MaskLID: Code-Switching Language Identification through Iterative Masking

Anonymous ACL submission

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 commonly using a softmax layer and trained on monolingual texts to provide single labels. 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 paper, we explore the use of the MaskLID method for two open-source LIDs (GlotLID and OpenLID), both based on the FastText architecture. Code and demo are available at [masked link].

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), requires the development of techniques to analyze CS in written text. However, there is a lack of CS data for researchers, making it difficult to study CS and train models effectively. This shortage affects many NLP applications dealing with CS scenarios (Solorio et al., 2021; Winata et al., 2023). The creation of high-quality CS corpora depends firstly on accurately identifying CS.

To address this need, in this paper, we introduce MaskLID, a method that uses high-quality sentence-level LID to identify CS. Previous works on CS LID mainly focus on building word-level LIDs for code-switching between specific pairs of languages, often limited to only two languages (Solorio et al., 2014; Nguyen and Doğruöz, 2013; Elfardy et al., 2013; Barman et al., 2014). However, this approach is not realistic on a larger scale, especially considering that texts on the web typically lack prior information about the languages that are being used. Our primary objective is to mine web text to construct CS corpora, which can then serve as valuable training data for applications designed to handle CS scenarios.

In light of this, Burchell et al. (2024) investigate the use of high-quality LID at the sentence-level for CS LID. They reformulate CS LID to assign a set of language labels at the sentence-level, rather than at the word level. Their investigation reveals the difficulty of achieving effective CS LID with these models. Furthermore, their findings indicate that these LIDs predominantly predict only one of the languages occurring in the sentence. MaskLID follows up on this work and aims to overcome this limitation. By masking the presence of the text features associated with the dominant language, MaskLID improves the ability to recognize a second language as well. We explain in detail how MaskLID works in cooperation with LIDs based on FastText architecture (§3). We evaluate MaskLID on two test datasets containing both CS and monolingual data, showing the benefits of using MaskLID (§4).

2 Previous Work

Here, we discuss the main previous work(s). For other related works, refer to §A.

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 languages L1 and L2. In their approach, a language classifier labels the document as language L1, even though the document also contains words that exclusively 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). They 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 calculated based on the monolingual corpora they used. To relax the definition of anchors, they introduce a "weak anchor" for a language L2 relative to each 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 considered CS if and only if it is predicted to be in language L1 by an LID model and contains at least one weak anchor from the L2 anchor set. Our method shares similarity with the approach of Mendels et al. (2018) in that, for L1+L2 sentences, the initial step consists in the identification of L1. However, the concept of "anchor" differs in our approach. Instead of compiling a set of weak anchors for each language pair, we *remove by masking* the presence of L1 textual features. Subsequently, we use *the same LID* to predict the remaining text as L2. Unlike Mendels et al. (2018), our method does not require computing a set of anchors or any other resources.

Another work 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 employs binary cross-entropy loss instead of a softmax activation; and 3) Franc, which uses trigram distributions in the input text and a language model to compute languages and their scores. However, their results on CS LID are not very promising especially in Turkish-English CS dataset. One reason is that the occurrence of a single English word in a Turkish sentence is tagged as 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., 2023a) and OpenLID, makes CS LID prediction more challenging. For instance, when the model encounters a Turkish-English sentence and predicts Turkish as the top language, the second best prediction might not be English, but a language closest to Turkish instead, such as North Azerbaijani or Turkmen, which have more ngram features available in the CS text than English. Consider 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 is predicted as the 15th language. For a person knowing either Turkish or English, it is obvious that the sentence is a mixture of just these two languages. To solve this MaskLID suggests to consider masking the Turkish part of the sentence:

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

If we 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 detects masked parts are in §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. Being a simple linear classifier, FastText has the advantage of delivering explainable classification decisions. 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 extracted features. Note that these extracted features are linearly ordered, i.e., f_i precedes f_{i+1}. 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 embeddings, denoted by d, is a hyperparameter. FastText LID estimates the posterior probability for a language by applying softmax over the logits of s for each language c as:

$$P(c = i|s) = \frac{\exp(\mathbf{b}_i \cdot \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t)}{\sum_{j=1}^{N} \exp(\mathbf{b}_j \cdot \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t)} \quad (1)$$

$P(c|s)$ is the base LID probability of the input text s belonging to language c, $\mathbf{b}_c$ is the weight vector for language c in base LID, 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 the summation of the feature embeddings that build each word. Noting W the number of words in a sentence, we define the $N \times W$ matrix $\mathbf{V}$, where each element $\mathbf{V}_{c,t}$ represents the logits for language c and word-level feature t as:

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

3.2 MaskLID Algorithm

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, a 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, a 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. List of predicted languages, along with word-level features assigned to each of them.

