In-depth evaluation of cross-domain language identification methods

Víctor Cabré Guerrero
Polytechnic University of Catalonia
IT University of Copenhagen
victor.cabre.guerrero@estudiantat.upc.edu, vcab@itu.dk

Abstract

Language identification is a fundamental Natural Language Processing (NLP) task with wide-ranging applications, from machine translation to preprocessing training data for Large Language Models. While existing models demonstrate high performance within specific domains, their effectiveness significantly deteriorates when applied across different linguistic contexts. This study presents an in-depth analysis of cross-domain language identification methods, evaluating their performance, capabilities, and inherent limitations. Our research investigates the challenges posed by the linguistic diversity found in modern communication.

Experiments conducted on a dataset spanning 2,034 languages reveal significant performance variations across domains. Models trained on specific domains like wiki, news, and religious texts show high in-domain accuracy but struggle to maintain performance when applied to different linguistic contexts. Our analysis highlights the need for more adaptable, context-aware language identification systems that can effectively handle the complexity of modern language use.

Key findings include the limited transferability of domain-specific features, the nuanced challenges of advanced tokenization, and the complex error patterns arising from language similarities and data inconsistencies. This research contributes to the ongoing dialogue about developing more robust language identification technologies that can adapt to our increasingly diverse linguistic landscape.[1]

1 Introduction

Language identification is the process of automatically determining the language of a given piece of text. It is a very common Natural Language Processing (NLP) task, and it has a wide variety of applications. For instance, it is used in machine translation to determine the language of an input, and in preprocessing pipelines for data used in training Large Language Models (LLMs) (Penedo et al., 2023). Existing models perform well within specific domains (Cavnar et al., 1994), but their performance often degrades when applied to texts from different domains. This is particularly important because the training data available for language identification models is often limited to specific domains, which constrains their ability to generalize. Some recent developments have introduced new methods, such as the use of deep neural networks, which are trained on labeled data from multiple domains.

Whereas traditional domains such as news and rights have not changed significantly, recent domains like social media have created new challenges for existing language identification technology. The use of *bad language*, defined as “text that defies our expectations about vocabulary, spelling, and syntax" (Eisenstein, 2013), introduces some phenomena which can degrade performance in current methods. Code-switching, non-standard spelling, abbreviations and unconventional characters such as emoji are some examples. Understanding the current open challenges in language identification is crucial for developing more robust and adaptable systems.

While language identification systems have shown almost perfect results in controlled settings, their performance often deteriorates significantly when applied across different domains. Text from different domains, such as social media, legal documents, and wiki data present distinct characteristics that challenge existing models. This cross-domain adaptation problem is particularly noticeable when dealing with informal language, non-standard terminology, and domain-specific abbreviations. This cross-domain scenario represents the most realistic and relevant setup, as real-world applications rarely

[1] Scripts used for the experiments are available at https://github.com/victorcabre/langid-cross-domain-eval

operate within the boundaries of a single domain.

Several factors contribute to the persistence of this challenge. First, the linguistic features that are relevant in one domain may not be equally useful in another. For example, models trained on text from rights or legal domains often struggle with social media content due to non-standard spellings, informal expressions, code-switching and non-standard characters or utterances like emojis, hashtags and usernames. Second, the distribution of languages varies significantly across domains, with some low-resource languages not being represented enough, leading to domain-specific biases in language identification systems.

In this paper, we aim to perform an in-depth analysis of different language identification methods on cross-domain data, evaluate their relative performance, and assess their capabilities and limitations, while also providing a holistic view of the current state of language identification technology.

2 Related work

Language identification has been a topic of research in NLP for a long time. High performances have been obtained with relatively simple methods, like N-gram based classification (Cavnar et al., 1994) and Naïve Bayes classifiers (Lui and Baldwin, 2012); and with more complex solutions, such as the use of static word embeddings (Mikolov et al., 2017), Long Short-Term Memory (Toftrup et al., 2021) or transformers (Imani et al., 2023).

