Word-level Language Identification using Encoder-Only and Decoder-Only Models

Javier Iranzo-Sánchez, Parnia Bahar,
Alejandro Pérez-González-de-Martos, Mattia Antonino Di Gangi
AppTek
{jiranzo,pbahar,aperez,mdigangi}@apptek.com

Abstract

Language identification is a task that often finds applications in NLP pipelines that serve multiple languages. The task is classically presented as a sentence classification problem and models' performance degrades quickly when applying them to short phrases or individual words. Although challenging, fine-grained language identification is key to improve the performance of downstream tasks. This work explores the performance of both Encoder-Only and Decoder-Only Transformer Language models for the task of automatic word-level language identification. The results show that for this particular task, small Encoder-Only models outperform larger Decoder-Only models.

1 Introduction

This paper explores Word-Level Language Identification (WLID) within the context of a cascaded Speech-to-Speech (S2S) translation system with human supervision as an example application. Although there are several promising end-to-end approaches, the cascaded approach remains the preferred choice when human intervention is desired at multiple steps of the process. For the Speech-to-Speech or dubbing task, an additional problem occurs when the text to be uttered automatically contains words belonging to a language other than the target language. These words are a source of errors because the normal rules for pronunciation of the target language cannot be applied. There are many possible sources for these words, such as named entities, slang and loanwords. Fine-grained language labels can enhance various applications, including Text-to-Speech (TTS) models, by generating more accurate phoneme sequences (Vesik et al., 2020; Zhu et al., 2022) or using language-specific embeddings (Yang et al., 2024).

The contributions of this paper are three-fold: 1) We annotate a novel dataset for the word-level language identification task under the translation setting, 2) we benchmark multiple automatic approaches to this problem, including both Encoder-Only and Decoder-Only Large Language Models (LLMs) and 3) we propose new techniques to alleviate LLMs hallucinations in the context of the WLID task.

1.1 Related work

To the best of our knowledge, there are no works that address the WLID task in the context of dubbing. The closest related task is code-switching identification, which we take as a starting point since it is the most similar. There are however significant differences between the two. Code-switching is a stylistic choice of the speaker, typically used in informal contexts, whereas this work deals with the presence of foreign words within text in the target language, which mainly occurs as a result of the translation of foreign media. Code-switching techniques and models can thus be used for this task, but the difference in domains and formality levels means that the techniques and findings of the standard code-switching approaches might not translate to this specific task. This motivates the need for specific training and evaluation data to assess and improve the performance of automatic systems.

Automatic approaches to code-switching can include both hand-crafted rules and statistical models, as well as hybrid systems that combine the two. Iliescu et al. (2021) compare multiple approaches using semi-supervised data, whereas Osmelak and Wintner (2023) train a Conditional Random Field system whose input is a sequence of word-level features. Sterner and Teufel (2023) proposed a rule-based system (TongueSwitcher) and compared it with a BERT-like model trained on the data labeled with TongueSwitcher and human labels, and observed similar performance for German-English. Additionally, much work has been done to study the effects of code-switched text on the performance

Table 1: Dataset statistics, including number of sentences (*#sent*), number of words (*#words*) and the number of those words that have been tagged as English (*#En words*).

	Spanish - TED			Spanish - Media			German - TED		
	#sent	#words	#En words	#sent	#words	#En words	#sent	#words	#En words
train	2048	39182	1849	-	-	-	1024	17719	1014
dev	1316	26076	310	-	-	-	1574	25269	300
test	2502	42294	454	1854	9959	139	2823	43197	575

of automatic models. Winata et al. (2021) compare multiple techniques and finds that good results are obtained with the XLM-RoBERTa family of models. Zhang et al. (2023) find that LLM's performance significantly decreases for code-switched data across a variety of tasks (Sentiment Analysis, Machine Translation, Summarization and Code-Switching Language Identification). Their results show that it is competitive to finetune a smaller model rather than using an LLM. In the present work, we explore further the relative performance of Encoder-Only models and larger LLMs (Decoder-Only) using different approaches.

