Data-Augmentation-Based Dialectal Adaptation for LLM

Fahim Faisal, Antonios Anastasopoulos
Department of Computer Science, George Mason University
{ffaisal,antonis}@gmu.edu

Abstract

This report presents GMUNLP's participation to the Dialect-Copa shared task at VarDial 2024 (Chifu et al., 2024), which focuses on evaluating the commonsense reasoning capabilities of large language models (LLMs) on South Slavic micro-dialects. The task aims to assess how well LLMs can handle non-standard dialectal varieties, as their performance on standard languages is already well-established. We propose an approach that combines the strengths of different types of language models and leverages data augmentation techniques to improve task performance on three South Slavic dialects: Chakavian, Cerkno, and Torlak. We conduct experiments using a language-family-focused encoder-based model (BERTić) and a domain-agnostic multilingual model (AYA-101). Our results demonstrate that the proposed data augmentation techniques lead to substantial performance gains across all three test datasets in the open-source model category. This work highlights the practical utility of data augmentation and the potential of LLMs in handling non-standard dialectal varieties, contributing to the broader goal of advancing natural language understanding in low-resource and dialectal settings.[1]

1 Introduction

Recent advancements in large language models (LLMs) have led to remarkable performance on a wide range of natural language understanding tasks, particularly in standard languages. However, the effectiveness of these models on non-standard dialectal varieties remains an open question (Faisal et al., 2024). The Dialect-Copa shared task, introduced by Ljubešić et al. (2024), aims to bridge this gap by evaluating the commonsense reasoning capabilities of LLMs on South Slavic dialects.

Commonsense reasoning, as originally proposed by Gordon et al. (2011), requires models to make plausible inferences based on everyday knowledge and understanding of the world. Extending this task to dialects poses unique challenges, as models must capture the nuances and variations specific to these language varieties. The Dialect-Copa shared task provides a platform to explore the adaptability and generalization capabilities of LLMs in this context.

In this GMUNLP submission, we explore the potential of data augmentation techniques in enhancing the performance of language models on dialectal commonsense reasoning tasks. Our approach harnesses the power of state-of-the-art LLMs to generate synthetic training data, which we combine with the provided training dataset. By employing a diverse set of language models, we aim to quantify the performance gains achievable through data augmentation. Specifically, we utilize three categories of language models to maximize dialectal task performance: (1) smaller language models that are well-suited for low-resource settings and can be easily customized, (2) mid-size language models that strike a balance between task-specific performance and language understanding capabilities, and (3) closed-source language models that generate high-quality synthetic task data to further enhance the performance of the other two categories of language models.

We achieved the highest scores across all three test datasets in the open-source model category. In addition, our solution performed on par with the GPT-4 zero-shot iterative prompting approach employed by one of the teams, demonstrating the competitiveness of the proposed approach against state-of-the-art closed-source models. Furthermore, we achieved substantial performance improvements for the small-scale, language-family-focused model BERTić by combining it with our data augmentation strategy, showcasing the effectiveness of our approach in boosting the performance of language models tailored for low-resource settings.

The remainder of this paper is organized as

---

[1]Code and data are publicly available: https://github.com/ffaisal93/dialect_copa

follows: Section 2 provides an overview of the Dialect-Copa shared task and dataset, Section 3 describes our methodology and experimental setup, Section 4 presents our results and analysis, and Section 5 concludes the paper and discusses future directions.

2 The Dialect-Copa shared task

Task Information   In the Dialect-Copa shared task, a premise sentence is provided along with a question that can be either a cause or an effect. The objective is to build a classifier that selects the most plausible response from two candidate answer choices based on the given premise and question. To illustrate, consider the following training example in English, where the task is to identify the most plausible cause:

```
{"premise": "My body cast a shadow over the grass.",
 "choice1": "The sun was rising.",
 "choice2": "The grass was cut.",
 "question": "cause", "label": 0, "idx": 0}
```

The Dialect-Copa dataset consists of such cause-effect examples across 8 languages and dialects, challenging models to perform commonsense reasoning in non-standard language varieties.

code	language	train	val.	test
en	English	400	100	
sl	Slovenian	400	100	
sl-cer	Cerkno	400	100	500
hr	Croatian	400	100	
hr-ckm	Chakavian	-	-	500
sr	Serbian	400	100	
sr-trans	Serbian (transliterated)	400	100	
sr-tor	Torlak	400	100	
sr-tor-trans	Torlak (transliterated)	400	100	500
mk	Macedonian	400	100	
mk-trans	Macedonian (transliterated)	400	100	

Table 1: Dialect-Copa dataset statistics for different languages and their dialectal varieties.