Baldwin and Lui (2010) identified some challenges that arise when the experimental setups are less constrained. A larger number of languages, shorter or less informative documents, and an imbalance in the distribution of language classes can negatively impact performance. Lui and Baldwin (2011) show that cross-domain setups are particularly challenging, and propose a solution based on feature selection. Van der Goot (2024) proposes a standardized benchmark for the evaluation of language identification technology, and provides a dataset with 2,034 languages.

3 Models

Our research focuses on some of the most popular language identification methodologies, selecting a standardized implementation for each approach. We select the following techniques and implementations:

N-gram overlap: TextCat

Cavnar et al. (1994) propose using N-Gram frequency profiles for text categorization, such as language or domain identification. We will focus on its language identification capabilities. The approach consists of generating language profiles for each language in the training data, by computing a ranked list of the most frequent character N-grams. For every new document, a document frequency profile is computed. Then, an Out-Of-Place measure is calculated between the language profiles and the document profile, and the language with the lowest distance is selected as the output language.

This approach achieves almost perfect performance in some of the tests conducted, although Baldwin and Lui (2010) show that a higher number of languages, less training data or documents of lower length can impact performance.

We use TextCat, an implementation of the method, developed by van Noord (1997), with additional performance improvements by Rob van der Goot.

Naïve Bayes: `langid.py`

Lui and Baldwin (2012) present `langid.py`, a language identification tool that uses a Naïve Bayes classifier trained over byte n-grams. Their evaluation shows that the model is more robust on cross-domain data compared to TextCat, while outperforming it in terms of speed. Furthermore, an approach for cross-domain feature selection in language identification is proposed in Lui and Baldwin (2011). Additionally, we provide our own implementation of a Naïve Bayes-based language identification method and attempt to extract common features from multiple domains and experiment with using different tokenization strategies. We use the following hyperparameters for our implementation: alpha of 1.0, fit prior to True. We limit the feature count of the vectorizer to save memory.

Static embeddings: `fastText`

`fastText`, introduced in Joulin et al. (2017), is a text classification library developed by Facebook AI Research. While primarily designed for text classification tasks, it has been successfully applied to language identification due to its speed and robust performance across diverse datasets. The model represents words as a bag of n-grams, enabling it to capture information about local word order more effectively.

Empirical evaluations show that fastText achieves near state-of-the-art accuracy on language identification tasks, while offering significantly faster training and inference compared to more complex deep learning models.

We use the hyperparameters in Kargaran et al. (2023), namely: minimal number of word occurrences set to 1,000; number of buckets set to 1,000,000; minimum and maximum n-gram length set to 2 and 5, respectively; size of word vectors set to 256; 2 epochs; and a learning rate of 0.8. All other parameters are set by default. We use 8 threads for training.

4 Data

The data used in the experiments was made available by van der Goot (2024). The dataset contains over 6 million utterances, each one with at most 100 characters. The data has been sourced from a variety of fields that cover multiple domains, including wiki, news, and rights texts. It does not include code-switched utterances. A portion of the data is provided separately, categorized by domain.

4.1 Sources

The following datasets are used. All data sources are publicly available.

MIL-TALE Brown (2014) provides a dataset that covers over 1,300 languages. The data is sourced from Bible translations, Wikipedia, and the Europarl corpus.

Universal Declaration of Human Rights The Universal Declaration of Human Rights is a concise and standardized text, approximately 90 lines long, which has been translated into numerous languages.

OpenLID Burchell et al. (2023) compile data from multiple sources, including news sites, Wikipedia, and religious texts. They highlight that they select these sources in order to avoid mislabeled samples and, as such, the majority of text is of a formal nature.

MassiveSumm A multilingual summarization dataset is presented in Varab and Schluter (2021). It contains 92 languages and over 28 million articles. We use the original, non-summarized texts.

Twituser A dataset focusing on the social domain is presented in Lui and Baldwin (2014). Users' tweets are collected based on their language if their profile is mostly monolingual.

