aligns with our setup, as sentence-level LID models trained on monolingual texts often demonstrate similar performance on CS data, primarily predicting the dominant language L1 (Burchell et al., 2024).

Mendels et al. (2018) also introduce the concept of *anchors* for each language, defining an anchor as a word belonging to only one language within a language pool $\mathbb{L}$. The set of anchors in their work is computed based on the analysis of monolingual corpora, and constitutes an external resource to their CS LID system. To relax the definition of anchors, they also introduce the notion of *weak anchor* for a language L2 relative to some other language L1: an anchor is considered a weak anchor' if it is observed in monolingual L2 corpora but not in monolingual L1 corpora.

In their definition of CS for L1+L2 sentences, a sentence is then considered CS if and only if it is predicted to be in language L1 by the LID model and contains at least one weak anchor from the L2 anchor set (relative to L1). Our method shares similarity with this work in that, for L1+L2 sentences, the initial step consists in the identification of L1. However, while their approach requires the identification of sets of weak anchors for each language pair, we identify the minority language(s) L2 using only features that are internal to the main LID model, dispensing from the need to compile external resources.

2.3 CS Detection as Set Prediction Problem

Another work that is closely related to ours is the research conducted by Burchell et al. (2024). They use three different sentence-level LID models for CS LID: 1) OpenLID (Burchell et al., 2023), a high-quality LID model operating at the sentence level; 2) Multi-label OpenLID, which is similar to OpenLID but is trained with a binary cross-entropy loss instead of the conventional cross-entropy, and delivers Yes-No decisions for each possible language;[1] and 3) Franc (Wormer, 2014), which uses trigram distributions in the input text and a language model to compute languages and their scores.

However, the result of these models on CS LID are not very promising especially for the Turkish-English CS dataset (see Section 4). One reason is that the occurrence of one single English word in a Turkish sentence is tagged in the gold reference as an instance of CS. Yet, one single word may not be enough to yield large logit values for the English label in these difficult predictions. But this is not the only reason these models fail. Scaling the baseline LID to support more languages, which is a strong motivation behind models such as GlotLID (Kargaran et al., 2023) and OpenLID, makes CS LID predictions more challenging. For instance, when the model encounters a Turkish-English sentence and predicts Turkish as the top

[1] See FastText documentation: fasttext.cc/docs/en/supervised-tutorial.html#multi-label-classification.

language, the second best prediction may not be English, but a language closest to Turkish instead, such as North Azerbaijani or Turkmen, which have more active ngram features in the CS sentence than English. Consider, for instance, the example sentence from Burchell et al. (2024, Table 9):

```
bir kahve dükkanında geçen film
tadında güzel bir şarkıya ayrılsın
gece falling in love at a coffee shop
```

OpenLID's top 5 predictions for this sentence are: 1) Turkish, 2) North Azerbaijani, 3) Crimean Tatar, 4) Turkmen, 5) Tosk Albanian, with English predicted as the 15th most likely language. Yet, for a speaker of either Turkish or English, it is obvious that this sentence is a mixture of just these two languages. To solve this, MaskLID suggests to mask the Turkish part of the sentence:

```
<MASK> film <MASK>
falling in love at a coffee shop.
```

If we now ask OpenLID to predict this masked sentence (without the token <MASK>), the top prediction would be English with 0.9999 confidence. MaskLID makes models such as OpenLID much more suitable for this task. Details on how MaskLID computes the masked parts are in Section 3.

3 MaskLID

3.1 FastText-based LIDs

In this paper, we explore the use of MaskLID for LIDs based on the FastText (Bojanowski et al., 2017) architecture. However, it is also possible to apply MaskLID to other LIDs, as long as they enable to determine how much each feature (e.g., word) contributes to each supported language. FastText is one of the most popular LID architectures due to its open-source nature, high performance, ease of use, and efficiency. FastText classifier is a multinomial logistic classifier that represents the input sentence as a set of feature embeddings, making it easy to assess each feature's contribution to the final prediction.

