Tokenization and Representation Biases in Multilingual Models on Dialectal NLP Tasks

Vani Kanjirangat[1], Tanja Samardžić[1], Ljiljana Dolamic[2], Fabio Rinaldi[1]

[1]SUPSI, IDSIA, Switzerland
[2]armasuisse S+T, Switzerland

{vani.kanjirangat, tanja.samardzic, fabio.rinaldi}@supsi.ch, Ljiljana.Dolamic@armasuisse.ch

Abstract

Dialectal data are characterized by linguistic variation that appears small to humans but has a significant impact on the performance of models. This dialect gap has been related to various factors (e.g., data size, economic and social factors) whose impact, however, turns out to be inconsistent. In this work, we investigate factors impacting the model performance more directly: we correlate Tokenization Parity (TP) and Information Parity (IP), as measures of representational biases in pre-trained multilingual models, with the downstream performance. We compare state-of-the-art decoder-only LLMs with encoder-based models across three tasks: dialect classification, topic classification, and extractive question answering, controlling for varying scripts (Latin vs. non-Latin) and resource availability (high vs. low). Our analysis reveals that TP is a better predictor of the performance on tasks reliant on syntactic and morphological cues (e.g., extractive QA), while IP better predicts performance in semantic tasks (e.g., topic classification). Complementary analyses, including tokenizer behavior, vocabulary coverage, and qualitative insights, reveal that the language support claims of LLMs often might mask deeper mismatches at the script or token level[1].

1 Introduction

Large Language Models (LLMs) pre-trained on massive text data in many languages have become the preferred solution for various Natural Language Processing (NLP) tasks. The use of this technology for processing dialects and regional varieties remains limited. Small variations in pronunciation and writing (Zampieri et al., 2018; Scherrer et al., 2023; Habash et al., 2024), which humans can easily ignore, lead to significant performance drops known as the *dialect gap* (Kantharuban et al., 2023). Including this variation is hard, although important for more human-like interactions with LLMs (Amadeus et al., 2024). It is especially important for a wide linguistic coverage, as many languages are not standardized or have multiple standards (Samardžić and Ljubešić, 2021).

In previous studies, dialect variances have been related to economic and social factors (Kantharuban et al., 2023), but the effects were inconsistent across different settings. Looking for more consistent factors directly related to how LLMs work, we turn to the representational biases in multilingual LLMs.

We study two aspects where the biases can be quantified with recently proposed measures. First, the tokenization bias has been shown to impact not only the performance, but also the costs of deploying LLMs across languages (Ahia et al., 2023). Recently, this bias was quantified as Tokenization Parity (TP) (Petrov et al., 2024). Second, Information Parity (IP) (Tsvetkov and Kipnis, 2024) measures how well an LLM compresses or represents the same content across languages. In both cases, the measures show a difference between a given language and English as a reference language.

To address the dialect gap, we correlate these measures to downstream performance on three dialect NLP tasks, each targeting a different level of representation: Dialect Identification (DI), which mostly relies on surface-level clues, Topic Classification (TC) as a primarily semantic task, and Extractive Question Answering (EQA) as a task that relies on both kinds of features. In all three cases, we work with multiple data sets representing different economic and cultural settings. This allows us to control for additional factors that are known to play an important role in creating biases. In particular, we control for the script (Latin vs. non-Latin) and resource level (high vs. low) (van Esch et al., 2022). On the model side, we control for the general type of pre-trained LLMs, distin-

[1] Code at https://github.com/vanikanjirangat/Tokenizer_Fairness_Dialect

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 24004–24022
November 4-9, 2025 ©2025 Association for Computational Linguistics

guishing between encoder-only (BERT-based) and decoder-only (E.g., GPT) multilingual models.

The **Key Findings** are:

1. Encoder-based models consistently outperform decoder-only LLMs[2] across the evaluated dialectal tasks.
2. TP is more sensitive to the type of the script, while IP reflects biases influenced by both script and resource availability. Additionally, both metrics show model-dependent variation, highlighting how architectural and training differences contribute to representational disparities.
3. Information Parity (IP) shows more substantial alignment with tasks requiring semantic understanding and complex reasoning, while Tokenization Parity (TP) is more predictive for tasks that rely on morphological and syntactic features, especially span-based extractive tasks. These correlations are further modulated by language resource availability and script type.

2 Dialect Tasks, Data and Models

Our selection of dialect NLP tasks, data and models was guided by the goal of covering as diverse settings as possible while keeping the computation feasible.

2.1 Tasks

Dialect Identification (DI) This task consists of assigning a dialect or region label to each input sentence or utterance. This task comes in two versions: monolabel (each utterance can belong to only one dialect) and multilabel (some utterances can belong to multiple dialects), with the latter being more realistic but harder to perform and evaluate. We used the datasets from several VarDial shared tasks: Nuanced Arabic Dialect Identification (NADI-2023), Swiss German dialect identification (GDI), Indo-Aryan Language Identification (ILI), and multi-label DSL-ML datasets (Abdul-Mageed et al., 2023; Samardzic et al., 2016; Zampieri et al., 2018; Chifu et al., 2024).

Topic Classification (TC) This task is similar to the monolabel DI task in that each snippet of text is assigned a single topic label. The difference is that predicting the label requires neutralizing surface-level differences between dialects. This task is included in the DialectBench benchmark (Faisal et al., 2024) as the SIB-200 dataset (Adelani et al., 2024) representing 200 languages. The topic classes include: {science/technology travel, politics, sports, health, entertainment, geography}. We conducted the fine-tuning experiments on 29 languages belonging to different scripts, along with the availability of language resources: eight Latin-high, nine Latin-low, five non-Latin-high, and seven non-Latin-low.

Extractive Question Answering (EQA) This task combines in some way the features of the previous two, as it requires identifying the relevant spans (surface features) but relying on deeper semantic representation (understanding the question-answer relationships). We experimented on 24 dialectal variants - eleven Latin-high, two Latin-low, nine non-Latin-High, and two non-Latin-low, from the dataset SDQA (Faisal et al., 2021) also provided via the DialectBench.

The general statistics and further details of datasets are given in Appendix A. The class distributions for DI and TC tasks are shown in Figures 7, 8 in Appendix A.

3 The Bias Metrics

3.1 Models & Tokenizers

Encoder type We used the multilingual mBERT for the encoder variant in all the tasks. Specifically, in the case of the DI task, we performed additional comparisons between mBERT and language-specific variants such as MARBERT (Arabic), IndicBERT(Indic), German-BERT[3] (Swiss-German), SpanBERTa (Spanish), and CamemBERT (French). We use the respective models from HuggingFace[4].

Decoder type Among the multilingual decoder-type models, we selected Phi-3.5-mini, Llama 3.2-3B, Mistral-7B, Falcon-7B, Gemma-7B, and SILMA-9B models. SILMA-9B represents an Arabic-specific LLM, while the other models discussed are English-centric or generalized multilingual LLMs, claiming support for a broad spectrum of languages. For the downstream task performance evaluation, we considered Phi-3.5 and Llama-3.2 by supervised fine-tuning (SFT) experiments to compare with the encoder variants.