Universal Dependency A very diverse dataset covering different domains and languages, curated by hundreds of human annotators. We use version 2.12 (Zeman et al., 2023).

4.2 Data preprocessing

The data cleaning procedure consists of grouping utterances by language label and dataset; removing data from dialects, macrolanguages and language codes different from ISO-639-3; standardizing the domain labels; ensuring the script in the data is consistent with the script of the language based on ISO 15924; and manually inspecting and removing mistakes in the data, such as the presence of XML tags. After applying this procedure, the data contains over 6 million utterances spanning 2,034 languages. The total language coverage is close to 50%, approximately (van der Goot, 2024).

5 Analysis of performance and resource-efficiency

In recent years, resource efficiency has emerged as a growing priority in the NLP community. This shift reflects increasing awareness of the environmental and computational costs associated with training and deploying large-scale language models. Researchers are now focusing on developing methods that optimize resource utilization, such as reducing energy consumption and minimizing memory usage, without compromising model performance.

In this section, we evaluate the performance of different language identification methods, taking into account their resource-efficiency. We will focus on their accuracy metrics, their execution time, their number of floating point operations (FLOPS) and their power consumption. We follow the guidelines established in Dürlich et al. (2023).

All the experiments are conducted under the same conditions, using the following hardware:

- AMD EPYC 7742 64-Core Processor
- 256 GiB RAM
- No GPU acceleration

All reported execution times represent the total duration, measured as the elapsed time from the beginning to the completion of execution. We

ID	# of utterances	Size (MB)
all.train	6,101,997	625
all.0.train	2,033,999	218
cut2.train	1,000,000	107
cut.train	500,000	54

Table 1: Number of utterances and disk usage of different data splits.

choose this metric because the `user` and `sys` metric reported by the `time` command in UNIX systems provide the total CPU time in either kernel or user mode, for each thread.

For the calculation of FLOPS, we use the formulas suggested in Kaplan et al. (2020), for training and inference, respectively:

$$\text{FLOP}_t = 6 \cdot n \cdot N \cdot S \cdot B \tag{1}$$

$$\text{FLOP}_i = 2 \cdot n \cdot N \cdot S \cdot B \tag{2}$$

where n is the length of the sequence, N is the total amount of parameters of the model, S is the number of steps (training or inference), and B is the batch size.

To estimate power consumption, we calculate the total energy usage in watt-hours (Wh) based solely on CPU utilization, disregarding other components like memory or disk. In addition, CPU power consumption can fluctuate depending on the workload. To simplify, we use the CPU's Thermal Design Power (TDP) of 225 W[2] as an upper bound for energy consumption.

The FLOPS and power consumption metrics pertain to the entire dataset (`all.train`).

5.1 Training

We train `textcat`, Naïve Bayes and `fastText` on the data, and record execution time. In this experiment, we train `textcat` on a single thread, since it doesn't provide a way to parallelize the execution, to the best of our knowledge. Naïve Bayes is trained on a single thread for the same reason. `fastText` is being executed on 8 threads, simulating a mid-range CPU.

In order to test the effect of data size on training time, each model is trained four times, each on a different split of the data. Table 1 shows the different splits that are available, their number of utterances and the size of the dataset. The different splits were obtained following a stratified sampling approach.

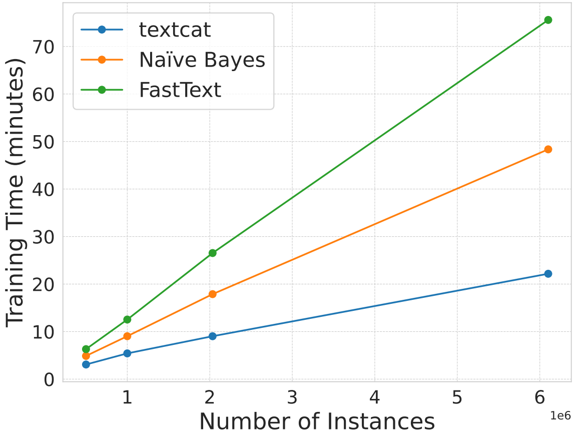

Figure 1: Effect of number of instances on training time.