Languages   The Dialect-Copa dataset encompasses training and validation data in 7 languages, including English, 6 moderately resourced South Slavic languages, and two related micro-dialects. The test dataset features these two micro-dialects along with an additional previously unseen dialect. The three dialects in the test set are as follows:

1. The Cerkno dialect of Slovenian, spoken in the Slovenian Littoral region, specifically in the town of Idrija.

Base model	Fine-tuning (FT)/Prompting	Epoch	Acc. (%)	
			en	hr
Aya-101	4 shot (2 cause, 2 effect)	-	**80**	**75**
MaLA-500	4 shot (2 cause, 2 effect)	-	50	–
Llama2-CHAT (7B)	4 shot (2 cause, 2 effect)	-	75	50
BERT	FT (eval. lang)	3	**66**	55
mBERT	FT (eval. lang)	3	55	57
XLM-R	FT (eval. lang)	3	54	54
BERTić	FT (eval. lang)	3	48	**64**

Table 2: Preliminary evaluation results on the English and Croatian validation set for different base models.

2. The Chakavian dialect of Croatian from the northern Adriatic, particularly from the town of Žminj.

3. The Torlak dialect, spoken in southeastern Serbia, northeastern North Macedonia, and northwestern Bulgaria, with the specific test instances coming from the town of Lebane.

Cerkno and Torlak dialects are present in all three dataset splits (training, validation, and test) whereas, the Chakavian dialect is intentionally held out from the training and validation splits and is exclusively encountered during the test phase. Each dialect in the test dataset comprises 500 instances. Table 1 presents the detailed statistics of the Dialect-Copa dataset, providing an overview of the distribution of instances across languages and dialects.

3 Experimental Phases

In this section, we report different phases of our experiments. We step by step perform experiments to choose appropriate base models followed by data augmentation, combination and task-specific model tuning.

Phase 1: Model Selection   In the preliminary phase of our experiments, we conduct a series of trials to identify base language models that demonstrate strong performance on language understanding tasks in a multilingual context. To achieve this, we fine-tune widely-used encoder-based models, such as BERT (Devlin et al., 2019), mBERT, and XLM-R (Conneau et al., 2020), on the English and Croatian subsets of the Dialect-Copa training dataset. Additionally, we explore the potential of more recently open-sourced large language models (LLMs) of varying sizes, such as LLaMA-2 (Touvron et al., 2023), Aya-101 (Üstün et al., 2024) and MaLA-500 (Lin et al., 2024), to gauge their effectiveness on the task.

Data Identifier	Description	Covered Language
`[lang]-train`	Original Dialect-Copa training data	en, hr, mk, sl, sl-cer, sr, sr-tor
`[lang]-trans`	Transliterated (Cyrillic → Latin) training data	mk, sr, sr-tor
`[lang]-claude`	Providing grammar rules and few-shot Croatian-Chakavian examples to generate synthetic parallel hr-ckm-train examples given the hr-train examples	hr-ckm
`[lang]-gpt4`	Additional synthetic English training data generated by GPT-4 (Whitehouse et al., 2023)	en
`[lang]-reverse`	Reverse-augmentation on `[lang]-train`, `[lang]-trans`, and `[lang]-claude` data	en, hr, mk, sl, sl-cer, sr, sr-tor
`[lang]-nllb`	Machine translation of `en-gpt4` source data to other languages using the NLLB-6B model	hr, mk, sl, sr

Table 3: Training data augmentation approaches

Our key observations from this preliminary phase are as follows:

→ BERT, mBERT, and XLM-R exhibit comparable performance on the Croatian subset, achieving an accuracy of around 55%(+/-) after 3 epochs of in-language fine-tuning. However, the monolingual English BERT model surpasses the multilingual models on the English subset when fine-tuned for the same number of epochs.