Given a sentence s, let $f_1, f_2, \ldots, f_T$ represent the features extracted from s. Note that these features are linearly ordered, i.e., f_i precedes f_{i+1} in s. FastText maps these features onto vectors in $\mathbb{R}^d$ via feature embeddings $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. The dimensionality of these embeddings, denoted d, is a hyperparameter. A base LID using FastText architecture computes the posterior probability for a language $c \in [1 : N]$ by applying the softmax function over logits as:

$$P(c|s) = \frac{\exp(\mathbf{b}_c \cdot \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t)}{\sum_{c'=1}^{N}\exp(\mathbf{b}_{c'} \cdot \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t)}. \quad (1)$$

$P(c|s)$ is the base LID probability of the input text s belonging to language c, $\mathbf{b}_c$ is the weight vector for language c, and N is the total number of classes supported by the base LID.

To evaluate how much each feature contributes to each supported language, we need to compute logits separately for each feature. For simplicity and alignment with the FastText tokenizer (which considers white-spaces as token boundaries), we set the level of granularity of features to be the word level. The word-level feature embedding is obtained as the summation of all feature embeddings that build each word. Noting W the number of words in a sentence s, we define the $N \times W$ matrix $\mathbf{V}(s)$, where each element $\mathbf{V}_{c,t}(s)$ represents the logits for language c and word-level feature $\mathbf{x}_t$:

$$\mathbf{V}_{c,t}(s) = \mathbf{b}_c \cdot \mathbf{x}_t. \quad (2)$$

3.2 The MaskLID Method

We define the MaskLID algorithm in alignment with Burchell et al. (2024): given an input text, the objective is to return a set of codes corresponding to the language(s) it contains. However, MaskLID is more explainable and provides insights into which parts of the sentence contributed to its decision. The MaskLID algorithm works as follows:

Input:

1) sentence s.
2) α, an integer parameter used to define *strong associations* between words and languages: having a language appear in the top-α logit values for a word is a strong cue that this word belongs to that language.
3) β, an integer parameter used to define *weak associations* between words and languages: languages appearing in the top-β logit values for a word are weakly associated with that word. β is always greater than α.
4) τ, a threshold representing the minimum size of a sentence (in bytes) for which the LID makes reliable decisions.
5) λ, a parameter defining the number of times the algorithm should be repeated.

Output:

1) List of predicted languages, along with their associated word-level features.

Procedure:

0) Take sentence s and compute $\mathbf{V}(s)$ using Eq. (2). Assign s to variable u.
1) Compute the posterior probability for each possible language using Eq. (1). Find the most likely class ($L1 = \arg\max_c P(c|u)$) along with its corresponding probability $P(L1|u)$. Assign L1 to variable L_u.
2) Process column $\mathbf{V}_{:,t}(s)$ for each unmasked word t in u. If the value of $\mathbf{V}_{\mathrm{L}_u,t}(s)$ is in the top-β values for that column, then assign word t to language L_u. If the value of $\mathbf{V}_{\mathrm{L}_u,t}$ is among the top-α values for that column, mask word t from sentence u.
   Masked words play here a role similar to the anchors used in (Mendels et al., 2018): recall that for these authors, anchor words are selected to uniquely identify one language – there removal is likely to decrease the recognition of L1, without impacting the ability to recognize L2. In our approach, we identify these *pseudo-anchors* on the spot, relying on the LID internal scoring procedure.
3) check if length of u (in bytes, ignoring masked words) is greater than τ. If not, then terminate. This is one termination condition (for additional considerations, refer to Appendix B). Setting $\tau = 0$ will just check that the masked sentence is not empty, but it is better to use a non-zero threshold, as most sentence-level LIDs do not reliably predict short sentences (Jauhiainen et al., 2019).
4) if the number of iterations is lower than λ then go to back to step 1, else stop.

The complexity of this greedy procedure is $O(\lambda \times T \times N \log \beta)$.