The results are reported in figure 1. They suggest that `textcat` is the most efficient model for training, followed by Naïve Bayes and `fastText`. Both Naïve Bayes and `textcat` are considerably fast at training. The first only requires calculating the probabilities for each class and feature, which involves relatively simple arithmetic operations; the second needs to compute profiles for every language. While `fastText` demonstrates lower overall performance, even when utilizing 8 threads, its multithreaded capabilities allow for improved performance scaling with additional processor cores, offering strong potential for high-throughput computing environments. All models exhibit linear growth as the number of instances increases.

5.2 Inference

We predict the labels of 10,000 utterances using the three models. All the utterances are passed at once to the models, i.e. the programs are not being relaunched constantly. Detailed results are provided in table 2.

5.3 Summary of results

Table 2 compares `textcat`, Naïve Bayes, and `fastText` across training and evaluation stages, focusing on execution time, computational resources, and accuracy.

The data suggests that `fastText` offers the most attractive combination of high accuracy, relatively moderate training resources, and extremely low evaluation time and power consumption. While it's not the fastest to train, its evaluation efficiency and superior accuracy make it the most promising

[2] https://www.amd.com/en/products/specifications/server-processor.html

Model	Training stage Time (m)	 Power (Wh)	 FLOPS	Evaluation stage Time (m)	 Power (Wh)	 FLOPS	Acc
TextCat	22.17	83.14	$4.88{\cdot}10^{8}$	44.23	165.86	$1.62{\cdot}10^{8}$	88.09
Naïve Bayes	48.35	181.31	$1.48{\cdot}10^{12}$	0.07	0.29	$4.94{\cdot}10^{11}$	90.18
fastText	75.59	283.46	$1.23{\cdot}10^{20}$	0.09	0.34	$4.12{\cdot}10^{19}$	93.43

Table 2: Summary of results. All models are trained using the `all.train` split.

model for this text classification task. `textcat` is the most computationally intensive model, demanding substantial resources for both training and inference. Despite these extensive computational demands, it delivers the lowest accuracy at 88.09%, which raises questions about the efficiency of its resource allocation. Naïve Bayes provides a good balance between computational efficiency and accuracy, potentially making it a strong alternative for resource-constrained environments.

6 Experiments

In this section, we perform an evaluation of the performance of some commonly used language identification methods. We start with an experiment on the effect of training a model with data from specific domains and seeing how well it generalizes to other areas. We also look at the effect of different tokenization strategies on the results, and attempt to create a domain-agnostic version of the model using feature selection.

6.1 Domain effect

We start with an experiment on the effect of domain-specific training data to see whether models can generalize to other domains. In this experiment, we focus on Naïve Bayes-based methods. For training, we use data from the following domains: wiki, news, religious, and combined (the data from all domains aggregated). We test the models' performance on the development dataset with the following domains: wiki, news, religious, rights, and social. We emphasize that the rights and social domains are not used for training due to their considerably smaller size.

Figure 2 shows the accuracy of each one of the models evaluated on data from multiple domains, and table 3 shows the average, non-weighted[3] accuracy of each model. As expected, the wiki, news, and religious models perform exceptionally well

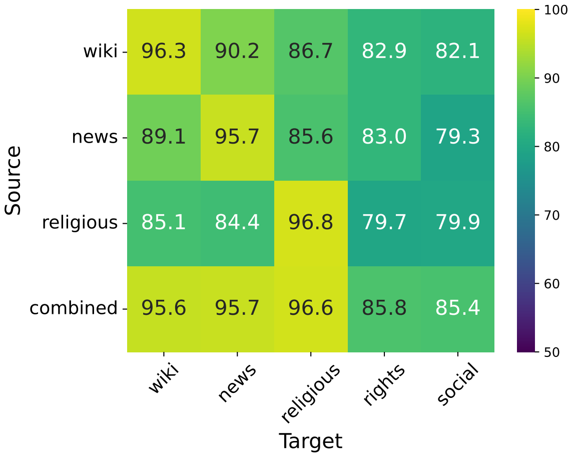

Figure 2: Accuracies of cross-domain configurations.