2 Methodology

2.1 Datasets

The main dataset used for the experiments reported on this paper is the MuST-C dataset (Di Gangi et al., 2019), a Speech Translation dataset that contains the recordings of multiple English TED talks as well as their translations into multiple languages. Specifically, we used the English-Spanish and English-German translation sets. We also experimented with an in-house dataset of media content. This dataset consists of English media with translations into Spanish.

The original MuST-C dataset does not include WLID labels, so we asked 2 native speakers of the target language to annotate each set. Table 1 reports a summary of the dataset statistics. The majority of the words are in Spanish, with around 1% of the words being in English. However, 10% of the sentences contain at least 1 English word, so even if the amount of words is low, it is common enough that the user-perceived quality is affected if this issue is neglected. The manually annotated training set was constructed so that there is a 1:1 proportion between sentences with and without English words. The remaining MuST-C train sentences were automatically annotated with Llama 3.1 70B, to be used for semi-supervised experiments. [1]

2.2 Models

Both Encoder-Only and Decoder-Only models are tested based on previous results from the literature. For the first case, XLM-RoBERTa (Conneau et al., 2020) was used, in both *base* (270M) and *large* (550M) configurations. We take the pre-trained model and fine-tune it for the WLID task following a token classification approach, similarly to what is done for Named Entity Recognition (NER). Additionally, the existing Encoder-Only TongueSwitcher (Sterner and Teufel, 2023) model is also tested, which is a multilingual BERT model (Devlin et al., 2019) (172M) German-English code-switching model. The TongueSwitcher model has two versions: a pre-trained version that has been trained for the language modeling task with 24.6M Tweets that contain mixed German and English, and a code-switch detection model that has been further fine-tuned with supervised code-switching annotations. For the second case, we used Decoder-Only LLM from the Llama family. The recently released Llama3.1 (Dubey et al., 2024) 8B and 70B models were selected. After iterating through multiple prompts, we ended up with the prompt format shown in Table 2. Making the model output a label for every word in the sentence rather than only those on a different language, as well as forcing the output to be generated in a CSV-like format were significantly helpful to improve the accuracy of the model and to ensure that the model copies the input sentence.

Even after iterating multiple times to find the optimal prompt, we still observe many occurrences of hallucinations, that is, the generation of a sequence of words that differs from the original sentence to be annotated. This is not acceptable because the WLID system should add language annotations if

[1] The labels to reproduce the dataset are made available at `https://github.com/mattiadg/wlid-annotations`.

Table 2: Prompt format used for LLM inference.

Instruction	The input is a {default_language} sentence. Your task is to output the language for each word in the sentence. Write one line for each word in the original sentence. Each output line will contain the word and the language, separated by a comma and a space. If a word exists in {default_language} and other languages, write {default_language}. Only answer to the last question and do not write additional questions.
Input	He comprado un ordenador ThinkPad.
Response	He, Spanish comprado, Spanish un, Spanish ordenador, Spanish ThinkPad., English.

necessary, but leave the input text unchanged otherwise. We propose two techniques to post-process LLM hypothesis for which hallucinations are detected. The first is to replace the LLM hypothesis by the default hypothesis, which is the one where no words are labelled as a foreign language. As a second technique, we propose a post-processing algorithm called AutoMap to match the generated text against the original sentence. Specifically, we initially assign the default target language label to every word on the original sentence. Then, we take each generated word and compare it with the words in the original sentence. If there is a match, we assign the label of the generated word. Figure 1 provides an example of *AutoMap* in action.