[2] All models are referred to as LLMs and are distinguished as encoder-only or decoder-only where relevant

[3] We also did experiments with Swiss-BERT, which gave similar performance as German-BERT

[4] https://huggingface.co/

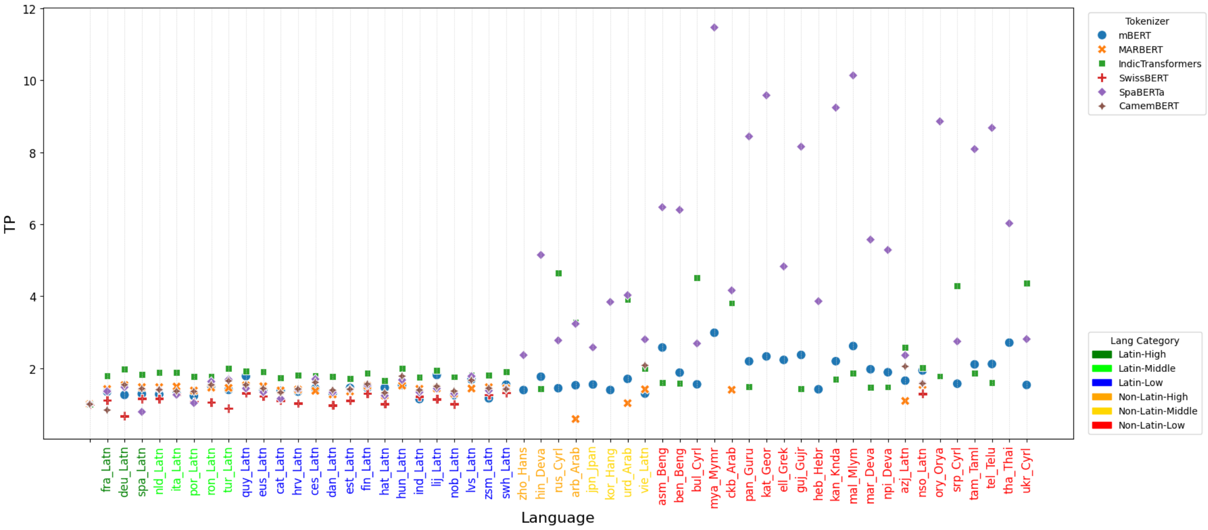


Figure 1: Tokenization Parity (TP) across languages resulting from the tokenizers used in encoder-type models.

Tokenizer & Languages: Language and tokenization are deeply intertwined, shaping LLMs' multilingual capabilities. Most current models use subword tokenization strategies such as BPE, SentencePiece, or byte-level methods. Newer models like LLaMA and Phi adopt the OpenAI tiktoken tokenizer[5], which operates at the byte level using UTF-8 encoding. This approach is language-agnostic, breaking input into bytes or fragments when unknown tokens are encountered. In contrast, SentencePiece typically defaults to character-level segmentation. Non-Latin scripts (e.g., Arabic, Hindi, Bengali) involve multi-byte characters in UTF-8, making them more prone to token fragmentation under byte-level fallback. This behavior impacts the vocabulary coverage and can hinder effective representation of non-Latin text. Details on tokenizer configurations and vocabulary sizes are provided in Table 3, Appendix B. The fairness or, inversely, the biases of pre-trained multilingual models can be measured considering either the surface level or deeper semantic features.

Tokenization Parity Following Petrov et al. (2024), we use TP as a metric to analyze the tokenization fairness. The metric systematically assesses how well the tokenizers treat parallel sentences across different languages. Parity occurs when a tokenizer exhibits similar tokenized lengths for the same sentence in different languages. Consider a sentence s_A in language A and its translation s_B to language B. Then, a tokenizer t achieves parity for A with respect to B at s_A and s_B if $|t(sA)|/|t(sB)| \approx 1$, where $t(s_A)$ is the tokenization of the sentence sA and $|t(s_A)|$ represents its length. The premium for A relative to B is the ratio $|t(s_A)|/|t(s_B)|$ (Petrov et al., 2024). A value close to 1 indicates fewer splits into subwords, which indicates that the tokenizer vocabulary covers the language well. When the value is greater than 1, it indicates that the language tokenizer requires more tokens to represent the same content. This may indicate a suboptimal representation of the language by the LLM, leading to inefficient representation and potentially poorer downstream task performance. At the same time, these values are language-dependent, and hence, the number of tokens required to represent the same sentence in different languages can affect TP values.

Information Parity Following Tsvetkov and Kipnis (2024), we adopt Information Parity (IP) as another metric for evaluating multilingual, specifically dialectal fairness in large language models (LLMs). IP draws on information-theoretic principles and quantifies the LLM's efficiency in compressing text in a given language relative to a reference language. For a text in language L, IP is defined as the ratio between the negative log-likelihood of the text in English and the negative log-likelihood of the same text in language L. In this context, English serves as a language-agnostic reference compressor. IP expresses the total amount of information or uncertainty in a sequence perceived by the LLM relative to the reference language. Unlike similar metrics such as perplexity, IP is less sensitive to variations in tok-

[5] https://github.com/openai/tiktoken

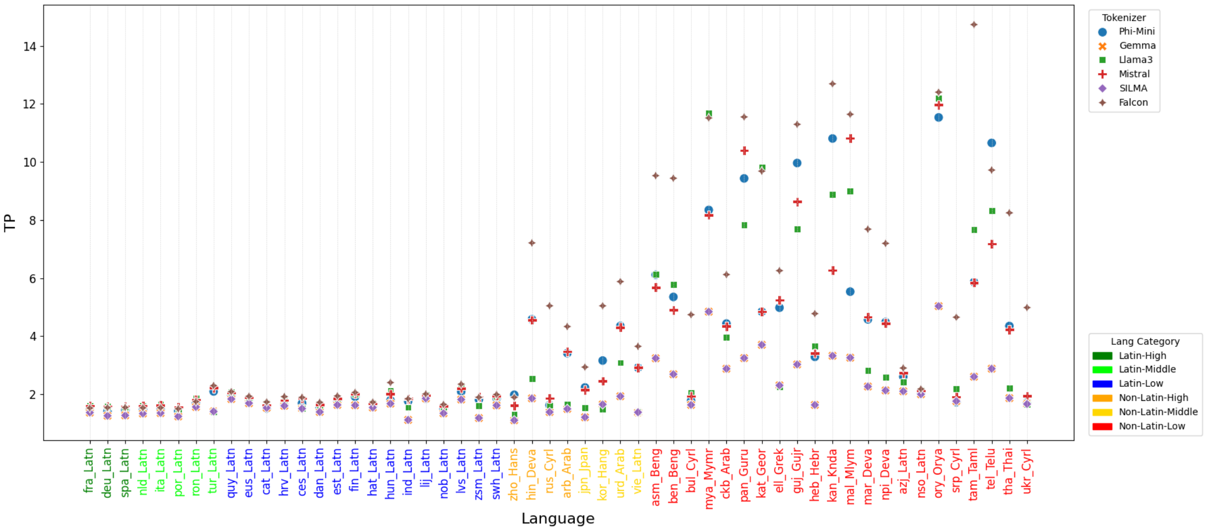


Figure 2: Tokenization Parity (TP) across languages resulting from the tokenizers used in decoder-type models.

enization across languages and models.

4 Experiments

Our first goal is to evaluate the LLMs' performance on the dialectal downstream tasks, controlling for various factors, namely scripts (Latin vs. non-Latin) and resource levels (High vs. Low). The latter categorization can be slightly biased as sometimes the distinction between high, medium, and low resources can be fine-lined. The categorization is reported in Table 4 of Appendix C. We then quantitatively analyze the model's script and representation biases, measuring the correlation between the observed performance on one side and the two bias metrics — tokenization parity (TP) and information parity (IP) — on the other. We complement these analyses with a vocabulary analysis and a manual inspection of the model tokenizers' output.

4.1 Model Fine-Tuning Methods and Parameters

We performed supervised fine-tuning (SFT) of the decoder-only LLMs - Phi-3.5 and Llama 3.2 models and compared them with encoder-only models, mainly mBERT, on the datasets described in Section 2. We decided to select fewer representative models to economize computing time. On the other hand, the parity score does not require a lot of computation, so we decided to keep multiple models to have a better overview. For decoder-only LLMs' fine-tuning, we used the parameterization techniques (PEFT) (Ding et al., 2023) with LoRA (Low-Rank Adaptation) (Hu et al., 2021) and bit quantizations to cope with memory issues and efficiency. Four-bit quantizations with LoRA *R=16 or 8* and *alpha =8*, *drop-out = 0.1*, *batch_sizes = 1, 2 or 4* with *gradient accumulation = 8*, learning rate *lr= 2e-4 or 5e-5* and lr scheduler, mostly cosine else linear were used. Parameter optimizations were done using the hyperparameter optimization framework, Optuna[6]. Further details of general experimental settings can be found in Appendix E. The prompts for instruction tuning each task are reported in Appendix D.

For the encoder-only models, we used full-finetuning (FFT), with 3 epochs of training, AdamW optimizer with learning rate of 2e-5, batch size of 8 or 16, and weight decay of 0.01.

In the multi-label setup of the DI task, we created a representative train-test sample dataset for the French dataset. This reduced the size of this automatically curated dataset (details in Appendix A) allowing us to avoid unnecessary computing costs. We used a custom trainer function to compute the multi-label loss using Binary Cross-Entropy with Logits (BCELoss with Logits).