→ BERTić (Ljubešić and Lauc, 2021), a transformer-based model pre-trained on Bosnian, Croatian, Montenegrin, and Serbian languages, aligns well with the target languages of the Dialect-Copa test set. Fine-tuning BERTić on the Croatian subset yields a notable performance improvement of approximately 12 percent (i.e. 7 percentage points) compared to the aforementioned multilingual models.

→ Employing 4-shot prompting with the LLaMA-2 7B parameter model results in better performance on the English subset. However, for the Croatian subset, LLaMA-2 generates random inferences. This finding aligns with expectations, as LLaMA-2 is primarily an English-centric model and not inherently multilingual. In an effort to address the multilingual limitations of LLaMA-2, Lin et al. (2024) proposed MaLA-500, a multilingual adaptation of the model that underwent fine-tuning using a causal language modeling objective. However, after this adaptation, MaLA-500 produces random-level inferences on the English subset.

→ Aya-101, a 13B parameter mt5-xxl-based model (Xue et al., 2021) instruction-tuned in 101 languages. It shows superior performance both in English and Croatian.

Based on these preliminary findings, we select the two best-performing models, Aya-101 and BERTić , for further experimentation in the subsequent phases of our study. We report our preliminary experimental findings in Table 2. The results of our preliminary experiments are summarized in Table 2.

Phase 2: Data Augmentation To address the limited size of the Dialect-Copa training dataset, which consists of only 400 instances per language, we employ various data augmentation techniques to expand the available training data. This step is crucial in mitigating the data scarcity bottleneck and improving the models' ability to generalize across diverse dialectal variations. By augmenting the training data, we aim to provide a more representative dataset for task-specific fine-tuning and instruction tuning of our selected language models. The data augmentation approaches we explore include:

→ The test dataset primarily contains instances written using the Latin script. Hence, we transliterate the Macedonian (mk) dataset from Cyrillic to Latin script to maintain consistency with the already available Serbian, and Torlak transliterated datasets.

→ For each instance in the training data, we swap the premise and the correct answer choice, effectively transforming cause examples into effect examples and vice versa, thereby doubling the number of training instances. For example consider the following premise and two 'effect' choices:

> premise: I poured water on my sleeping friend.
> choice1: My friend awoke. ✓
> choice2: My friend snored. ✗

Now our proposed reverse-augmentation

Setting	Description	Data Combination
o	All **o**riginal dialect-copa training data mixed together	[en, hr, mk, sl, sl-cer, sr, sr-tor]-train [sr, sr-tor]-trans
otrsl	Combining all **o**riginal, **t**ransliterated as well as **r**everse-augmented and **s**ynthetic training data (only **l**atin script ones)	[en, hr, sl, sl-cer]-train [sr, sr-tor, mk]-trans [en, hr, mk-trans, sl, sl-cer, sr-trans, sr-tor-trans]-reverse en-gpt4, hr-ckm-claude [hr, sl, mk-trans, sr-trans]-nllb
otrslc	Combining all original, transliterated as well as reverse-augmented and synthetic training data (Both latin and cyrillic script)	all available training data
otrsl$_{mk-hr-ckm}$	Selective otrsl setting with upsampled data count by repetition for mk, hr and hr-ckm	hr-train, mk-trans, hr-ckm-claude [hr-train, mk-trans, hr-ckm-claude]-reverse [hr, mk-trans]-nllb
otrsl$_{hr-ckm}$	Selective otrsl setting with upsampled data count by repetition for hr and hr-ckm	hr-train, hr-ckm-claude [hr-train, hr-ckm-claude]-reverse hr-nllb
otrsl$_{sl-cer}$	Same as previous but for sl and sl-cer	[sl, sl-cer]-train [sl, sl-cer]-reverse sl-nllb
otrsl$_{sr-tor}$	Same as previous but for sr and sr-tor	[sr, sr-tor]-trans [sr-trans, sr-tor-trans]-reverse sr-nllb-trans
otrslc$_{sr-tor}$	Same as previous but we include both transliterated as well as Cyrillic script data	[sr, sr-tor]-train, [sr, sr-tor]-trans [sr, sr-trans, sr-tor, sr-tor-trans]-reverse sr-nllb, sr-nllb-trans
otrsl$_{mix}$	Cross-lingual mix and match using all data from otrsl setting	[en, hr, sl, sl-cer]-train [sr, sr-tor, mk]-trans [en, hr, mk-trans, sl, sl-cer, sr-trans, sr-tor-trans]-reverse en-gpt4, hr-ckm-claude [hr, sl, mk-trans, sr-trans]-nllb
otrsl$_{mix-mk-hr-ckm}$	Cross-lingual mix and match using all data from otrsl$_{mk-hr-ckm}$ setting	hr-train, mk-trans, hr-ckm-claude [hr-train, mk-trans, hr-ckm-claude]-reverse [hr, mk-trans]-nllb
otrsl$_{mix-hr-ckm}$	Cross-lingual mix and match using all data from otrsl$_{hr-ckm}$ setting	hr-train, hr-ckm-claude [hr-train, hr-ckm-claude]-reverse hr-nllb
otrslc$_{mix-testset}$	Cross-lingual mix and match using all data from otrslc setting except English	all available training data except English