4 Experiments and Results

Here, we provide an overview of our baselines and test data. We assess the performance of the baselines by testing them both with and without MaskLID. Our setting of hyperparameters is explained in Appendix C.2.

4.1 Baselines

Our baseline LID models are OpenLID[2] (supporting ≈200 languages) and GlotLID v3.0[3] (supporting ≈2100 languages), two LIDs based on the FastText architecture. For a fair comparison between these models, we limit the languages that GlotLID supports to the same set as OpenLID (see details in Appendix C.1). Two exceptions are romanized Nepali (nep_Latn) and Hindi (hin_Latn), which are not supported by OpenLID, but for which we also have test data that is also used to evaluate MaskLID with GlotLID.

4.2 Test Data

We choose Turkish-English (Yirmibeşoğlu and Eryiğit, 2018), Hindi-English (Aguilar et al., 2020), Nepali-English (Aguilar et al., 2020) and Basque-Spanish (Aguirre et al., 2022), as our test datasets. We have data for four CS labels and six single labels (see Table 1). Details regarding these test sets, preprocessing, their descriptions, and information on access are in Appendix D.

4.3 Metrics

We use the number of exact (#EM) and partial matches (#PM), along with the count of false positives (#FP) as the main metrics in our evaluation. To ensure clarity and prevent misinterpretation of the results, we report the absolute number of instances rather than percentages.

1) #EM: This metric counts a prediction as a match when it exactly matches the true
2) #PM: This metric counts a prediction as a match when only part of the information is correct: for a single label, if it is part of the prediction; for a CS label, if part of the label exactly matches the prediction.
3) #FP: If any label other than X is misclassified as X, it counts as an FP for X. We do not consider the #FP for single labels, as partial matches of CS are counted as FP for single labels. Therefore, we only report the FP for CS sentences.

4.4 Results

Table 1 presents the results on the test data for two baseline LIDs and two settings, with and without MaskLID. The best exact match (#EM) for CS labels is in boldface in the table, demonstrating that

[2] https://huggingface.co/laurievb/openlid
[3] https://huggingface.co/cis-lmu/glotlid

		Baseline + MaskLID				Baseline			
		#EM/#PM ↑		#FP ↓		#EM/#PM ↑		# FP ↓	
	#S	GlotLID	OpenLID	GlotLID	OpenLID	GlotLID	OpenLID	GlotLID	OpenLID
CS Turkish–English	333	**91**/328	<u>68</u>/327	0	0	4/327	4/326	0	0
CS Basque–Spanish	440	<u>43</u>/430	**47**/426	0	0	9/426	9/424	0	3 (from Spanish)
CS Hindi–English	253	**29**/219	-	0	-	<u>5</u>/211	-	0	-
CS Nepali–English	712	**22**/444	-	0	-	<u>0</u>/420	-	0	-
Single Basque	357	354/354	355/355	-	-	353/353	355/355	-	-
Single Spanish	347	335/337	297/300	-	-	337/340	287/311	-	-
Single Turkish	340	333/337	329/334	-	-	335/337	329/335	-	-
Single Hindi	29	18/19	-	-	-	17/18	-	-	-
Single Nepali	197	63/75	-	-	-	68/72	-	-	-
Single English	508	459/490	428/469	-	-	486/490	455/462	-	-

Table 1: Number of exact (#EM) and partial matches (#PM) and count of false positives (#FP) calculated over CS and single label test instances. The best exact match for CS instances is in bold, and the second is underlined. #S reports the number of sentences for each test set.

the baseline with MaskLID achieves better performance compared to the baseline without it. Partial matches (#PM) in both settings (with and without MaskLID) are quite similar.

For CS Turkish-English, MaskLID detects 91 CS at best, compared to 4 without it. For Basque-Spanish, MaskLID detects 47 CS, versus 9 without it. For Hindi-English, MaskLID detects 29 CS, compared to 5 without it. For Nepali-English, MaskLID detects 22 CS, while none are detected without it.