4.2 Bias Metrics Measurements

We measure Tokenization Parity (TP) and Information Parity (IP) across six multilingual models: Phi-Mini-3.5, Gemma-7B, LLaMA-3.2 (3B), Mistral-7B, SILMA-9B, and Falcon-7B. Although SILMA is Arabic-focused, it builds on the multilingual Gemma architecture. The initial evaluation is conducted on 54 languages and dialectal variants from the FLORES-200 dataset—a parallel cor-

[6] https://optuna.org/

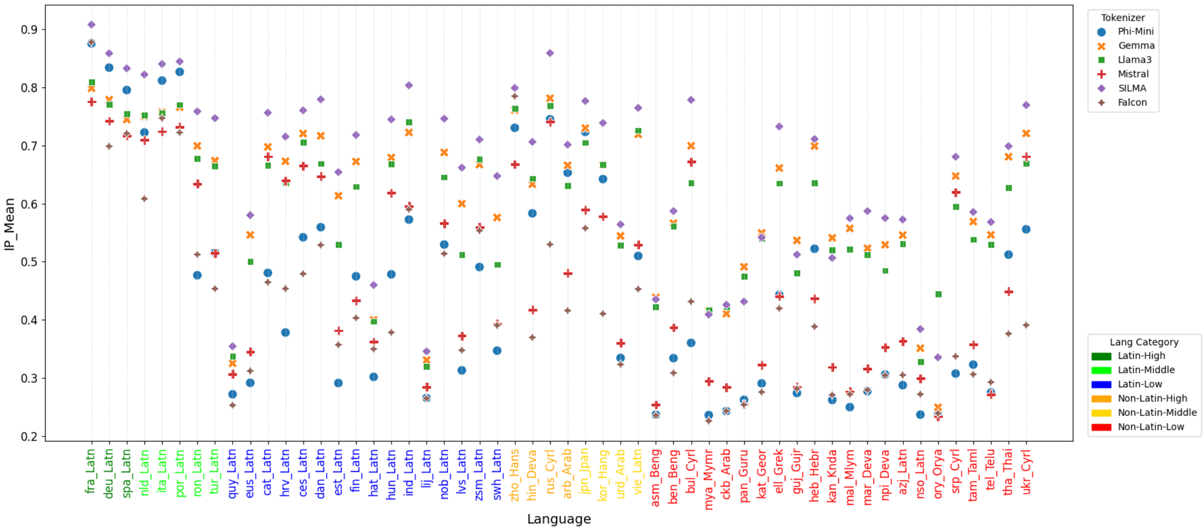


Figure 3: Information Parity (IP) across languages resulting from decoder-type models.

pus of 2,000 human-translated Wikipedia sentences across 200 languages (Costa-jussà et al., 2022). A subset of these score is then used for the correlation analysis on the dialect NLP tasks.

5 Results

Table 1 shows the F1-scores on the DI task. The encoder-type models outperform the heavily pre-trained decoder-type models across all datasets in both mono-label (ML) and multi-label (MuL) setups. Language-specific BERT models score better than mBERT in all cases except for the Swiss-German.[7].

Figure 4 shows the results for the TC and EQA tasks. Here we report F1-score averages for the script and resource level groups, the detailed tabular results per dialectal variety are presented in Tables 6 and 7 in Appendix F. On these tasks too, the encoder-type model, mBERT performs much better than the fine-tuned decoder multilingual LLMs. Regarding the controlled categories, it can be noted that the resource level affected the performance more than the script (the skewness of the polygons to the right), especially in decoder-type models. The differences between the model-types are smaller on the EQA task, as well as the impact of the resource level (except for Phi-3.5). Even though the impact of the script is smaller than that of the resource level, a bias towards Latin scripts is present, especially on the EQA task.

5.1 The distribution of the bias metrics values across languages

Figures 1 and 2 show the distribution of the TP score on the sample of 54 FLORES languages sorted (and colored) according to the controlled categories (resource level and script type). A comparison of these two graphs shows that encoder-type tokenizers result in a more stable TP than the tokenizers of the decoder-type models. However, a clear divide emerges in both model types: Latin-script languages maintain relatively stable TP and closer to 1 across all models, whereas non-Latin languages show substantial variability—particularly in lower-resource settings. Among decoder-type models, Gemma and SILMA demonstrate more consistent TP across language groups, while others show language-specific disparities.

When the TP values deviate more from 1, it shows larger disparities. For instance, with the mBERT tokenizer, the TP in German (Latin-High) is 1.26, while mBERT in Kannada (non-Latin-Low)is 2.19. This means the tokenizer produces 26% more tokens for German than for English, which is a good tokenizer premium, indicating that German is fairly close to English in efficiency, since it uses Latin script and shares vocabulary with English. In contrast, with Kannada, the tokenizer produces 119% more tokens than English for the same content, splitting the text into smaller fragments.

Figure 3 illustrates IP performance (this score applies only to decoder-type models). High-resource

[7] For curiosity, we tested also SILMA, the best performing Arabic decoder-type LLM on the NADI dataset. Although being an Arabic-specific model, it lags behind the mBERT model by almost 6 points and the Arabic-specific BERT model by about 28 points

Language (Type)	Decoder-only		Encoder-only	
	Phi-3.5	Llama 3.2	mBERT	Language-Specific
Arabic (ML)	0.54	0.26	0.62	**0.84** (MARBERT)
Swiss-German (ML)	0.49	0.46	0.59	**0.60** (SwissBERT)
Indo-Aryan (ML)	0.74	0.32	0.88	**0.90** (IndicTransformers)
French (MuL)	0.61	0.35	0.70	**0.75** (CamemBERT)
Spanish (MuL)	0.40	0.79	0.83	**0.85** (spanBERTa)

Table 1: Performance (F1-scores) on dialect identification task across models. **ML** = Mono-label version of the task, **MuL** = Multi-label version of the task.

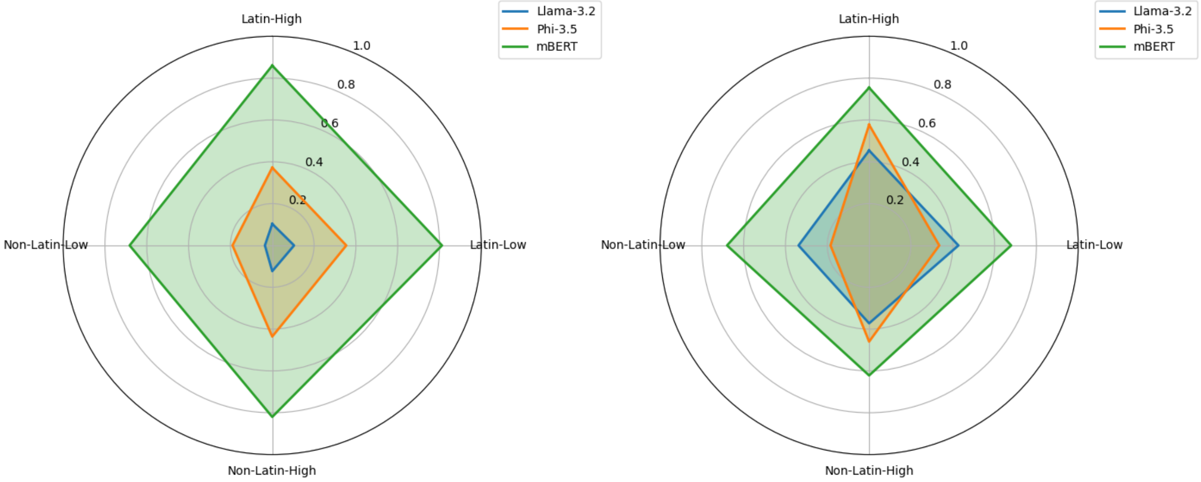


Figure 4: Average performance (F1-score) of models per category on TC (left) and EQA (right) tasks.

Latin-script languages generally exhibit higher IP, while non-Latin and low-resource languages display wider variation. Unlike TP, IP appears to be more dependent on resource levels.