Table 4: After performing data augmentation, we create various data combinations by merging the augmented data blocks described in Table 3 with the original training datasets. These carefully designed data settings are then employed to conduct task-specific fine-tuning or instruction tuning on the selected base models, enabling us to evaluate the impact of different data configurations and therefore, select the suitable ones for the test-set evaluation phase.

method will transform the above example in a 'casue-specific' question as follows:

> premise: My friend awoke.
> choice1: I poured water on my sleeping friend.✓
> choice2: My friend snored. ✗

→ We utilize a publicly available English COPA-style synthetic dataset generated by GPT-4 (Achiam et al., 2023), as introduced by White-house et al. (2023). To expand the coverage of this synthetic data to other languages, we translate the English examples using the NLLB-6B machine translation model (Team et al., 2022) to all the four Dialect-Copa standard languages: Croatian, Macedonian, Serbian and Slovenian.

→ The Dialect-Copa dataset does not provide any training or validation data for the Chakavian dialect. To overcome this limitation, we compile a set of Croatian to Chakavian conversion rules and corresponding examples from online language community forums (uni). In addition to these rules, we also gather a few Croatian to Chakavian lyrics translations (lyr). We then prompt the Claude-3 language model (ant) with these rules and examples, instructing it to translate the Croatian sentences from the Dialect-Copa training set into their Chakavian equivalents. Through this process, we create a synthetic Chakavian training set in the style of Dialect-Copa, which we refer to as [lang]-claude. Here is an example with ground truth Croatian to Chakavian translation (words identically translated to the available gold translations are bolded):

> → Croatian (source): Djevojka je pronašla kukca u žitaricama. Izgubila je apetit.
>
> → Chakavian (gold-translation): Mlada je **našla** neko blago va žitaricah. Je **zgubila tiek**.
>
> → Chakavian (claude-translation): Divojka je **našla** buba u žitarican. **Zgubila** je **tiek**.

We observe that only a small number of words, specifically three in this instance, are correctly translated from Croatian to Chakavian. Despite the limited accuracy of the translation, this synthetic translated dataset enables us to train and evaluate models on the Chakavian dialect, despite the absence of original training data for this specific dialect. The detailed report on the dialect conversion rules and the Claude-3 prompt template used for generating the synthetic Chakavian dataset can be found in Appendix A.

Table 3 provides a comprehensive overview of the data augmentation techniques employed and the languages covered by each approach.

Phase 3: Data Selection Following the data augmentation process, we create various data combinations by merging the augmented data with the original training datasets. Table 4 provides a comprehensive overview of the various training data combination settings we employ, along with their respective descriptions and the specific data sources included in each combination. These combinations are designed to investigate the impact of different data characteristics on the performance of our models. For instance, the otrsl setting combines all original, transliterated, reverse-augmented, and synthetic data while excluding any data written in the Cyrillic script. The rationale behind this combination is to assess whether our Latin-only Dialect-Copa test set benefits from the absence of script variations in the training data. Additionally, we introduce a language-agnostic data combination denoted as otrsl_{mix}, in which we perform cross-lingual modifications by ensuring that the premise, choice1, and choice2 for each example are presented in different languages. This combination allows us to evaluate the models' ability to handle language-agnostic reasoning.