3 Experiments

All development decisions are made based on the results on the MuST-C dev set. XLM-RoBERTa models are trained with Adam (Kingma and Ba, 2015) using 1e-5 learning rate and batch size 16, for a total of 8k steps with early-stopping every 500 steps. The learning rate is linearly scaled during the first 10% steps. Table 3 reports the results for the XLM-RoBERTa model based on the number of available training samples. Additionally, we also test wheter using the semi-supervised data annotated with Llama 3.1 is helpful, by adding 2048 sentences to the largest configuration, for a total of 4096 sentences (*+SSup*). Results are reported using the F1 score of the English class, as all of the tested configurations achieve 1.00 F1 score for the non-English class after rounding-up. The model is able to obtain acceptable results starting from 128 training samples, with increases in quality each time the available data doubles in size, starting to plateau when reaching 2048. Adding additional semi-supervised data degrades the performance rather than helping.

Table 3: XLM-RoBERTa results on the MuST-C Spanish dev set, using either the **B**ase or the **L**arge configuration. *+SSup* includes an additional 2048 examples automatically annotated with Llama. F1 scores for the English class.

	Number of training samples					
	128	256	512	1024	2048	+SSup
B	0.73	0.75	0.78	0.81	0.82	0.62
L	0.77	0.80	0.80	0.82	0.83	0.67

LLM models were tested both using the in-context learning (ICL) approach as well as fine-tuning (FT) with LoRA (Hu et al., 2022). Sampling is disabled when generating the LLM hypothesis, as we found that this helped to slightly increase quality and reduce hallucinations. Table 4 shows the performance of the LLM ICL approach on the MuST-C dev set. The train subset was shuffled once and then the first n samples were selected to be used in the prompt. That is, the example selected for $n = 1$ is also used for $n = 2$ and so on. We observe no performance improvements for increasing the number of examples beyond 1.

Table 4: LLM evaluation results for MuST-C Spanish dev set, using n in-context samples. Results show English-class F1 score.

			n			
Model	1	2	4	8	16	32
L-8B	0.54	0.52	0.49	0.47	0.50	0.50
L-70B	0.71	0.71	0.70	0.70	0.70	0.69

For fine-tuning with LoRA, the best results were obtained with learning rate 1e-4, rank 16, α =32, dropout 0.05 and 8 epochs of fine-tuning. Table 5 compares the results of both ICL and FT depending on the post-processing technique. The results high-

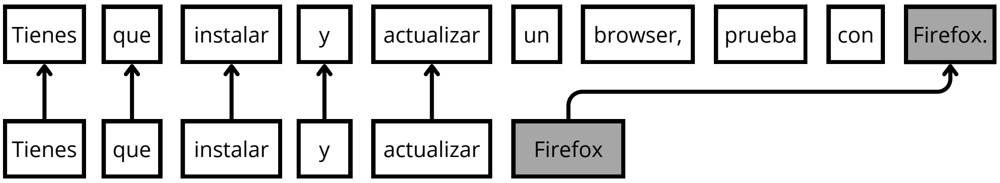

Figure 1: Example of AutoMap post-processing for LLM hallucination. The labels of the LLM hypothesis (bottom) are mapped to the original text (top) by looking for exact matches (ignoring casing and punctuation) between the hypothesis and the original text. The text is in Spanish and the shaded box represents a word detected as English. The LLM hallucinated and failed to generate a label for *"un browser, prueba con"*, which also includes the English word *browser*, so it retains the default labels for those words.

Table 5: LLM performance on the MuST-C Spanish dev set. We compare scoring the raw output (∅), using AutoMap with exact matches (Ams) and using AutoMap but ignoring casing and punctuation (Am). F1 scores for the English class.