Source domain	Acc	Acc (GLOT500)
wiki	87.64	83.18
news	86.54	82.16
religious	85.18	81.24
combined	91.82	89.68

Table 3: Average, non-weighted accuracy of models trained on different domains, using the standard tokenizer (section 6.1) and the GLOT500 tokenizer (section 6.2).

[3] We compute a non-weighted average, meaning we don't take into account the amount of instances in each one of the domains, because we want to focus on the models' generalization capabilities.

in-domain, reporting scores of up to 96.8%. However, these models do not generalize well to other domains, with accuracy metrics dropping as low as 79% in some cases. Here, we highlight that some domain pairs show less effect of this generalization gap, particularly when the domains are more similar to each other. For example, the performance of the wiki model on the news domain is notably higher compared to its performance on more dissimilar domains, such as rights and social. This suggests that models trained on domains with overlapping content tend to have better accuracy when tested on those similar domains, as the underlying features may be more transferable. We also emphasize that the model trained on the combined data is the best-performing, reaching almost 92% accuracy and surpassing the second-best model, wiki, by over 4 points.

6.2 Effect of using different tokenizers

We also test the effect of using a different tokenization strategy, following the same setup as in section 6.1. The objective is to investigate whether advanced tokenization techniques can overcome the limitations of more simple tokenizers, particularly when dealing with the linguistic diversity of our dataset. We aim to improve language identification accuracy, especially for low-resource and morphologically complex languages that may be challenging for standard tokenization approaches. We use the tokenizer provided in the GLOT500 model (Imani et al., 2023). We use the base version (`cis-lmu/glot500-base`[4]), with the count vectorizer configured with an N-gram range of 1 to 2.

Figure 3 shows the accuracy of each one of the models evaluated on data from multiple domains. Table 3 shows the average accuracy of each one of the models, and a comparison to the default tokenization strategy. Unexpectedly, the results on average are significantly worse compared to the models that used the standard tokenizer. However, there is a small but consistent increase in in-domain accuracy. We point out that the domain that is most affected is social.

We also test for the effect of the N-gram range, using values of (1,1) and (1,4), but we don't observe any meaningful difference. It is unclear why the model is performing poorly with the more advanced tokenizer. Further investigation is needed to understand the underlying causes of the performance degradation and determine potential solutions.

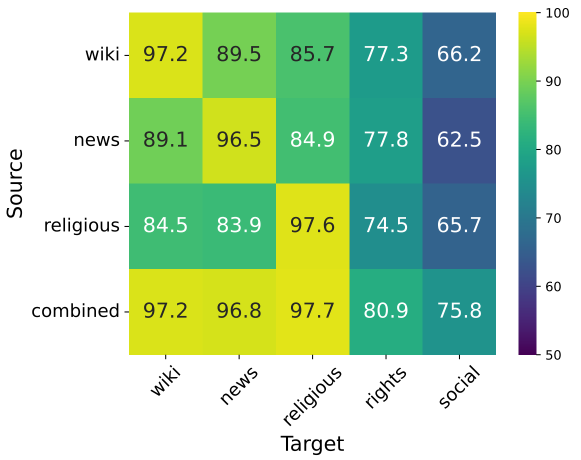

Figure 3: Accuracies of cross-domain configurations using GLOT500 tokenizer.

Target	Accuracy
wiki	85.4
news	89.4
religious	88.9
rights	70.4
social	68.2
Average	80.46

Table 4: Accuracy of the domain-agnostic model on each target domain.

6.3 Domain-agnostic model

We attempt to create a domain-agnostic model that can generalize better across domains. To develop a robust cross-domain model, we first conduct an analysis of the feature spaces across different domains (wiki, news, and religious texts). We extract and examine the common linguistic features that are independent of domain-specific nuances, and train a new model based on these features.