Phase 4: Prompt Design Encoder-based models can be fine-tuned using any of the data settings created in the previous steps. However, to perform few-shot prompting or instruction tuning with generative language models (LLMs), we need to design prompt-based instructions. During our preliminary experiments, we observed that using 4-shot same-class prompting (i.e., providing 4 cause examples for a cause-based question) yields slightly better results compared to combining 2 cause and 2 effect examples in the prompt. Specifically, this approach led to a 4.9% improvement on the English validation set. So we opted for 4-shot same-class prompting to perform inference. Here, these 4-shot same-class examples are randomly drawn from the training set of the target dialect. When the training set is unavailable for a specific dialect, a closely related language is used (eg. Croatian training set for Chakavian examples).

The following prompt template is used for inference and instruction tuning of the Aya-101

model:

> Instruction: Given the premise, {premise}, What is the correct {question} {'before'/'after'} this?
> A: {choice1}
> B: {choice2}
> Correct {question}: {correct_answer}

By designing the prompt in this manner, we provide the model with a clear instruction, the premise, and the two answer choices. The model is then expected to select the correct answer based on the given question type (cause or effect). This template is employed both during inference and instruction tuning of the Aya-101 model to ensure consistency and optimize performance on the Dialect-Copa dataset.

Phase 5: Task-Specific Tuning We employ two distinct approaches for task-specific tuning of our selected models. The first approach, known as full model fine-tuning, involves updating all the weights of the model during the training process. We apply this method to the BERTić model, fine-tuning it for 5-10 epochs on the Dialect-Copa dataset. However, for mid-size models like Aya-101, full fine-tuning may be unnecessarily computationally expensive, especially considering the limited amount of training data available. To address this concern, we use LoRA (Low-Rank Adaptation) adapter tuning (Hu et al., 2022) which is a more parameter-efficient tuning approach. LoRA introduces a small number of trainable parameters in the form of low-rank matrices, which are inserted between the layers of the pre-trained model. Note that this draws from a long history of efficient adaption using dedicated units (Houlsby et al., 2019; Pfeiffer et al., 2020; Faisal and Anastasopoulos, 2022). During training, only these newly introduced parameters are updated, while the original model weights remain frozen. This approach significantly reduces the number of trainable parameters, making it more suitable for fine-tuning on smaller datasets. By employing LoRA adapter tuning, we can effectively adapt the Aya-101 model to the Dialect-Copa dataset without the need for full model fine-tuning, thereby striking a balance between performance and computational efficiency.

4 Results and Discussion

In this section, we present and discuss the results of our experiments on the Dialect-Copa dataset.

4.1 Validation Set Insights

Table 5 summarizes the key takeaways from our incremental experiments conducted on the validation dataset using BERTić and Aya-101. First, we observe that combining datasets from multiple languages boosts performance on the Croatian subset, as opposed to training on a single language. This finding motivates us to utilize all available training data and prepare different data combinations, as described in Table 4. Second, we find that increasing the data quantity through various data augmentation techniques (Table 3) primarily improves performance for most languages and low-resource dialects. Furthermore, discarding instances written in the Cyrillic script can boost performance for certain languages and dialects (e.g., Croatian, Cerkno, and Serbian), while hurting others.

We also explore cross-lingual mix-and-match strategies, but we do not find any conclusive patterns indicating that this approach consistently makes the model more language-agnostic, as it helps in some cases while hindering performance in others. Additionally, we experiment with discarding English examples and upsampling specific language groups (e.g., Serbian and Torlak examples for the Torlak dialect), which leads to slight performance improvements for the Torlak dialect.

Notably, we observe that full fine-tuning of the comparatively smaller, non-instruction-tuned, but language-specific BERTić model cannot surpass the performance of the multilingual, instruction-tuned Aya-101 model. Finally, we apply the same data combinations to perform instruction tuning on the Aya-101 model and observe an overall performance boost. However, our experiments with different numbers of training epochs (5 and 10) yield inconclusive findings.