In all single-language test instances, GlotLID outperforms OpenLID. This is also the case for CS language instances, except for Basque-Spanish. Considering the relatively poorer performance of OpenLID in both single Basque and single Spanish, overall, GlotLID proves to be the better model for these tasks.

Additional Considerations. For CS instances: 1) The difference between #PM and #EM corresponds to the number of times only one of two mixed languages in a CS instance is predicted. 2) The difference between number of sentences (#S) and #PM corresponds to the number of times none of the languages in the CS instance is predicted. In all CS setups, the #EM and #PM value in the baseline with MaskLID are always greater than without. Additionally, the difference between #PM and #EM is also smaller, which indicates a higher precision in CS LID.

For single language instances: 1) The difference between #PM and #EM corresponds to the number of times the single label instance is classified as part of a multi-label instance. 2) The difference between #S and #PM corresponds to the number of times a single label is never predicted, even as part of a multi-label instance. For all single language instances, the results are quite similar except for single English, where the number of incorrect CS in baseline with MaskLID (#PM - #EM) is greater than with baseline alone. To address this, using a larger minimum length τ helps decrease the number of CS false positives. For single English, in GlotLID with MaskLID setting, increasing τ from 20 to 25 raises the #EM from 459 to 473; however, it reduces the #EM in GlotLID with MaskLID setting for CS Turkish-English from 91 to 67, CS Hindi-English from 29 to 26, and CS Nepali-English from 22 to 18. Examples of successes and failures of MaskLID are provided in Appendix E.

5 Conclusion

We present MaskLID, a simple, yet effective, method for scalable code-switching (CS) language identification (LID). MaskLID is designed to complement existing high-performance sentence-level LID models and does not require any training. In our experiments, MaskLID increases CS LID by a factor of 22 in Turkish-English, by 22 in Nepali-English, by 6 in Hindi-English and by 5 in Basque–Spanish.

In future work, we aim to explore the use of subword-level, instead of word-level features, extending the applicability of the method to languages that do not use spaces for word separation. Additionally, we plan to generalize this method to other LID models using techniques like LIME (Ribeiro et al., 2016) to map features to languages. Last, we intend to apply MaskLID on the web data, in the hope that it will help build larger high-quality web corpora for CS.

Limitations

The CS testsets we use in this study only represent a small subset of potential uses of CS languages. Creating additional CS datasets for more languages would definitely be an extension of this work. MaskLID uses hyperparameters, and changing the model and the set of languages it supports may require adjustments to these parameters. Although MaskLID detects more CS than the standalone baseline LID models, it still has a long way to go to predict the majority of them. One important source of remaining errors is loan words, where the L2 insert is just one word long: these cannot be detected with out current hyperparameter settings. The performance of MaskLID is also bound by the LID it uses; it might not have good performance for some languages, resulting e.g. in a large number of false positives.

Ethics Statement

MaskLID uses openly available open-source LID models and does not require any additional resources except for hyperparameters. Concerning the evaluation data, these datasets have undergone anonymization to safeguard the privacy of all parties involved. We provide links to the data and do not host it ourselves. We provide detailed descriptions of our method and evaluation process. Additionally, we make our code openly available to foster collaboration and reproducibility.

Acknowledgements

The authors thank the anonymous reviewers and editors for their comments of the previous version of this work. This research was supported by DFG (grant SCHU 2246/14-1).

References

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Inciarte. 2022. AfroLID: A neural language identification tool for African languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1958–1981, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Maia Aguirre, Laura García-Sardiña, Manex Serras, Ariane Méndez, and Jacobo López. 2022. BaSCo: An annotated Basque-Spanish code-switching corpus for natural language understanding. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3158–3163, Marseille, France. European Language Resources Association.

Mohamed Al-Badrashiny and Mona Diab. 2016. LILI: A simple language independent approach for language identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1211–1219, Osaka, Japan. The COLING 2016 Organizing Committee.

Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity*. Routledge.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "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*, pages 116–126, Doha, Qatar. Association for Computational Linguistics.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23, Doha, Qatar. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ralf D Brown. 2012. Finding and identifying text in 900+ languages. *Digital Investigation*, 9:S34–S43.

Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 865–879, Toronto, Canada. Association for Computational Linguistics.

Laurie Burchell, Alexandra Birch, Robert Thompson, and Kenneth Heafield. 2024. Code-switched language identification is harder than you think. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 646–658, St. Julian's, Malta. Association for Computational Linguistics.

Amitava Das and Björn Gambäck. 2013. Code-mixing in social media text. *Traitement Automatique des Langues*, 54(3):41–64.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian

social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Jonathan Dunn. 2020. Mapping languages: The corpus of global language use. *Language Resources and Evaluation*, 54:999–1018.

Jonathan Dunn and Lane Edwards-Brown. 2024. Geographically-informed language identification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7672–7682, Torino, Italia. ELRA and ICCL.

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2013. Code switch point detection in Arabic. In *Natural Language Processing and Information Systems: 18th International Conference on Applications of Natural Language to Information Systems, NLDB 2013, Salford, UK, June 19-21, 2013. Proceedings 18*, pages 412–416. Springer.

John J Gumperz. 1982. *Discourse strategies*. 1. Cambridge University Press.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2022. HeLI-OTS, off-the-shelf language identifier for text. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3912–3922, Marseille, France. European Language Resources Association.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2020. A survey of current datasets for code-switching research. In *2020 6th international conference on advanced computing and communication systems (ICACCS)*, pages 136–141. IEEE.

Amir Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. GlotLID: Language identification for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218, Singapore. Association for Computational Linguistics.

Amir Hossein Kargaran, François Yvon, and Hinrich Schütze. 2024. GlotScript: A resource and tool for low resource writing system identification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7774–7784, Torino, Italia. ELRA and ICCL.

Laurent Kevers. 2022. CoSwID, a code switching identification method suitable for under-resourced languages. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 112–121, Marseille, France. European Language Resources Association.

Ben King and Steven Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119, Atlanta, Georgia. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2017. LanideNN: Multilingual language identification on character window. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 927–936, Valencia, Spain. Association for Computational Linguistics.

Thomas Lavergne, Gilles Adda, Martine Adda-Decker, and Lori Lamel. 2014. Automatic language identity tagging on word and sentence-level in multilingual text sources: a case-study on Luxembourgish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3300–3304, Reykjavik, Iceland. European Language Resources Association (ELRA).

Holy Lovenia, Samuel Cahyawijaya, Genta Winata, Peng Xu, Yan Xu, Zihan Liu, Rita Frieske, Tiezheng Yu, Wenliang Dai, Elham J. Barezi, Qifeng Chen, Xiaojuan Ma, Bertram Shi, and Pascale Fung. 2022. ASCEND: A spontaneous Chinese-English dataset for code-switching in multi-turn conversation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7259–7268, Marseille, France. European Language Resources Association.

Manuel Mager, Özlem Çetinoğlu, and Katharina Kann. 2019. Subword-level language identification for intra-word code-switching. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2005–2011, Minneapolis, Minnesota. Association for Computational Linguistics.

Gideon Mendels, Victor Soto, Aaron Jaech, and Julia Hirschberg. 2018. Collecting code-switched data from social media. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lesley Milroy and Pieter Muysken. 1995. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*, volume 10. Cambridge University Press.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Dong Nguyen and A. Seza Doğruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 857–862, Seattle, Washington, USA. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on Twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Thamar Solorio, Shuguang Chen, Alan W. Black, Mona Diab, Sunayana Sitaram, Victor Soto, Emre Yilmaz, and Anirudh Srinivasan, editors. 2021. *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Online.

Aleksander Stensby, B John Oommen, and Ole-Christoffer Granmo. 2010. Language detection and tracking in multilingual documents using weak estimators. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 600–609. Springer.