5.2 Correlation analysis of TP & IP metrics with the downstream tasks

To examine whether trends in Tokenization Parity (TP) and Information Parity (IP) across languages correlate with model performance on dialectal downstream tasks, we compute Pearson correlation coefficients between downstream task scores and the TP/IP metrics, using fine-tuned versions of Phi-3.5-Mini, LLaMA-3.2, and mBERT. Note that the direction of the correlation score is important for a meaningful interpretation of the results.

In the case of TP, scores closer to 1 are considered better, while TP $>$ 1 indicates that the tokenizer uses more tokens to encode the same content compared to English (more fragmentation). Intuitively, we would expect a negative correlation between the value of TP and the downstream performance. High text fragmentation compared to the reference means that the input sequences are longer, which increases the complexity of the attention mechanism and makes modeling harder, which can impact the performance. In contrast, the expected correlation between IP and downstream performance is intuitively strongly positive, since a higher IP indicates greater representational efficiency, which should have a positive impact on the performance.

Figure 5 visualizes these correlations as heatmaps, with detailed tabular values provided in Appendix F, Table 8. To make the trends easier to follow, color codes show the expected correlations: blue for the expected, red for the opposite.

Dialect Identification (DI) Contrary to the expected direction, we see a positive correlation between TP and DI performance in the two models that perform better (mBERT and Phi-3.5 in the map Figure 5a, cf. Table 1), while the expected negative correlation is observed only in Llama-3.2, whose performance is low. On the other hand, higher IP, reflecting more efficient information compression, is correlated with worse performance on dialect classification in Phi-3.5 (map Figure 5b). This outcome is also contrary to what we expected. The fact that the correlation is positive in Llama-3.2 only confirms this observation because the low performance of Llama-3.2 indicates that the task was not learned, and the model might be performing some other classification. Note also that the correlations are stronger in models that perform the task better.

While we expected that higher tokenization disparity would lead to a performance drop, another picture appeared: it turns out that more fragmented text (compared to the reference), might, in fact, help models make surface-level distinctions if the task is learned at all. This could be attributed to the fact that dialects differ mostly at the surface level (spelling, morphology, and token patterns). If diacritics or other surface-level phenomena end up encoded as separate tokens due to higher text fragmentation, they might be exploited by models as useful dialect features even if the meaning of these units is not well captured in their vector representation. In other words, models do not need to “understand” the meaning of the small fragments to grasp their dialect specificity. In contrast, higher IP scores (expressing more equal compression) can be indicative of deeper level (semantic) similarity between the texts written in different dialects, making their differentiation harder even if the meaning is better captured. This would explain the surprising negative correlation between the IP score and the performance on the DI task.

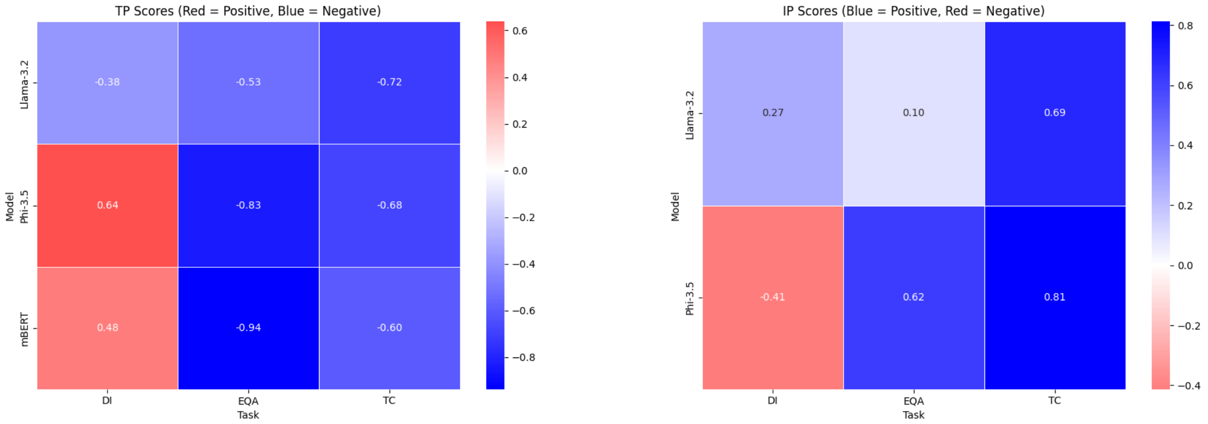
Topic Classification (TC) Comparing the two maps in Figure 5, we can see that IP is more strongly correlated with downstream performance than TP on this task, which applies more to the better model (Phi-3.5) than to the one with worse performance (Llama-3.2). This suggests that improved information compression across languages enhances performance on the TC task, but TP also shows a moderate correlation, indicating that tokenization may still impact the performance.

Extractive Question Answering (EQA) It is especially interesting to see in Figure 5a that the correlation between TP and the performance on this task is strong both in mBERT and Phi-3.5. This suggests that variation in tokenization can significantly impact the model’s ability to extract the correct span. There is also a moderate correlation with IP (the map Figure 5b), indicating that more consistent information representation across dialects may help the model extract relevant answers more effectively. In contrast, Llama-3.2 shows a moderate correlation with TP, but the correlation with IP is negligible. These findings suggest that tokenization disparities play a more significant role than general information preservation in extractive QA tasks, where accurate token-level span prediction is crucial.

Taken together, our results suggest that higher TP (more tokens per word for some languages) usually hurts the performance in both surface-level and semantically rich tasks as long as semantic representations are needed for the task. Higher IP scores (more similar compression), on the other hand, are associated with better downstream performance in both cases. The stronger association between the TP scores and the surface-level tasks, on one hand, and between the IP scores and the semantically rich tasks, on the other, is in line with the previous results reported by Tsvetkov and Kipnis (2024), where TP was better correlated with extractive or text similarity tasks (e.g., PAWSX, XQuAD). At the same time, IP correlated better with tasks requiring semantic consistency (e.g., reasoning), corresponding to our TC setting. The results on the DI task suggest an inverse relation between the TP and IP scores and the downstream task, which has not been reported in previous studies. In this case, higher TP is associated with better performance, while higher IP with worse downstream performance. As discussed above, the explanation for these effects comes from the fact that the task of distinguishing between dialect does not require deep semantic representation of surface-level features, while deeper semantic similarity (potentially captured by a high IP score) can even hurt the performance making the dialects harder to distinguish.

5.3 A closer look at the Llama tokenizer

As the Llama-3.2 model performance was behind all other models on all tasks, we take a closer look into its tokenizer and how it deals with non-Latin scripts. For this, we output the tokenization of a given input text as shown in the Arabic example in Figure 6. It can be observed that Llama-3.2 outputs misaligned tokens, which turned out to be characters misinterpreted as Latin-1, which is induced by the byte-level fallbacks. For instance, the token Ø³ has the following Unicode description: [’LATIN CAPITAL LETTER O WITH STROKE,’ ’SUPERSCRIPT THREE’]. The characters may break into smaller byte-level components if not directly present in the tokenizer vocabulary. These byte sequences may be aligned to Latin-character



(a) Correlations to TP across tasks and models

(b) Correlations to IP cross tasks and models

Figure 5: Correlation heatmaps showing Tokenization and Information Parity across dialectal tasks for different models; blue for the expected, red for the opposite of the expected.

English reference:
The find also grants insight into the evolution of feathers in birds
Arabic:
شيرلا روطت ىلع ةرظن اًضيأ فاشتكلاا حنمي
رويطلا يف

```
Llama3_Tokenizer_Output: ['ÙĬ', 'ÙħÙĨ',
'ØŃ', 'ĠØ§ÙĦ', 'Ø§Ùĥ', 'ØªØ´', 'Ø§Ùģ',
'ĠØ£ÙĬØ¶Ø§', 'Ùĭ', 'ĠÙĨØ¸', 'Ø±Ø©',
'ĠØ¹ÙĦÙī', 'ĠØª', 'Ø·ÙĪØ±', 'ĠØ§ÙĦ',
'Ø±ÙĬ', 'Ø´', 'ĠÙģÙĬ', 'ĠØ§ÙĦØ·',
'ÙĬ', 'ÙĪØ±']
```

Figure 6: Example of Llama-3.2 tokenizer output.

tokens due to a high bias toward Latin script in pretraining data. During decoding, the tokenizer may reassemble these tokens into the correct Unicode characters that match the non-Latin script. However, this can degrade the performance of non-Latin language tasks, as the model may not be able to capture the semantics and produce longer token sequences. Also, this script raises questions about how well the model captures the semantic meaning and linguistic nuances.