Procedure. 1) Take sentence s, and compute $\mathbf{V}$ (Eq. 2). Estimate the probability for each possible language using Eq. 1. Find the most likely class ($L1 = \arg\max_i P(c = i|s)$) along with its corresponding probability $P(c = L1|s)$. Assign the sentence s to the variable u and L1 to variable L_u.

2) Look at the column $\mathbf{V}_{:,t}$ for each word-level feature t in u. If the value of $\mathbf{V}_{\mathrm{L}_u,t}$ is among the top-β values of the column, then assign feature t to language L_u. If the value of $\mathbf{V}_{\mathrm{L}_u,t}$ is among the top-α values of the column, mask it from the sentence u and update u accordingly.

The masking operation is aligned with (Mendels et al., 2018), since the anchors in language L2 should not be shared with anchors in language L1. By masking the features most likely belonging to L1, we may detect the presence of L2 by examining the masked sentence, which does not have any anchor from language L1.

3) check if the length of u is larger than τ. If not, then terminate. This is one termination condition (for additional considerations, refer to §B); a more naive version would just check that the masked sentence is not empty ($\tau = 0$). But it is better to use a non-zero threshold, as most sentence-level LIDs do not reliably predict short sentences.

4) predict the top label for updated sentence u and assign it to L_u, then go to back step 2 if the number of iterations is lower than λ, else stop.

4 Experiments and Results

Here, we provide an overview of our baselines and test data. We assess the performance of baselines by testing them both with and without MaskLID. Hyperparameters settings is explained in §C.1.

4.1 Baselines

Our baseline LID models are OpenLID (supporting +200 languages) and GlotLID V2 (supporting +1600 languages), both LIDs based on the FastText architecture. For a fair comparison among these models, we limit the languages that GlotLID supports to the same set as OpenLID (details in §C.2).

4.2 Test Data

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

		Baseline + MaskLID				Baseline			
		#EM ↑ / #PM ↑		#FP ↓		#EM ↑ / #PM ↑		# FP ↓	
	#S	GlotLID	OpenLID	GlotLID	OpenLID	GlotLID	OpenLID	GlotLID	OpenLID
CS Turkish–English	333	**69**/326	68/327	0	0	1/327	4/326	0	0
CS Basque–Spanish	440	43/429	**47**/426	0	0	6/428	9/424	0	3 (from Spanish)
Single Basque	357	354/354	355/355	-	-	354/354	355/355	-	-
Single Spanish	347	323/329	297/300	-	-	325/330	287/311	-	-
Single Turkish	340	323/336	329/334	-	-	331/337	329/335	-	-

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

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 number of instances rather than percentages. 1) #EM: This metric counts a prediction as a match when it exactly matches the true label. 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 settings: baseline with and without MaskLID. The best exact match for CS labels is in boldface in the table, demonstrating that the baseline with MaskLID achieves superior performance compared to the baseline without it. For CS Turkish-English with MaskLID, it detects at best 69 of the CS; however, without MaskLID, it only predicts at best 4 of them. For Basque–Spanish, with MaskLID, it detects at best 47 of CS, but without MaskLID, it only detects at best 9 of them.

#PM scores in both settings (with and without MaskLID) are quite similar. For CS labels: 1) The difference between #PM and #EM means the number of times only one of the languages in CS labels is predicted. 2) The difference between #S and #PM means the number of times none of the languages in the CS labels is predicted.

For single labels: 1) The difference between #PM and #EM means the number of times the single label language is classified as part of a CS. 2) The difference between #S and #PM means the number of times a single label is never predicted, even as part of CS. In the [single Turkish, GlotLID, baseline + MaskLID] setting, the number of incorrect CS is greater than the baseline. On the other hand, in the [single Spanish, OpenLID, baseline] setting, the number of incorrect CS predictions is greater than the baseline + MaskLID. Also, #EM is greater for baseline + MaskLID than baseline.

The difference between GlotLID and OpenLID in these scenarios shows that to achieve the best performance with MaskLID, we need to choose the best LID for our set of languages. Also, using a larger minimum length τ helps decrease the number of wrong predictions. In the [single Turkish, GlotLID, baseline + MaskLID] setting, just increasing the τ from 20 to 25 increases the #EM to the same number without MaskLID; however, it decreases the #EM for Turkish-English CS from 69 to 46. Refer to §E for examples of failures and successes of MaskLID.