4.2 Test Set Insights

Team-specific ranking Table 6 presents a comparison of the best-performing submissions from different teams on the Dialect-Copa test set. We categorize the submissions into two groups: Category 1 includes teams that utilize closed-source model weights, while Category 2 consists of teams that rely on open-source model weights. Our submissions belong to the latter category. We observe that the closed-source GPT-4 model achieves the best overall performance. Team JSI employs GPT-4 with a 10-shot prompting approach, where they provide the first 10 test instances without revealing

Base Model	Setting	en	hr	mk	sl	sl-cer	sr	sr-tor	Avg. (acc)
Takeaway 1: Combining all training data helps									
BERTić	Finetune (hr)	-	64	-	-	-	-	-	-
BERTić	o	0.65	0.67	0.55	0.67	0.49	0.66	0.59	0.61
Takeaway 2: Data augmentation helps for low-resource languages in most cases									
BERTić	otrslc	0.46	0.7	0.69	0.67	0.42	0.68	0.65	0.61
Takeaway 3: Script choice makes a difference (Using only Latin script performs better on Latin script evaluation)									
BERTić	otrsl	0.51	0.77	0.64	0.64	0.59	0.72	0.64	0.65
Takeaway 4: Cross-lingual mix-and-match effect: inconclusive									
BERTić	$\text{otrsl}_{\text{mix}}$	0.56	0.76	0.68	0.57	0.52	0.66	0.63	0.63
Takeaway 5: Upsampling certain language groups might help targeted evaluation in some cases									
BERTić	$\text{otrsl}_{\text{sr-tor}}$	0.55	0.74	0.63	0.61	0.48	0.64	0.66	0.62
BERTić	$\text{otrsl}_{\text{sl-cer}}$	0.54	0.68	0.65	0.57	0.58	0.66	0.64	0.62
Takeaway 6: A multilingually-instruction-tuned 13B parameter model performs significantly better									
Aya-101	4-shot	0.83	0.77	0.75	0.76	0.62	0.81	0.73	0.75
Takeaway 7: Further task-specific instruction tuning helps even more									
Aya-101	otrsl	0.86	0.79	0.81	0.91	0.7	0.82	0.77	0.81

Takeaway 7: Training for 10 epochs instead of 5: inconclusive

setting	epochs	en	hr	mk	sl	sl-cer	sr	sr-tor	mean	max count
otrsl	10	0.52	0.74	0.65	0.56	0.61	0.67	0.59	0.62	5
	5	0.50	0.72	0.62	0.57	0.58	0.65	0.61	0.61	2
$\text{otrsl}_{\text{mk-hr-ckm}}$	10	0.49	0.76	0.66	0.62	0.51	0.67	0.70	0.63	1
	5	0.48	0.76	0.68	0.64	0.52	0.71	0.67	0.64	4
$\text{otrsl}_{\text{sl-cer}}$	10	0.56	0.64	0.65	0.62	0.58	0.65	0.60	0.62	5
	5	0.54	0.68	0.65	0.57	0.58	0.66	0.64	0.62	5
$\text{otrslc}_{\text{sr-tor}}$	10	0.50	0.70	0.65	0.55	0.48	0.62	0.68	0.60	4
	5	0.49	0.69	0.63	0.60	0.51	0.62	0.70	0.61	4

Table 5: Takeaways from incremental experiments performed on the Dialect-Copa validation dataset. The best language-specific scores for each setting are underlined (Takeaway 7).

Team	Base Model	System Description	sl-cer	hr-ckm	sr-tor	Avg. (acc)
Closed Source Model Weights						
JSI	GPT-4	10-shot with first 10 test instances (without answer)	**0.734**	**0.890**	**0.974**	**0.866**
UNIRI	GPT-4	RAG implementation; Chakavian and Cerkno lexical dictionary; Reasoning instruction and self referral grading task	0.708	0.764	-	-
UNIRI	GPT-4	0-shot iterative prompt	0.664	0.774	0.894	0.777
Open Source Model Weights						
GmuNLP	Aya-101	4-shot prompting	0.694	0.756	0.840	**0.763**
GmuNLP	Aya-101	LORA adapter tuning on $\text{otrsl}_{\text{hr-ckm}}$ → 4-shot prompting	**0.700**	0.750	0.824	0.758
GmuNLP	Aya-101	LORA adapter tuning on otrsl → 4-shot prompting	0.682	**0.760**	0.824	0.755
GmuNLP	Aya-101	LORA adapter tuning on $\text{otrsl}_{\text{hr-ckm}}$ → 4-shot prompting	0.660	0.742	**0.848**	0.750
WueNLP	Mixtral	LORA adapter tuning on standard variety of target dialect	0.556	0.606	0.738	0.633
CLaC	XLM-R	Fine-tuning XLM-RoBERTa base for multiple choice QA task	0.564	0.522	0.570	0.552