The same behavior was also noted in the non-Latin script language Hindi. For instance, रावन was tokenized as [’à¤°’, ’à¤¾à¤µà¤¨’], where the Hindi character र corresponds to the UTF-8 bytes [E0 A4 B0] [8]. This sequence is interpreted as [à ¤ °] in Latin-1, which is then represented as the token à¤° . Similar observations were made in all non-Latin scripts experimented with, where Latin characters were recognized. It should be noted, though (from Appendix B) that both Phi-3 and Llama-3 tokenizers are based on TikToken. This means that the tokenization behavior also largely depends on the tokenizer’s knowledge of the pretrained language.

As additional analyses, we examined the correlation between missing character proportions and downstream performance and investigated the language support specifications of the LLMs. Our findings suggest that these relationships remain highly model- and task-dependent. Detailed results and discussions are provided in Appendix G.

6 Related Work

The fairness and biases of LLM tokenizations have been analyzed using parallel language corpora by (Petrov et al., 2024; Ahia et al., 2023; Rust et al., 2021). Language-specific investigations (Toraman et al., 2023), temporal evaluations (Spathis and Kawsar, 2024), and adversarial impacts (Wang et al., 2024) and tokenizer comparisons (Kanjirangat et al., 2023; Batsuren et al., 2024) are other directions. The general conclusion pinpointed the importance of tokenization - *tokenization matters*. Following the limitations of tokenizers and other multilingual biases, another research dimension proposes alternative tokenization approaches (Hofmann et al., 2022) and even tokenless models (Barrault et al., 2024; Pagnoni et al., 2024). Extending the understanding and analysis of representational biases in multilingual LLMs, some potential works

[8] https://www.utf8-chartable.de/unicode-utf8-table.pl?start=2304&number=128

on metrics related to information theory perspectives (Tsvetkov and Kipnis, 2024) and (Land and Bartolo, 2024). The primary line of existing research in dialectal tasks focus on performance improvements across various datasets using LLMs (Scherrer et al., 2024; Alam et al., 2024; Frei and Schneider, 2023), with a recent focus on multi-label DI (Bernier-Colborne et al., 2023; Keleg and Magdy, 2023; Chifu et al., 2024; Kanjirangat et al., 2024). The primary research focused on assessing GPT-based models' multilingual capabilities, highlighting their limitations, with a few exceptions. GPT capability in Arabic was evaluated in (Khondaker et al., 2023), unveiling the shortcomings of dialectal Arabic and the supremacy of encoder models. In (Lai et al., 2023; Bang et al., 2023), ChatGPT was evaluated in diverse languages, showing the predominance of high-resource languages. Recently, a comprehensive dialectal benchmark dataset was introduced, DialectBench (Faisal et al., 2021), which encompasses various dialectal tasks covering a wide range of dialectal varieties. While there has been notable research in dialectal tasks and multilingual NLP individually, efforts to bridge the two remain limited. Existing work has largely focused on performance comparisons, with less attention to understanding the underlying causes of degraded performance.

7 Conclusion

In this paper, we go beyond traditional performance-based evaluations of dialectal downstream tasks to examine multilingual fairness and potential biases arising from disparities in tokenization and representation. We show that Tokenization Parity (TP) and Information Parity (IP) correlate with downstream task performance in a consistent, although sometimes surprising, way. Our results reveal consistent disparities in TP between Latin and non-Latin scripts, while IP variations are influenced by both script and resource availability. TP is more strongly associated with tasks involving syntactic, morphological, and span-based features, whereas IP aligns more closely with tasks requiring semantic understanding and reasoning. The role of token-level disparity is especially interesting in surface-level tasks such as dialect identification, which can help models make distinctions between dialects. As a future direction, we emphasize the importance of developing language-aware, adaptive tokenizers that can mitigate pre-training biases and flexibly operate across multiple levels of granularity.

8 Limitations

There is significant scope for further enhancing the LLMs examined in this work. The tokenization analysis can be improved by leveraging more extensive and diverse corpora, enabling more profound insights into tokenization strategies and their implications. While the primary focus here was to analyze the relationship between tokenization, language-specific factors, and their impact on language-dependent tasks, future work could explore additional use cases by identifying and incorporating relevant tasks. In this study, we concentrated on the dialectal tasks: first, as a challenging language-dependent task, it offers a robust testbed for examining tokenization impacts; and second, this aspect has largely been overlooked in prior research, where the emphasis has predominantly been on performance metrics. Expanding this investigation to include other complex language-dependent tasks could further elucidate the role of tokenization in multilingual LLM performance.

Acknowledgments

This work was supported by the project "fairTOK", funded by armasuisse S&T. The authors also gratefully acknowledge the reviewers for their insightful and constructive feedback, which helped improve the quality of this work.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Muhammad Abdul-Mageed, Abdelrahim Elmadany, Chiyu Zhang, Houda Bouamor, Nizar Habash, et al. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. In *Proceedings of ArabicNLP 2023*, pages 600–613.

David Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba Alabi, Yanke Mao, Haonan Gao, and En-Shiun Lee. 2024. Sib-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923.

Firoj Alam, Shammur Absar Chowdhury, Sabri Boughorbel, and Maram Hasanain. 2024. Llms for low resource languages in multilingual, multimodal and dialectal settings. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33.

Marcellus Amadeus, Jose Roberto Homeli da Silva, and Joao Victor Pessoa Rocha. 2024. Bridging the language gap: Integrating language variations into conversational AI agents for enhanced user engagement. In *Proceedings of the 1st Worskhop on Towards Ethical and Inclusive Conversational AI: Language Attitudes, Linguistic Diversity, and Language Rights (TEICAI 2024)*, pages 16–20, St Julians, Malta. Association for Computational Linguistics.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718.

Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R Costa-jussà, David Dale, et al. 2024. Large concept models: Language modeling in a sentence representation space. *arXiv e-prints*, pages arXiv–2412.

Khuyagbaatar Batsuren, Ekaterina Vylomova, Verna Dankers, Tsetsuukhei Delgerbaatar, Omri Uzan, Yuval Pinter, and Gábor Bella. 2024. Evaluating subword tokenization: Alien subword composition and oov generalization challenge. *arXiv preprint arXiv:2404.13292*.

Gabriel Bernier-Colborne, Cyril Goutte, and Serge Leger. 2023. Dialect and variant identification as a multi-label classification task: A proposal based on near-duplicate analysis. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 142–151.

Adrian-Gabriel Chifu, Goran Glavaš, Radu Tudor Ionescu, Nikola Ljubešić, Aleksandra Miletic Haddad, Filip Miletić, Yves Scherrer, and Ivan Vulić. 2024. Vardial evaluation campaign 2024: Commonsense reasoning in dialects and multi-label similar language identification. In *Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–15. The Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pretrained language models. *Nature Machine Intelligence*, 5(3):220–235.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fahim Faisal, Orevaoghene Ahia, Aarohi Srivastava, Kabir Ahuja, David Chiang, Yulia Tsvetkov, and Antonios Anastasopoulos. 2024. Dialectbench: An nlp benchmark for dialects, varieties, and closely-related languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14412–14454.

Fahim Faisal, Sharlina Keshava, Md Mahfuz Ibn Alam, and Antonios Anastasopoulos. 2021. Sd-qa: Spoken dialectal question answering for the real world. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3296–3315.

Claudio Frei and Philippe Schneider. 2023. Automatic identification of swiss german dialects using large language models.

Nizar Habash, Houda Bouamor, Ramy Eskander, Nadi Tomeh, Ibrahim Abu Farha, Ahmed Abdelali, Samia Touileb, Injy Hamed, Yaser Onaizan, Bashar Alhafni, et al. 2024. Proceedings of the second arabic natural language processing conference. In *Proceedings of The Second Arabic Natural Language Processing Conference*.

Valentin Hofmann, Hinrich Schuetze, and Janet B Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Vani Kanjirangat, Tanja Samardzic, Ljiljana Dolamic, and Fabio Rinaldi. 2023. Optimizing the size of subword vocabularies in dialect classification. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 14–30.