Genta Winata, Sudipta Kar, Marina Zhukova, Thamar Solorio, Mona Diab, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali, editors. 2023. *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Singapore.

Titus Wormer. 2014. Franc library.

Zeynep Yirmibeşoğlu and Gülşen Eryiğit. 2018. Detecting code-switching between Turkish-English language pair. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 110–115, Brussels, Belgium. Association for Computational Linguistics.

A Related Work

LID has been a longstanding and active research area in NLP (Jauhiainen et al., 2019). Past research in LID can be classified into two primary subcategories: 1) monolingual LID; 2) CS LID.

The first category is designed under the assumption that the text is entirely monolingual, or the text contains discrete monolingual chunks (e.g., sentences) in different languages. The aim of these works is to identify the language of the whole text or each chunk. The majority of research on this topic has been focused on covering more languages, with recent work claiming to cover over a thousand (Kargaran et al., 2023; Adebara et al., 2022; NLLB Team et al., 2022; Burchell et al., 2023; Dunn, 2020; Dunn and Edwards-Brown, 2024; Jauhiainen et al., 2022; Brown, 2012).

The second category has received less attention than the first category. LID at either the document or sentence level is not effective in accurately identifying CS, which may occur within a sentence.

LIDs that identify languages at the word level are proposed to address this issue. The majority of studies have focused on scenarios where two predefined languages are looked for in the input, specifically concentrating on binary language detection at the word level (Nguyen and Doğruöz, 2013; Das and Gambäck, 2014; Elfardy et al., 2013; King and Abney, 2013; Al-Badrashiny and Diab, 2016). While some attempts choose sentence-level granularity (Stensby et al., 2010; Lavergne et al., 2014), most CS LIDs prefer operating at the word or token level. Nevertheless, certain approaches broaden the analysis to the character level (Kocmi and Bojar, 2017). Among the most recent works on CS LID, Kevers (2022) propose a method to locate CS, primarily in multilingual documents when language diversity is unstructured. It uses a sliding window and determines the local language of each token. This method requires linguistic resources such as word lists and monolingual corpora. Rijhwani et al. (2017) acknowledge the challenges in building word-level LID for CS LID. They propose an unsupervised word-level LID approach and apply it to estimate language pairs code-switched on Twitter. Their findings indicate that approximately 3.5% of tweets were code-switched. Mager et al. (2019) extend the LID task from the word level to the subword level, involving the splitting of mixed words and tagging each part with an LID. However, training such LID models at the subword level requires CS training data, which is not practical on a larger scale.

B Confidence in MaskLID

We discuss here additional considerations regarding the design MaskLID, notably aimed the keeping a good balance between over and under detection of labels, which is a key aspect to reliably detect instances of CS.

A first comment is that in our approach, the value of parameter α is kept constant. An extension would vary this value during iterations, depending on the desired level of CS-sensitive results. However, selecting a smaller α increases the likelihood of a language being chosen again in the next round(s). In such cases, the α value for the next round should be increased so that more words belonging to L1 are masked.

To ensure that MaskLID yields a low false positive rate (FPR), the feature set assigned to language L_u in step 2 should have a minimum length (in byte) τ. If not, we should increase the β value and repeat the process again to obtain a larger feature set, and evaluate whether the confidence probability prediction for this set is high. If not, terminate the procedure. It is important to note that β does not play a role in masking, as only α affects this process. The reason for defining both α and β instead of relying solely on α is to ensure a minimum byte size so that the probability prediction for this feature set can be trusted and to guarantee its high confidence. Typical α values should thus be lower than β and only target the features that strongly cue language and should accordingly be masked.

Maintaining high confidence in steps 1 and 4 is more tricky; the reason for the low confidence probability in these steps could be the presence of another language. However, it could also be because the text is not among the languages supported by the LID (Kargaran et al., 2023). We suggest using a low confidence threshold for these steps or not using one at all.