Table 6: Performance comparison of different submissions on Dialect-Copa test set.

base model	setting	epoch	sl-cer	hr-ckm	sr-tor	Avg. (acc)
Aya-101	4-shot	-	0.694	**0.756**	0.840	**0.763**
Aya-101	otrslc$_{\text{mix-mk-hr-ckm}}$	5	0.690	**0.756**	0.836	0.761
Aya-101	otrsl$_{\text{mix-hr-ckm}}$	5	**0.700**	0.750	0.824	0.758
Aya-101	otrsl	5	0.682	**0.760**	0.824	0.755
Aya-101	otrsl$_{\text{mk-hr-ckm}}$	5	0.660	0.742	0.848	0.750
Aya-101	otrsl$_{\text{sl-cer}}$	5	0.686	0.718	0.836	0.747
BERTić	otrsl$_{\text{hr-ckm}}$	10	0.572	0.626	**0.722**	**0.640**
BERTić	otrsl$_{\text{mk-hr-ckm}}$	5	**0.582**	**0.634**	0.682	0.633
BERTić	otrslc$_{\text{mix-testset}}$	5	0.576	0.622	0.692	0.630
BERTić	otrsl	10	0.540	0.622	0.700	0.621

Table 7: GMUNLP system submissions for test-set evaluation. The best dialect-specific scores for each base-model type are bolded.

the answers. Interestingly, even the 0-shot prompting using GPT-4 (by team UNRI) outperforms all submissions in Category 2 using open-source models. Among the Category 2 submissions, GMUNLP (our submission) achieves the highest performance on all varieties. The base Aya-101 model with 4-shot prompting yields the best average score across all languages. However, LoRA adapter tuning on different data combinations results in language-specific best scores.

GMUNLP submission Table 7 presents the results of our selected 10 system submissions. We observe that the best performance achieved by the BERTić model on the otrsl setting is 62%, which is approximately 17% lower compared to the otrcl-tuned Aya-101 model. When comparing language-specific results, we find that the Torlak (sr-tor) dialect is the easiest to predict for both the Aya-101 and BERTi'c models, while the Cerkno dialect (sl-cer) proves to be the most challenging to learn.

Interestingly, upsampling the Cerkno dialect-related data (otrsl$_{\text{sl-cer}}$-tuned) does not yield the best score for the Cerkno test-set. Instead, upsampling the Chakavian dialect-related data using the otrsl$_{\text{mk-hr-ckm}}$ setting leads to better scores on the Cerkno test set. This observation holds true for both the Aya-101 and BERTić base models, indicating that leveraging data from more closely related languages does not always provide the most significant benefit. However, all varieties here are from the same language group and it is not unnatural that oversampling other languages or dialects can have a positive impact as well (given the brittleness of the fine-tuning process). We believe this phenomenon warrants further investigation to gain a deeper understanding of the complex interplay between language-relatedness and task-specific model performance.

5 Conclusion

In this study, we explored the impact of data augmentation techniques on fine-tuning multilingual language models for improving common sense reasoning in dialectal variations. Our experiments encompassed a range of language models, from smaller to mid-sized architectures, to investigate their adaptability to dialectal nuances. The observed variations in performance and the upper limits achieved by different models reflect the diverse ways in which language models handle and adapt to dialectal variations. The insights gained from this work may contribute to the development of more robust and adaptable language models that can handle the challenges posed by dialectal variations. Future work can explore advanced data augmentation techniques, investigate the impact of domain-specific knowledge integration, and develop novel architectures tailored to the unique characteristics of dialects.

References

Claude — anthropic.com. https://www.anthropic.com/claude. [Accessed 25-03-2024].

Croatian-chakavian features -UniLang — forum.unilang.org. https://forum.unilang.org/viewtopic.php?t=14771. [Accessed 28-03-2024].

Elitni Odredi - Ljubavi Moja lyrics + Croatian (Chakavian dialect) translation — lyricstranslate.com. https://lyricstranslate.com/en/ljubavi-moja-jubavi-moja.html. [Accessed 28-03-2024].