Vani Kanjirangat, Tanja Samardzic, Ljiljana Dolamic, and Fabio Rinaldi. 2024. Nlp_di at nadi 2024 shared task: Multi-label arabic dialect classifications with an unsupervised cross-encoder. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 742–747.

Anjali Kantharuban, Ivan Vulić, and Anna Korhonen. 2023. Quantifying the dialect gap and its correlates across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7226–7245, Singapore. Association for Computational Linguistics.

Amr Keleg and Walid Magdy. 2023. Arabic dialect identification under scrutiny: Limitations of single-label classification. *arXiv preprint arXiv:2310.13661*.

Md Tawkat Islam Khondaker, Abdul Waheed, Muhammad Abdul-Mageed, et al. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 220–247.

T Kudo. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Viet Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hiéu Mãn, Franck Dernoncourt, Trung Bui, and Thien Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11631–11646.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, et al. 2024. Byte latent transformer: Patches scale better than tokens. *arXiv preprint arXiv:2412.09871*.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2024. Language model tokenizers introduce unfairness between languages. *Advances in Neural Information Processing Systems*, 36.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. 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 3118–3135.

Tanja Samardžić and Nikola Ljubešić. 2021. *Data Collection and Representation for Similar Languages, Varieties and Dialects*, page 121–137. Studies in Natural Language Processing. Cambridge University Press.

Tanja Samardzic, Yves Scherrer, and Elvira Glaser. 2016. Archimob-a corpus of spoken swiss german. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4061–4066.

Yves Scherrer, Tommi Jauhiainen, Nikola Ljubešić, Preslav Nakov, Jörg Tiedemann, and Marcos Zampieri. 2023. Tenth workshop on nlp for similar languages, varieties and dialects (vardial 2023): Proceedings of the workshop. In *Workshop on NLP for Similar Languages, Varieties and Dialects*. The Association for Computational Linguistics.

Yves Scherrer, Tommi Jauhiainen, Nikola Ljubešić, Marcos Zampieri, Preslav Nakov, and Jörg Tiedemann. 2024. Proceedings of the eleventh workshop on nlp for similar languages, varieties, and dialects (vardial 2024). In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast wordpiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103.

Dimitris Spathis and Fahim Kawsar. 2024. The first step is the hardest: Pitfalls of representing and tokenizing temporal data for large language models. *Journal of the American Medical Informatics Association*, 31(9):2151–2158.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–21.

Alexander Tsvetkov and Alon Kipnis. 2024. Information parity: Measuring and predicting the multilingual capabilities of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7971–7989.

Daan van Esch, Tamar Lucassen, Sebastian Ruder, Isaac Caswell, and Clara Rivera. 2022. Writing system and speaker metadata for 2,800+ language varieties. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5035–5046.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9154–9160.

Dixuan Wang, Yanda Li, Junyuan Jiang, Zepeng Ding, Guochao Jiang, Jiaqing Liang, and Deqing Yang. 2024. Tokenization matters! degrading large language models through challenging their tokenization. *arXiv preprint arXiv:2405.17067*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, et al. 2018. Language identification and morphosyntactic tagging. the second vardial evaluation campaign.

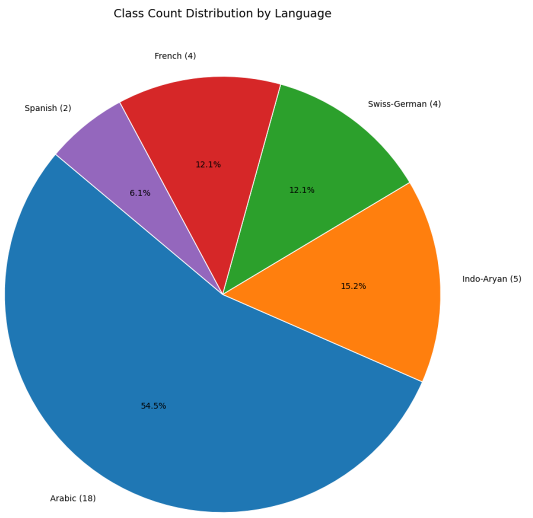


Figure 7: Class distributions of dialect classification task (Appendix A)

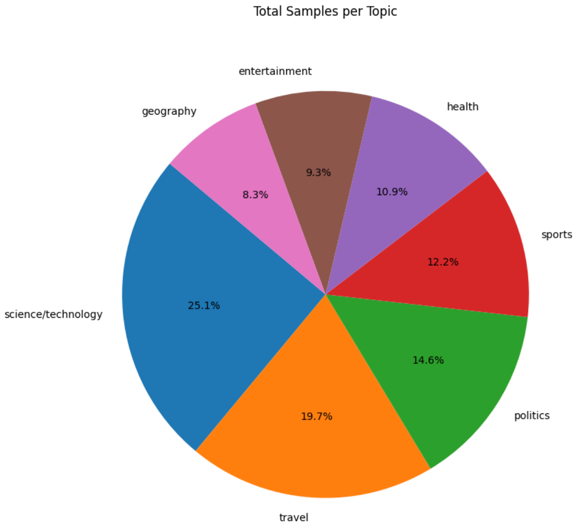


Figure 8: Class distributions topic classification task (Appendix A)

A Dataset Details

Table 2 shows the general statistics of DI task datasets. In the DI task, NADI-2023 has 18 dialects from Arabic-speaking regions such as Iraq, Oman, Saudi Arabia, Palestine, Bahrain, Egypt, Jordan, Libya, Sudan, UAE, Algeria, Kuwait, Tunisia, Lebanon, Morocco, Yemen, Syria, and Qatar. In GDI, we had four dialects: Zurich, Luzern, Basel, and Bern. For ILI, it was Hindi, Braj Bhasha, Awadhi, Bhojpuri, and Magahi. In multi-label settings, we used the datasets from the Multi-label classification of similar languages (DSL-ML) 2024 shared task, focusing on manually labeled Spanish and automatically labeled French data. For French, data was from the FreCDo dataset, including French (FR-FR), Swiss (FR-CH), Canadian (FR-CA), and Belgian (FR-BE) with {'FR-BE': 120653, 'FR-CH': 115664, 'FR-FR': 83127, 'FR-CA': 19041, 'FR-BE, FR-FR': 1052, 'FR-BE, FR-CH': 603, 'FR-CH, FR-FR': 162, 'FR-BE, FR-CH, FR-FR': 61} as multi-label samples. For Spanish, the two varieties were Argentinian and peninsular Spanish, with 1131 multi-label samples.

Under the multi-lingual setup, we created a representative train-test sample dataset for the French dataset due to the massive size of the automatically curated dataset. We selected 5000 mono-label samples from each class and all the multi-label samples comprising the train set. 1000 mono-label samples with all multi-label samples were selected for the test set. This constitutes 21878 (20000 (mono)+ 1878 (multi)) train and 4120 (4000+120) test samples.

The class distributions of the DI and TC tasks are shown in Figures 7 and 8.

B Tokenizer details

The details of the tokenizers and pre-trained vocabulary sizes used by the evaluated models are shown in Table 3.

C Script & resource Categorizations

The details of the languages under different scripts and resource categories are shown in Table 4.

D Prompt details

This section presents the instruction-tuning prompts used for the experiments in decoder-only LLMs. Figures 9, 10, and 11 represent the prompts for DI, TC, and EQA tasks, respectively.

E Experimental settings details

We evaluated the experiments on HPC clusters with a100 and v100 GPUs. The runtime varied between approximately. 4 hours - 1.5 days for decoder models and 1-2 hours for encoder models. All experimental models were accessed from the HuggingFace library.

F Result details

In this section, we present the detailed tabular results for the TC and EQA tasks per dialectal variety.