Finally, our algorithm uses two termination conditions, one based on the minimum sentence length (τ) , one based on the maximum number of languages in a given sentence (λ): 2 or 3 is recommended. In our test dataset, we know in advance that the number of languages is at most 2.

C Experimental Settings

C.1 The Label Sets of LIDs

Following the labeling proposed by NLLB Team et al. (2022), our two baseline LIDs use language-scripts as labels. They define a language-script as a combination of a ISO 639-3 language code and a ISO 15924 script code.

We constrain GlotLID to the set of languages supported by OpenLID. Most of the labels supported by OpenLID are supported by GlotLID. The total number of labels is 201 for OpenLID, and we select 200 labels for the constrained version of GlotLID. The only difference is due to the fact that OpenLID uses two labels for the Chinese language (zho), written in Hans and Hant scripts, whereas GlotLID combines both under the label Hani. Also, GlotLID does not support acq_Arab, nor does it not support labels pes_Arab and prs_Arab individually (as OpenLID does) but as the merged macrolanguage fas_Arab. To compensate for the lack of these two labels and to also perform experiments for Hindi and Nepali in romanized script, we add hin_Latn and npi_Latn to the set of labels for con-

strained GlotLID.

To restrict a FastText-based LID model to a specific subset of languages, as indicated by Eq. (1), we only need to consider the $\mathbf{b}_c$ values for languages c that are members of the chosen set of languages. This implies that languages not included in this set will be excluded from the softmax computation. Additionally, the rows belonging to these languages are also deleted from the matrix $\mathbf{V}(s)$ (Eq. (2)).

C.2 Hyperparameters

We here explain the hyperparameters specific to each method.

MaskLID. We generated 12 small synthetic code-switch corpora by combining sentence parts from French, English, Arabic, and Persian languages, ensuring a presence of at least 30% from each of the two languages participating in the final sentence. Subsequently, we applied MaskLID with different hyperparameters to achieve the best results. The hyperparameters derived from this method, which we used for the experiments in this paper, are as follows: $\alpha = 3$, $\beta = 15$, $\lambda = 2$, and $\tau = 20$. Additionally, we employed a high-confidence threshold of 0.9 for OpenLID and GlotLID to evaluate the probability predictions for the feature set in step 2 of the algorithm, as further detailed in Section B.

Baseline. Following Burchell et al. (2024), we use a threshold of 0.3 to select languages (i.e., among all languages supported by the model, the languages with confidence probability greater than 0.3 are selected). However, for a fairer comparison (since $\lambda = 2$), we only consider the top two that pass this threshold.

D Data Selection

The CS test sets available for consideration cover a small potential language set (Jose et al., 2020; Aguilar et al., 2020). Accessing suitable CS test sets for evaluating our method poses several challenges:

1) Arabic dialects, such as Standard Arabic-Egyptian Arabic, are represented in some CS datasets (Elfardy et al., 2013; Aguilar et al., 2020). However, none of the baseline LID models yield impressive results for Arabic dialects. For instance, according to Burchell et al. (2024, Table 3), OpenLID exhibits the worst FPR for Standard Arabic and Egyptian Arabic among all the languages it supports.

2) Certain datasets present unrealistic scenarios for testing our method. For example, Mandarin-English datasets with Mandarin written in Hani script and English in Latin script (Lovenia et al., 2022). Methods employing script detection can separate perfectly Hani from Latin, and perform two separate LID predictions.[4] This does not showcase the advantages of MaskLID and the performance only is dependent to the LID performance.

3) Many accessible datasets involve CS between one language and English.

Given these challenges, we decided to use datasets involving English in three sets (Turkish-English, Hindi-English, Nepali-English) and another set with CS between languages without English (Basque-Spanish). The Turkish-English and Basque-Spanish datasets are also used by Burchell et al. (2024). We use the code provided by these authors to label them into sentence-level tags.