		8B			70B	
	∅	Ams	Am	∅	Ams	Am
ICL	0.01	0.45	0.54	0.23	0.60	0.71
FT	0.01	0.45	0.59	0.01	0.56	0.72

light the importance of the AutoMap technique in mitigating hallucinations. It can be observed how results are very poor without AutoMap, as the model struggles to reproduce the input sentence. However, the introduction of AutoMap (Ams) significantly boosts the performance of the system. Results are improved further if punctuation and casing are not taken into account when looking for word matches (Am), which indicates that casing and punctuation account for a significant portion of the mistakes. When using AutoMap, the finetuned models improve the ICL results by 0.05 F1 for the 8B model, and 0.01 F1 for the 70B model. Once again, this highlights the importance of AutoMap, as it allows to extract better performance from the fine-tuned models. The results also suggest that fine-tuning is able to increase the linguistic knowledge of the model, which helps to better detect foreign words, but it is not helpful for the model to learn to copy the input.

Table 6 shows the evaluation of the final models on the selected test sets. The English-German models are also compared with two versions of TongueSwitcher: the code-switch detection BERT-based model (TS) pre-trained on ample English-German code-switching data, as well as the baseline TS model fine-tuned with our WLID data (FT-TS). Similarly to what was observed on the dev set, RoBERTa-based models outperform the Llama 3 models on the TED talks evaluation set, both for the Spanish and the German case. The TS code-switching system underperforms the other systems, and its performance only recovers when it has been trained with our WLID data (FT-TS). This highlights the need for specific data for WLID, as the existing code-switching systems cannot be directly applied to this task.

Table 6: Final evaluation results on the test sets, for XLM-RoBERTa (R-Base, R-Large) and Llama3.1 (L-8B, L-70B) models. Precision/Recall for the English class.

	Spanish				German	
	Ted		Media		Ted	
Model	P	R	P	R	P	R
R-Base	0.68	0.94	0.69	0.91	0.62	0.92
R-Large	0.69	0.98	0.73	0.94	0.68	0.92
L-8B	0.40	0.93	0.60	0.86	0.42	0.95
L-70B	0.48	0.97	0.68	0.86	0.45	0.96
TS	-	-	-	-	0.64	0.49
FT-TS	-	-	-	-	0.73	0.86

4 Conclusions

This work has introduced a new setting for word-level language identification, and provided a set of in-depth experiments to assess the performance of automatic models. Two interesting findings arise out of this research. First, there is still room for improvement on this task, on both the in-domain talks and out-of-domain media settings. Secondly, unlike current trends that tend to favor Decoder-Only LLMs, Encoder-Only models are a competitive, cost-efficient alternative for this task.

In terms of future work, Encoder-Only models can be extended to the multilingual setting in order to simplify deployment, reduce costs and to improve quality and robustness. Additionally, the performance of both Encoder-Only and Decoder-Only models should be tested on a zero-shot setting, to assess their capabilities on language pairs for which little or no training data exists.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. 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 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783v1*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Dana-Maria Iliescu, Rasmus Grand, Sara Qirko, and Rob van der Goot. 2021. Much gracias: Semi-supervised code-switch detection for Spanish-English: How far can we get? In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 65–71, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Doreen Osmelak and Shuly Wintner. 2023. The denglisch corpus of German-English code-switching. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 42–51, Dubrovnik, Croatia. Association for Computational Linguistics.

Igor Sterner and Simone Teufel. 2023. TongueSwitcher: Fine-grained identification of German-English code-switching. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–13, Singapore. Association for Computational Linguistics.

Kaili Vesik, Muhammad Abdul-Mageed, and Miikka Silfverberg. 2020. One model to pronounce them all: Multilingual grapheme-to-phoneme conversion with a transformer ensemble. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 146–152, Online. Association for Computational Linguistics.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 142–153, Online. Association for Computational Linguistics.

Huai-Zhe Yang, Chia-Ping Chen, Shan-Yun He, and Cheng-Ruei Li. 2024. Bilingual and code-switching tts enhanced with denoising diffusion model and gan. In *Interspeech 2024*, pages 4938–4942.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Aji. 2023. Multilingual large language models are not (yet) code-switchers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. ByT5 model for massively multilingual grapheme-to-phoneme conversion. In *Proc. Interspeech 2022*, pages 446–450.