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 17 in Turkish-English CS LID and by 5 in Basque–Spanish CS LID.

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 to construct CS corpora, in the hope that it will help build larger high-quality web corpora.

Limitations

The CS testsets we use in this study only represents 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. The performance of MaskLID is 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 open source LID models and does not require any additional resources. 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.

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.

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.

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.

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.

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, 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 Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023a. GlotLID: Language identification for low-resource languages. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Amir Hossein Kargaran, François Yvon, and Hinrich Schütze. 2023b. Glotscript: A resource and tool for low resource writing system identification. *arXiv preprint arXiv:2309.13320*.

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.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

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.

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.

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 this category of 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 some claiming to cover over a thousand (Kargaran et al., 2023a; Adebara et al., 2022; NLLB Team et al., 2022; Burchell et al., 2023; Dunn, 2020).

The second category receives 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 pre-known languages are aimed to be identified 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 MaskLID Additional Considerations

Here, we discuss additional considerations reagarding MaskLID.

1) Another termination condition is the number of times we repeat this process (λ), which depends on the number of languages we believe are included in the sentence (2 or 3 is recommended).

2) The parameter α could vary for each round, 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.

3) To ensure that this procedure yields a low false positive rate (FPR), the feature set assigned to language L_u in step 2 should have a minimum byte size τ. Else, increase the β value and repeat the process again to obtain a larger feature set. Then, 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 much lower than β and only target the features that strongly cue language and should accordingly be masked.

4) 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., 2023a). We suggest using a low confidence threshold for these steps or not using one at all.

C Experiment Details

We constrain GlotLID to the set of languages supported by OpenLID. The number of labels is 201 for OpenLID and 200 for GlotLID. The difference is attributed to the fact that OpenLID employs two labels for the Chinese language (zho), written in Hans and Hant scripts, whereas GlotLID combines both under the label Hani. Following the labeling proposed by NLLB Team et al. (2022), these LIDs use language-scripts as labels. They define a language-script as a combination of a ISO 639-3 language code and a script based on ISO 15924.

C.1 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 0.95 for GlotLID to evaluate the probability predictions for the feature set in step 2 of the algorithm, as further detailed in §B, item 3.

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.

C.2 Limiting LID Prediction to Fewer Languages

To restrict a FastText-based LID model to a specific subset of languages, as indicated by Eq. 1, it only needs to consider exclusively the $\mathbf{b}_c$ values for language c that are members of the specified language set. 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 matrix $\mathbf{V}$ (Eq. 2).

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) For some datasets, languages are written in a different writing system than our current baseline LID models support, such as Hindi-English (Khanuja et al., 2020), where Hindi language is written in Romanized script in CS datasets but supported with Devanagari in our baselines.

2) 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. (2023) Table 3, OpenLID exhibits the worst FPR for Standard Arabic and Egyptian Arabic among all the languages it supports.

3) 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.[1] This does not showcase the advantages of MaskLID and the performance only is dependent to the LID performance.

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

Given these challenges, we decided to use datasets involving English in one set (Turkish-English) and another set with CS between languages without English (Basque-Spanish). These 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 language pair. The dataset is sourced from Twitter and the Ekşi Sözlük online forum. The labels in this dataset is 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: `https://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: `https://github.com/Vicomtech/BaSCo-Corpus`.

[1]For example, GlotScript (Kargaran et al., 2023b) provides a separate_script function that divides text based on different scripts. The Python package is available at: `https://github.com/cisnlp/GlotScript`.

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.

E Examples

We showcase here some failed and successful examples of MaskLID.

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

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

Since we define the minimum size of features selected for each selected language to be at least $\tau = 20$ byte, the sentence gets classified as Turkish, which is acceptable. If, otherwise, "status" would be evaluated, OpenLID would predict "Norwegian Nynorsk" language, and GlotLID "Dutch". This is the reason why τ is important to be set because otherwise the result of LID cannot be trusted. The average size of the English part of sentences from CS Turkish-English getting classified solely as Turkish by GlotLID + MaskLID is 19.996 bytes and by OpenLID + MaskLID is 19.248 bytes. So the main reason for failing these models here is the English part of this sentences is short, and it does not pass the minimum length condition.

Successful Example. In this example, "deadline crash walking I heard it at study" are the words from English. These words are not necessarily next one to the other, so methods that consider sliding windows might fail. MaskLID does not depend on the position of words in a sentence and 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 45.362 bytes and by OpenLID + MaskLID is 45.465 bytes.