Turkish-English. Yirmibeşoğlu and Eryiğit (2018) developed a Turkish–English dataset for CS as part of their work on CS LID for this langauge pair. The dataset is sourced from Twitter and the Ekşi Sözlük online forum. Labels in this dataset are assigned at the token level, indicating whether each token is Turkish or English. The dataset comprises 376 lines of data, and 372 of these sentences are labeled as CS. However, for our purposes, we also require monolingual datasets in these languages, not just CS data. To address this, we created a monolingual version of the CS data for the Turkish language by removing tokens labeled as English. A similar approach cannot be applied to create an English monolingual dataset, as the English parts of the data are short sentences and would adversely impact the quality of the English monolingual data. The original dataset can be found here: `github.com/zeynepyirmibes/code-switching-tr-en`.

Basque-Spanish. The Basque–Spanish corpus (Aguirre et al., 2022) comprises Spanish and Basque sentences sourced from a collection of text samples used in training bilingual chatbots. Volunteers were presented with these sentences and tasked with providing a realistic alternative text with the same meaning in Basque–Spanish CS. The dataset consists of 2304 lines of data, with 1377 sentences labeled as CS, 449 as Basque, and 478 as Spanish. The original dataset is available at:

[4] For example, GlotScript (Kargaran et al., 2024) provides a `separate_script` function that divides text based on different scripts: `github.com/cisnlp/GlotScript`.

github.com/Vicomtech/BaSCo-Corpus.

Hindi-English & Nepali-English. Aguilar et al. (2020) provide a benchmark for linguistic CS evaluation, used in previous shared tasks on CS LID (Solorio et al., 2014; Molina et al., 2016). We test on two of its language pairs, Hindi– English and Nepali-English, using the validation sets since the test sets are private. These datasets are both sourced from Twitter and are annotated at the word level. The Hindi-English dataset has 739 lines: 322 CS, 31 Hindi, and 386 English sentences. The Nepali-English dataset has 1332 lines: 943 CS, 217 Nepali, and 172 English sentences. We consider both CS and monolingual data for experiments.

Preprocessing Sentence-level LIDs may not perform well on very short sentences. In the corpus creation pipelines using these LIDs, shorter sentences are typically discarded. Therefore, we filter sentences with a length of 20 byte or fewer for monolingual sentences and sentences with a length of 40 byte or fewer for CS sentences. The remaining number of sentences (#S) for each portion of the data is detailed in Table 1. In addition, we clean user tags and emojis from the datasets before applying LIDs.

E Examples

We showcase below some failed and successful examples of MaskLID.

Failed Example. In this example, the only English word is "status".

```
yarın bir status yapıp
işlerin üstünden geçelim
```

As we define the minimum length for each selected language to be at least $\tau = 20$ byte, this sentence gets classified as Turkish, which is acceptable. If, otherwise, "status" would be evaluated alone, OpenLID would predict "Norwegian Nynorsk" language, and GlotLID "Kinyarwanda". This is the reason why τ is important to be set because otherwise the result of LID cannot be trusted. The average length of the English part of sentences in the CS Turkish-English getting classified solely as Turkish by GlotLID + MaskLID is 17.858 bytes and by OpenLID + MaskLID is 19.877 bytes. So the main reason for failing these models here is the English part of this sentences is short and often does not pass the minimum length condition.

Successful Example. In this example, "deadline crash walking I heard it at study" are the English words inserted in the Turkish sentence. These words are not next one to the other, so methods that only consider sliding windows might fail. MaskLID does not depend on the position of words in a sentence and correctly classify this example as Turkish-English CS.

```
ya deadline gelmişti çok büyük
bir crash olmuş arkadaşlarla
walking yaparken I heard it at
boğaziçi sesli study
```

However, predicting it using solely based on OpenLID results in the top 3 labels being "Turkish", "Turkmen", and "North Azerbaijani". The average length of the English part of sentences from CS Turkish-English getting classified correctly as CS Turkish-English by GlotLID + MaskLID is 42.121 bytes and by OpenLID + MaskLID is 45.294 bytes.