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Adeola Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Adrian Chifu, Goran Glavaš, Radu Ionescu, Nikola Ljubešić, Aleksandra Miletić, Filip Miletić, Yves Scherrer, and Ivan Vulić. 2024. VarDial evaluation campaign 2024: Commonsense reasoning in dialects and multi-label similar language identification. In *Eleventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Tech-*

nologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fahim Faisal, Orevaoghene Ahia, Aarohi Srivastava, Kabir Ahuja, David Chiang, Yulia Tsvetkov, and Antonios Anastasopoulos. 2024. Dialectbench: A nlp benchmark for dialects, varieties, and closely-related languages.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 434–452, Online only. Association for Computational Linguistics.

Andrew S. Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024. Mala-500: Massive language adaptation of large language models.

Nikola Ljubešić and Davor Lauc. 2021. BERTić - the transformer language model for Bosnian, Croatian, Montenegrin and Serbian. In Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing, pages 37–42, Kiyv, Ukraine. Association for Computational Linguistics.

Nikola Ljubešić, Nada Galant, Sonja Benčina, Jaka Čibej, Stefan Milosavljević, Peter Rupnik, and Taja Kuzman. 2024. DIALECT-COPA: Extending the standard translations of the COPA causal commonsense reasoning dataset to south slavic dialects. In Eleventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2024), Mexico City, Mexico. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In Proceedings

of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations, pages 46–54, Online. Association for Computational Linguistics.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 671–686, Singapore. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498, Online. Association for Computational Linguistics.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre,

Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model.

A   Croatian to Chakavian conversion rules

Here we report the collected Croatian to Chakavian conversion rules and their corresponding examples (uni), the lyrics translations (lyr) and the Claude-3 prompt we used to generate the Cahakavian synthetic translations.

Conversion rules

```
Rule: m's at the end of words become n's
Examples:
Ja sam = Ja san
osam = osan
s ženom = s ženon
vidim = vidin

Rule: đ becomes j
Examples:
mlađi=>mlaji
među=>meju

Rule: The sounds ć and đ actually do not
exist separately from č and dž

Rule: genitive plural in the feminine and
neuter takes a zero ending:
Examples: žena = žen, sela = sel

Rule: dative/instrumental plurals are
also somewhat different.

Rule: Masculine and neuter nouns can have
an alternate ending.
Examples:
it can be gradovima or gradoviman

Rule: For feminine nouns, it's shortened.
Examples:
ženama=>ženan

Rule:-ao endings are shortened to just -a
Examples:
išao je => iša je
rekao sam => reka san

Rule:-io endings change to -ija
Examples:
govorio je => govorija je
```

```
vidio sam => vidija san

Rule:ča, aside from meaning "what" can
also be used as a particle meaning "away"
or "out".
Examples:
gremo ča (let's get out of here)

Rule: some of the third person plural
forms can often be extended into longer
-u ending forms.
Examples:
govore -> govoru -> govoridu
rade -> radu -> radidu
pišu -> pišedu

Rule: Infinitive is shortened .
There is no I at the end.
In some speeches there is neither T/Ć at
the end
Examples:
bit =>bi
pivat =>piva
znat => zna
plivat => pliva
ronit => roni

Rule: change lj=>j
Examples:
zaljubiti se=>zajubit se
ljubav =>jubav
ljuska =>juska
ljudi=>judi

Rule: Change O => E in some cases
Example:
nekoga-nikega
svakoga-svakega
tomu-temu
toga-tega
bijeloga-bilega
jednoga-jenega
jednomu-jenemu

Rule: In standard croatian third person
plural has ending in some verbs E.
In chakavian it is always U
Examples:
vide =>vidu
hoće =>hoću
stoje =>stoju
```

```
stave =>stavu
motre =>motru
leže=>ležu


Rule: Change H=>V
Examples:
kruh =>kruv
kuhati =>kuvati
suh =>suv
gluh =>gluv[/b]
```

Lyrics translations We collected Croatian to Chakavian lyrics translations from the lyricstranslate.com site (lyr).

Claude-3 prompt We use the following prompt consisting the the above mentioned conversion rules, examples and lyrics translations:

```
<conversion_file.txt>


This file contains Croatian to Chakavian
dialect conversion grammar rules with
examples.


Now here are some Croatian sentences and
it's parallel Chakavian sentences:


<Croatian_lyrics.txt>
<Chakavian_lyrics.txt>.


Given these resources, I want you to translate
the following Croatian sentences to Chakavian
dialect.


<Croatian_training_sentences.txt>
```