Table 4 shows the accuracy of the model on each target domain. Overall, the performance is decreased. The social domain is particularly affected, with accuracy dropping below 70%. It is likely that this performance degradation is caused by an insufficient amount of information, since all domains are affected. The new model's vectorizer has 335,804 features, whereas the original vectorizers had significantly more (e.g. the model trained on news data has 1,215,140 features). This represents a loss of over 70% of features. The substantial reduction in feature dimensionality suggests a critical loss of

[4] https://huggingface.co/cis-lmu/glot500-base

discriminative information across domains, which leads to decreased performance.

7 Analysis of errors

In this section, we conduct a comprehensive analysis of the errors encountered during our language identification experiments. Understanding the sources and patterns of misclassification is crucial for improving the model's performance and identifying potential limitations in our approach.

We highlight that the procedures followed in this section are, above all, approximations. Our methodology relies on empirical observations, which cannot capture the full complexity of linguistic variation.

We categorize these errors into several key types to provide insights into the challenges of cross-domain language identification:

- **Language similarity (LS):** Lexically similar languages can pose a challenge for language identification methods. We define similar languages as those that have a similarity score greater than 0.4. Appendix A describes how similarity scores are calculated.

- **Data errors (DE):** Errors in the dataset can result from annotation mistakes, inconsistencies in text formatting, or noisy input such as improperly cleaned text that includes residual XML and HTML tags, or other markup.

- **Other (O):** Any reason that doesn't fit into the previous categories. This includes low-resource languages, problems with the scripts used in the data, code-switching, or an unknown reason.

Figure 4 shows that the error distribution follows a Zipfian distribution. The Zipfian distribution manifests as a pattern where the first few error categories account for a significant portion of the total error, and subsequent error types follow a rapid decline in frequency. This suggests that addressing just a few key error sources could potentially mitigate a large proportion of the overall error.

We identify the 30 language pairs with the most errors in table 5. The number of errors of a language pair is the total amount of errors made by the model, regardless of direction (i.e. wrongfully classifying an utterance written in language A as language B, or vice versa).

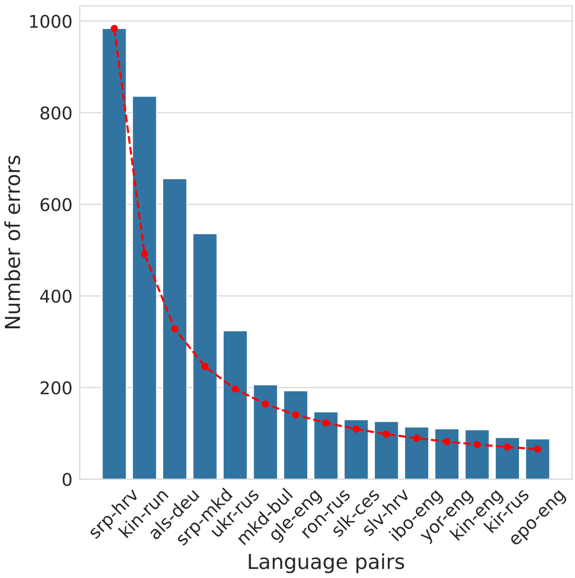

Figure 4: Amount of errors in the top 10 language pairs by number of errors.

We highlight that, out of the 30 language pairs, 12 of them contain English. This amounts to 40% of the top 30. Our analysis reveals that a lot of English samples are wrongfully annotated as other languages, including Irish (gle), Esperanto (epo), and Igbo (ibo), among others. In addition, we point out that the languages *srp-hrv* (Serbian and Croatian) indeed have a high language similarity, but their data uses different scripts. Our Levenshtein similarity metric is not able to account for that. It is a limitation of this method.

8 Conclusion

Language identification across diverse domains remains a complex and challenging task in Natural Language Processing. Our research provides an in-depth analysis of the performance, limitations, and potential improvements of language identification methods when applied to cross-domain data. Our key findings highlight several critical insights:

- Domain-specific performance variability: Models trained on specific domains like wiki, news, and religious texts demonstrate high in-domain accuracy but struggle significantly when generalizing to other domains. This emphasizes the critical challenge of developing robust, domain-agnostic language identification systems.