	GDI	ILI	NADI	DSL-ML-FR	DSL-ML-ES
Train	14647	68453	18000	340363	3467
Test	4752	9032	1800	17090	989
No. of labels	4	5	18	4	2

Table 2: Dataset statistics (Appendix A)

Models	Tokenizer	Model_Vocab_Sizes
Phi3	tiktoken (Abdin et al., 2024) [9]	32011
Gemma	SentencePiece tokenizer(Kudo, 2018; Team et al., 2024)	256000
Llama3	tiktoken (Dubey et al., 2024)	128256
Bloom	Byte-level BPE (Wang et al., 2020)	250680
Mistral	tekken- Modified tiktoken (Jiang et al., 2024)	32000
NLLB	SentencePiece tokenizer tailored	256204
BERT-based	WordPiece (Song et al., 2021; Devlin, 2018)	30522
MARBERT	WordPiece	100000
mBERT	WordPiece	119547
IndicBERT	WordPiece	200000
SpanBERTa	WordPiece	50265
CamemBERT	WordPiece	32005

Table 3: Tokenizers and vocabulary Sizes of LLMs (Appendix B)

```
TRAINING_CLASSIFIER_PROMPT = """[INST]What is the dialect of the given input
sentence.
Sentence:{sentence}
Class:{label}[/INST]"""
INFERENCE_CLASSIFIER_PROMPT = """[INST] Classify the dialect of the sentence.
Choose from one of the following options:{allowed_labels}.
Sentence:
{sentence}
[/INST]
Class:"""
```

Figure 9: Instruction-tuning prompt for dialect classification task (Appendix D)

Table 6 and 7 present the results on TC and EQA tasks, respectively. Table 8 presents the detailed correlation values of TP and IP over different downstream tasks across Phi-3.5, Llama-3.2 and mBERT models.

G Vocabulary analysis details

G.1 Language support details

Table 9 presents the language support details of the decoder-only LLMs. The information is based on the support claims of each model from their respective HuggingFace pages. When a model claims *"support"*, it may often refer to some representation of the language in its training data and the ability to generate or understand basic text in that language under ideal tokenization conditions. It may not guarantee robust handling of the language's words, characters, or scripts.

G.2 Missing character proportions

In this experiment, we compute the percentage of missing characters—those not represented as standalone tokens—in the vocabulary of each LLM. This analysis is limited to high-resource languages such as English, German, Spanish, French, Arabic, and Hindi across various LLMs. We aim to investigate potential correlations between character-level coverage and performance on dialectal downstream tasks. Although character-level analysis may initially seem counterintuitive given that most LLMs employ subword tokenizers, it sometimes becomes relevant due to their reliance on byte-level fallback mechanisms. Our qualitative analysis reveals that this fallback can sometimes negatively impact non-Latin scripts.

The Unicode ranges of the main character set used and the special characters are given in Table 5. As shown in Figure 12, all LLMs exhibit nearly complete character coverage for English,

Category	Languages (Flores Code)
Latin-High	Spanish (spa_Latn), German (deu_Latn), French (fra_Latn)
Latin-Middle	Dutch (nld_Latn), Italian (ita_Latn), Romanian (ron_Latn), Turkish (tur_Latn), Portuguese (por_Latn)
Latin-Low	Ayacucho Quechua (quy_Latn), Haitian Creole (hat_Latn), Basque (eus_Latn), Hungarian (hun_Latn), Catalan (cat_Latn), Danish (dan_Latn), Estonian (est_Latn), Indonesian (ind_Latn), Standard Latvian (lvs_Latn), Standard Malay (zsm_Latn), Finnish (fin_Latn), Swahili (swh_Latn), Norwegian Bokmål (nob_Latn), Croatian (hrv_Latn), Czech (ces_Latn), Ligurian (lij_Latn)
Non-Latin-High	Standard Arabic (arb_Arab), Russian (rus_Cyrl), Chinese (Simplified) (zho_Hans), Hindi (hin_Deva)
Non-Latin-Middle	Urdu (urd_Arab), Korean (kor_Hang), Vietnamese (vie_Latn), Japanese (jpn_Jpan)
Non-Latin-Low	North Azerbaijani (azj_Latn), Thai (tha_Thai), Marathi (mar_Deva), Odia (ory_Orya), Gujarati (guj_Gujr), Nepali (npi_Deva), Burmese (mya_Mymr), Assamese (asm_Beng), Central Kurdish (ckb_Arab), Tamil (tam_Taml), Malayalam (mal_Mlym), Bulgarian (bul_Cyrl), Eastern Panjabi (pan_Guru), Ukrainian (ukr_Cyrl), Bengali (ben_Beng), Kannada (kan_Knda), Greek (ell_Grek), Northern Sotho (nso_Latn), Serbian (srp_Cyrl), Telugu (tel_Telu), Hebrew (heb_Hebr), Georgian (kat_Geor)

Table 4: Language categories and corresponding languages with FLORES codes: Appendix C

```
TRAINING_CLASSIFIER_PROMPT = """[INST]What is the topic of the following text?
\nSentence:{sentence}\nClass:{label}[/INST]"""
INFERENCE_CLASSIFIER_PROMPT = """[INST] Classify the topic of the following
sentence.
Choose from one of the following options:{allowed_labels}.
Sentence:
{sentence}
[/INST]
Class:"""
```

Figure 10: Instruction-tuning prompt for topic classification task (Appendix D)

```
TRAINING_CLASSIFIER_PROMPT = """[INST]Extract the answer of the question from the
given context
\nQuestion:{sentence}\nContext:{context}\nAnswer:{label}[/INST]"""
INFERENCE_CLASSIFIER_PROMPT  = """[INST] Answer the question based on
given context. Output from the given context only as in extractive QA.
Question:
{sentence}
Context:{context}
[/INST]
Answer: """
```

Figure 11: Instruction-tuning prompt for EQA task (Appendix D)

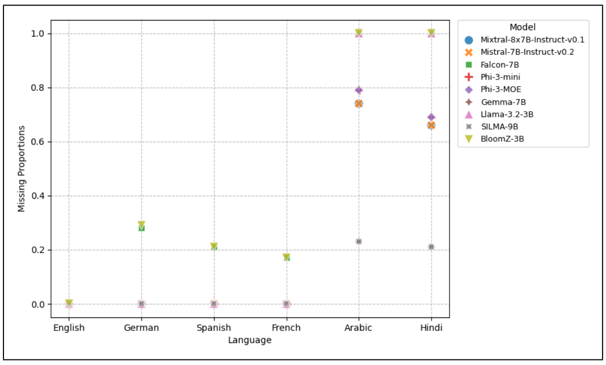


Figure 12: Missing proportions of language characters in decoder-only tokenizer vocabulary (Appendix G.2)

with a missing proportion of 0.0 (lower is better). In contrast, the missing proportion for non-Latin languages is considerably higher than for Latin languages across most multilingual decoder-based models (e.g., LLaMA, Phi, Mixtral), with the exception of the Gemma model. Among language-specific decoder models, the missing proportion is notably lower—for instance, SILMA reports only 23% missing characters in Arabic.

For encoder-only models (see Figure 13), language-specific encoders tend to achieve better character coverage in their respective languages.

Language	Unicode ranges & special characters
Hindi	(0x0900, 0x097F + 1)
Arabic	(0x0600, 0x06FF + 1)
English	(0x41, 0x5B) and (0x61, 0x7B)
Spanish	[’á’, ’é’, ’í’, ’ó’, ’ú’, ’ü’, ’ñ’, ’Á’, ’É’, ’Í’, ’Ó’, ’Ú’, ’Ü’, ’Ñ’]
French	[’à’, ’â’, ’ä’, ’ç’, ’é’, ’è’, ’ê’, ’ë’, ’î’,’ï’, ’ô’, ’ö’, ’ù’, ’û’, ’ü’,’À’, ’Â’, ’Ä’, ’Ç’, ’É’, ’È’, ’Ê’, ’Ë’, ’Î’, ’Ï’, ’Ô’, ’Ö’, ’Ù’, ’Û’, ’Ü’]
Swiss-German	[’ä’, ’ö’, ’ü’, ’Ä’, ’Ö’, ’Ü’, ’ß’]

Table 5: Character Unicode ranges and special characters for different languages - all special characters are contained in the Latin-1 supplement Unicode block 0x0080-0x00FF. (Appendix G.2)

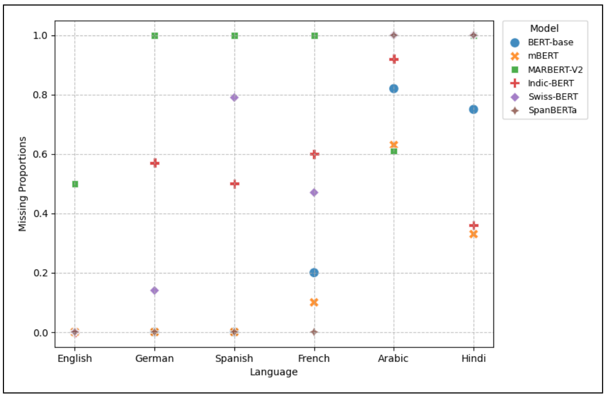


Figure 13: Missing proportions of language characters in encoder-only tokenizer vocabulary (Appendix G.2)