- Tokenization limitations: Contrary to expectations, an advanced tokenization technique

Language pair	Similarity	Reason
srp-hrv	0.0028	LS
kin-run	0.6930	LS
als-deu	0.1766	DE
srp-mkd	0.6818	LS
ukr-rus	0.5629	LS
mkd-bul	0.6799	LS
gle-eng	0.2189	DE
ron-rus	0.0014	DE
slk-ces	0.6118	LS
slv-hrv	0.7551	LS
ibo-eng	0.1917	DE
yor-eng	0.1001	DE
kin-eng	0.2184	DE
kir-rus	0.1462	DE
epo-eng	0.2710	DE
cat-eng	0.2393	O
run-slk	0.1433	O
mal-eng	0.0353	DE
tgk-rus	0.1798	O
mni-ben	0.0000	O
swe-eng	0.3376	DE
tam-eng	0.0405	DE
sna-eng	0.3195	DE
nld-eng	0.3482	DE
hin-npi	0.4269	LS
mar-hin	0.3162	O
cat-fra	0.4556	LS
eng-deu	0.3069	DE
cat-spa	0.5573	LS
ron-ita	0.4237	LS

Table 5: Ranking of the 30 language pairs with the most errors and the manually identified reason for the mistakes.

did not improve language identification performance. In our setup, the GLOT500 tokenizer showed decreased accuracy across domains, suggesting that more advanced tokenization does not automatically translate to better generalization.

- Error analysis observations: Our error analysis revealed significant challenges, in particular with language similarity, where lexically similar languages pose substantial identification challenges; and data quality, because annotation errors and inconsistent text formatting contribute to misclassification.

The dynamic nature of modern communication demands a paradigm shift in language identification technologies. Future research should focus on developing more adaptable models that can effectively handle linguistic diversity, creating robust feature selection techniques that surpass domain-specific limitations, addressing challenges with low-resource languages and non-standard text representations and investigating advanced machine learning approaches that can better capture cross-domain linguistic nuances. Additionally, research should focus on developing methods which are less resource-intensive and more environmentally friendly.

Our research contributes to the growing understanding of language identification challenges in an increasingly diverse digital landscape. By highlighting the limitations of current approaches, we provide a foundation for more sophisticated, context-aware language identification systems. As communication continues to evolve, particularly with the rise of social media and informal digital platforms, developing flexible and accurate language identification technologies becomes ever more crucial.

Limitations

We only examine languages included in the ISO-639-3 standard, which doesn't provide full language coverage and can be problematic from a social standpoint (Morey et al., 2013).

Domain coverage is limited to wiki, news, religious, rights, and social texts, which doesn't include many potential communication domains such as academic publications, technical documentation, literary works, or specialized contexts.

Execution times are not completely accurate. They include processes not related to the models, such as OS scheduling tasks. However, we choose this metric in order to be able to account for multithreaded execution. Additionally, we perform each test three times to mitigate this effect on the reported metrics.

References

Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237, Los Angeles, California. Association for Computational Linguistics.

Ralf Brown. 2014. Non-linear mapping for improved identification of 1300+ languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 627–632, Doha, Qatar. Association for Computational Linguistics.

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.

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175, page 14. Ann Arbor, Michigan.

Luís Morgado da Costa, Francis Bond, and František Kratochvíl. 2016. Linking and disambiguating swadesh lists. In *GLOBALEX 2016 Lexicographic Resources for Human Language Technology Workshop Programme*, page 29.

Luise Dürlich, Evangelia Gogoulou, and Joakim Nivre. 2023. On the concept of resource-efficiency in NLP. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 135–145, Tórshavn, Faroe Islands. University of Tartu Library.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Amir Hossein 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.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 553–561, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of Twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 17–25, Gothenburg, Sweden. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *Preprint*, arXiv:1712.09405.