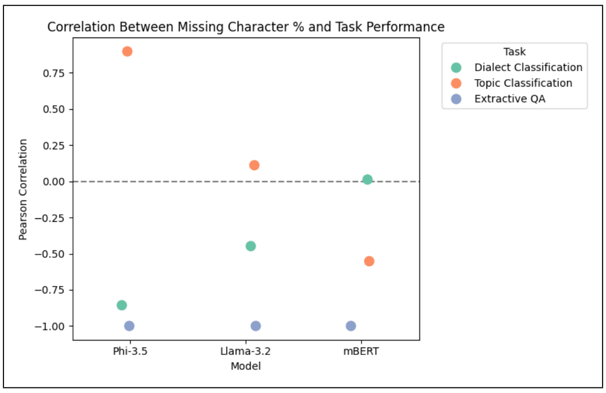


Figure 14: Correlation of missing character proportions to dialectal tasks

Notably, mBERT maintains reasonable coverage over non-Latin scripts. However, MARBERT exhibits a substantial proportion of missing Arabic characters at the token level. This is likely due to its frequency-based subword tokenizer, where individual characters are often absorbed into larger subword units.

While such missing character coverage does not necessarily impair performance in language-specific models—owing to their strong modeling of linguistic structure across granularities—it can pose challenges for broader multilingual LLMs. Ensuring at least character-level granularity in these models may help mitigate issues arising from multibyte representations of non-Latin scripts.

We used the three models for correlation analysis - mBERT, Phi-3.5, and Llama-3.2. From Figure 14, it can be observed that negative correlations dominate, especially for Phi-3.5 and Llama-3.2. All models show high negative correlations with EQA task, indicating that higher character coverage (fewer missing characters) improves performance. In TC, both decoder-only LLMs show positive correlations.

Category	Language	LLaMA-3.2	Phi-3.5	mBERT
Latin-High	Dutch (nld_Latn)	0.1096	0.4178	0.894
	English (eng_Latn)	0.0494	0.3210	0.897
	French (fra_Latn)	0.1436	0.4274	0.910
	German (deu_Latn)	0.1187	0.3692	0.862
	Italian (ita_Latn)	0.1137	0.3309	0.872
	Portuguese (por_Latn)	0.0823	0.3523	0.868
	Spanish (spa_Latn)	0.1313	0.3813	0.821
	Romanian (ron_Latn)	0.0791	0.3762	0.857
Latin-Low	Catalan (cat_Latn)	0.1010	0.3502	0.858
	Croatian (hrv_Latn)	0.0972	0.4823	0.858
	Estonian (est_Latn)	0.1686	0.4508	0.766
	Finnish (fin_Latn)	0.1241	0.3067	0.809
	Haitian Creole (hat_Latn)	0.0700	0.1946	0.61
	Hungarian (hun_Latn)	0.1206	0.3671	0.861
	Indonesian (ind_Latn)	0.0937	0.3666	0.847
	Norwegian Bokmål (nob_Latn)	0.1024	0.3592	0.862
	Basque (eus_Latn)	0.0630	0.2946	0.82
Non-Latin-High	Arabic (arb_Arab)	0.1491	0.5194	0.811
	Hebrew (heb_Hebr)	0.1507	0.3685	0.83
	Hindi (hin_Deva)	0.0888	0.4972	0.742
	Japanese (jpn_Jpan)	0.0704	0.4043	0.888
	Russian (rus_Cyrl)	0.1565	0.3897	0.827
Non-Latin-Low	Bengali (ben_Beng)	0.0288	0.2192	0.773
	Gujarati (guj_Gujr)	0.0000	0.0824	0.597
	Kannada (kan_Knda)	0.0287	0.0952	0.78
	Malayalam (mal_Mlym)	0.0089	0.0938	0.66
	Marathi (mar_Deva)	0.1011	0.3732	0.744
	Nepali (npi_Deva)	0.0503	0.3610	0.762
	Orya (ory_Orya)	0.0281	0.1056	0.461

Table 6: Macro F1 scores for languages under different resource-script categories in the topic classification task. (Appendix F)

Category	Language Code	Llama3	Phi-3.5	mBERT
Latin-High	english-kenya	0.431337	0.540717	0.725
	english-nzl	0.462533	0.596464	0.767
	english-irl	0.462224	0.600408	0.755
	english-ind_n	0.443038	0.573698	0.746
	english-phl	0.455140	0.588592	0.764
	english-nga	0.452876	0.566533	0.736
	english-aus	0.458323	0.593847	0.757
	english-ind_s	0.431355	0.543300	0.719
	english-usa	0.479512	0.608536	0.772
	english-gbr	0.471987	0.596156	0.764
	english-zaf	0.464280	0.594952	0.766
Latin-Low	swahili-kenya	0.443695	0.350143	0.724
	swahili-tanzania	0.410629	0.320931	0.635
Non-Latin-High	arabic-sau	0.361613	0.457623	0.778
	arabic-mar	0.361082	0.445782	0.767
	arabic-jor	0.360686	0.45141	0.773
	arabic-tun	0.358351	0.448329	0.767
	arabic-bhr	0.359525	0.456000	0.775
	arabic-dza	0.357209	0.455835	0.778
	arabic-egy	0.345871	0.441766	0.765
	korean-korn	0.432525	0.481986	0.10
	korean-kors	0.414520	0.500209	0.092
Non-Latin-Low	bengali-ind	0.325579	0.176780	0.686
	bengali-dhaka	0.349494	0.192330	0.673

Table 7: F1 and EM scores by language code and category for EQA task -Appendix F

Task	Model / Category	Tokenization Parity	Information Parity
Dialect Classification (DI)	Phi-3.5	0.638	-0.413
	Llama-3.2	-0.380	0.268
	mBERT	0.4836	—
Topic Classification (TC)	Phi-3.5	-0.683	0.812
	Latin-High	0.873	0.765
	Latin-Low	-0.785	0.165
	Non-Latin-High	0.202	-0.862
	Non-Latin-Low	0.876	0.634
	Llama-3.2	-0.716	0.687
	Latin-High	0.974	0.671
	Latin-Low	0.077	0.328
	Non-Latin-High	-0.547	0.209
	Non-Latin-Low	-0.805	0.623
	mBERT	-0.605	—
	Latin-High	-0.242	—
	Latin-Low	-0.706	—
	Non-Latin-High	-0.826	—
	Non-Latin-Low	-0.828	—
Dialectal Extractive QA (EQA)	Phi-3.5	-0.834	0.618
	Llama-3.2	-0.528	0.097
	mBERT	-0.938	—

Table 8: Correlation values (overall and per category) between model tokenization/information parity and dialectal task performance-Appendix F

	English	**Arabic**	**German**	**Spanish**	**French**	**Hindi**
			Encoder-only			
BERT-base	✓	✗	✗	✗	✗	✗
BERT-base-uncased	✓	✗	✗	✗	✗	✗
mBERT	✓	✓	✗	✓	✓	✓
MARBERT-V2	✗	✓	✗	✗	✗	✗
Indic-Transformers	✓	✗	✗	✗	✗	✓
Swiss-BERT	✗	✗	✓	✗	✓	✗
SpanBERTa	✗	✗	✗	✓	✗	✗
CamemBERT	✗	✗	✗	✗	✓	✗
			Decoder-only			
Mixtral-8x7B-Instruct-v0.1	✓	✗	✗	✓	✓	✗
Mistral-7B-Instruct-v0.2	✓	✗	✗	✗	✗	✗
Falcon-7B	✓	✗	✗	✗	✗	✗
phi3-mini	✓	✓	✗	✓	✓	✗
phi3-MOE	✓	✓	✗	✓	✓	✗
Gemma-7B	✓	✓	✗	✓	✓	✓
Llama3.2-3B	✓	✗	✗	✓	✓	✓
SILMA-9B	✓	✓	✗	✗	✗	✗

Table 9: Language support details of LLMs (Appendix G.1)