Stephen Morey, Mark W Post, and Victor A Friedman. 2013. The language codes of iso 639: A premature, ultimately unobtainable, and possibly damaging standardization.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza

Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Mads Toftrup, Søren Asger Sørensen, Manuel R. Ciosici, and Ira Assent. 2021. A reproduction of apple's bi-directional LSTM models for language identification in short strings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 36–42, Online. Association for Computational Linguistics.

Rob van der Goot. 2024. Identifying open challenges in language identification. Manuscript under review.

Gertjan van Noord. 1997. Textcat language guesser. Accessed: 29 october 2024.

Daniel Varab and Natalie Schluter. 2021. MassiveSumm: a very large-scale, very multilingual, news summarisation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150–10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Lene Antonsen, Tatsuya Aoyama, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, H̑órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Mariana Avelãs, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Shabnam Behzad, Kepa Bengoetxea, İbrahim Benli, Yifat Ben Moshe, Gözde Berk, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, António Branco, Kristina Brokaitė, Aljoscha Burchardt, Marisa Campos, Marie Candito, Bernard Caron, Gauthier Caron, Catarina Carvalheiro, Rita Carvalho, Lauren Cassidy, Maria Clara Castro, Sérgio Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Francisco Costa, Marine Courtin, Mihaela Cristescu, Ingerid Løyning Dale, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Adrian Doyle, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Masaki Eguchi, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Farah Essaidi, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Amal Fethi, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Fabrício Ferraz Gerardi, Kim Gerdes, Luke Gessler, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóǧa, Andre Kåsen, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Kris Kyle, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Lauren Levine, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Yi-Ju Jessica Lin, Krister Lindén, Yang Janet Liu, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena

Martín Rodríguez, André Martins, Cláudia Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Aaron Miller, Karina Mischenkova, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Siyao Logan Peng, Rita Pereira, Sílvia Pereira, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Yuval Pinter, Clara Pinto, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andreia Querido, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Joana Ramos, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Arij Riabi, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Silva, Aline Silveira, Natalia Silveira, Sara Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Ted Sither, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Per Erik Solberg, Barbara Sonnenhauser, Shafi Sourov, Rachele Sprugnoli, Vivian Stamou, Steinh͡ór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn H͡órðarson, Vilhjálmur H͡orsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Socrates Vak, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Yilun Zhu, Anna Zhuravleva, and Rayan Ziane. 2023. Universal dependencies 2.12. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Language similarity computation procedure

This appendix describes the procedure for computing the similarity of two languages. Initially, we used `lang2vec` (Littell et al., 2017) to compute language distance. However, lexical similarity is more relevant in our case, since each utterance only has lexical information, and not other information that `lang2vec` may encode, such as phonetic similarity. Therefore, we use Swadesh lists and calculate the Levenshtein similarity between two sets of commonly used words.

The Swadesh lists were sourced from PanLex[5] and da Costa et al. (2016). The Swadesh lists of the following missing languages (with their corresponding ISO 639-3 code) were synthetically generated using Google Translate[6]: Kinyarwanda (kin), Rundi (run), Igbo (ibo), Manipuri (mni), Shona (sna), Nepali (npi), and Marathi (mar). The words were translated from English to these target languages.

[5] https://old.panlex.org/

[6] https://translate.google.com

The procedure we follow to calculate language similarity is as follows:

1. Compute a set of common words between the Swadesh lists of the two languages[7].

2. For each word pair, calculate their normalized insertion-deletion (indel) distance. The indel distance is the minimum number of insertions and deletions required to change one word into the other. Then, normalize it using the following formula:

$$\text{normalized similarity} = 1 - \frac{\text{indel distance}}{\text{len1} + \text{len2}}$$

   where len1 is the length of the first word and len2 is the length of the second word.

3. Calculate the average of all the similarities. This is the final language similarity score.

[7] This is necessary because each Swadesh list has a different set of words. There is substantial overlap between the two lists in most cases.