JGU Mainz's Submission to the AmericasNLP 2024 Shared Task on the Creation of Educational Materials for Indigenous Languages

Minh Duc Bui
Johannes Gutenberg University Mainz
minhducbui@uni-mainz.de

Katharina von der Wense
Johannes Gutenberg University Mainz
University of Colorado Boulder
k.vonderwense@uni-mainz.de

Abstract

In this paper, we present the four systems developed by the Meenzer team from JGU for the AmericasNLP 2024 shared task on the creation of educational materials for Indigenous languages. The task involves accurately applying specific grammatical modifications to given source sentences across three low-resource Indigenous languages: Bribri, Guarani, and Maya. We train two types of model architectures: finetuning a sequence-to-sequence pointer-generator LSTM and finetuning the Mixtral 8x7B model by incorporating in-context examples into the training phase. System 1, an ensemble combining finetuned LSTMs, finetuned Mixtral models, and GPT-4, achieves the best performance on Guarani. Meanwhile, system 4, another ensemble consisting solely of fine-tuned Mixtral models, outperforms all other teams on Maya and secures the second place overall. Additionally, we conduct an ablation study to understand the performance of our system 4.[1]

1 Introduction

Natural language processing (NLP) serves as a valuable educational tool for facilitating the learning of (endangered) languages. One effective method for generating learning material involves a system automatically transforming sentences based on specific properties. Subsequently, language learners are tasked with replicating the transformation, thus reinforcing their understanding of the language structure. The AmericasNLP 2024 shared task on the creation of educational materials for Indigenous languages (ST 2) (Chiruzzo et al., 2024) focuses on creating such material for three low-resource Indigenous languages: Bribri, Guarani, and Maya. Participants are tasked with applying a specific grammatical property to a given source sentence and producing the accurate modification.

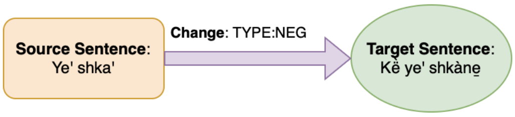

Figure 1: A Bribri sample from the shared task.

Our systems (which we submitted under the name "Meenzer Team") are ensembles composed of a range of models: finetuned character-level pointer-generator LSTMs (See et al., 2017), finetuned Mixtral 8x7B large language models (LLMs) (Jiang et al., 2024) via training on in-context examples, and GPT-4 (OpenAI, 2023). The main metric of the shared task is accuracy. We outperform all teams on Guarani by employing an ensemble across all models. Additionally, our ensemble of finetuned Mixtral models achieves the highest performance on Maya and reaches the second place overall.

The remainder of this paper is organized as follows: Section 2 details the task at hand and introduces the provided data. Following that, Section 3 dives into the details of our four system submissions. Section 4 presents the outcomes observed on both the development and test sets of the shared task. Lastly, an ablation study on our best performing system is provided in Section 5.

2 Task and Data

2.1 Task

In the context of this shared task, a source sentence is accompanied by a designated change feature, which the system is tasked with applying, see Figure 1. These features include modifications related to grammar, such as negation, and each sample can entail multiple concatenated grammatical alterations. While the shared task bears resemblance to morphological inflection shared tasks (Cotterell et al., 2016), where the goal is to modify a single word, our scenario necessitates adjustments to the

[1] The code is available at https://github.com/MinhDucBui/SharedTaskAmericasNLP2024.

Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP), pages 195–200
June 21, 2024 ©2024 Association for Computational Linguistics

	Train	Dev	Test
Bribri	310	213	481
Guarani	179	80	365
Maya	595	150	311

Table 1: Dataset sizes for each language and split.

entire sentence to accurately represent a specified property.

2.2 Data

The dataset encompasses three Indigenous languages: Bribri, Guarani, and Maya.[2] For each language, a training and a development set are provided. Additionally, the input side of the test set is given and used to submit predictions for the shared task's final evaluation. Within the training set, Bribri comprises 28 unique features, resulting in 135 distinctive combinations; Guarani encompasses 19 unique features, forming 21 combinations; and Maya has 33 unique features, yielding 52 combinations. A summary of the sample distribution per language and split is presented in Table 1.

3 Meenzer Team's System

Our systems consist of ensembles comprising various models, including finetuned character-level pointer-generator LSTMs, finetuned Mixtral 8x7B LLMs utilizing in-context finetuning, and GPT-4.

3.1 Pointer-Generator LSTM

Our first model group is a character-level sequence-to-sequence LSTM architecture, featuring an LSTM encoder and decoder equipped with an attention mechanism, alongside a pointer-generator (Bahdanau et al., 2015; See et al., 2017). The pointer-generator allows the LSTM to both copy words through pointing and generate characters from a predefined vocabulary (Vinyals et al., 2015).

In contrast to the typical sequence-to-sequence LSTM models, we use a separate LSTM encoder to encode the provided change features. For a detailed explanation of the sequence-to-sequence LSTM, we refer to Bahdanau et al. (2015). Furthermore, we deploy a pointer generator with a character-level vocabulary: At timestep t, given the attention distribution a^t over the characters in the source sequence, the decoder state s_t and the context vector h_t^*, the generation probability $p_{\text{gen}} \in [0, 1]$ is determined as:

<USER>: This is [LANGUAGE]. I will give you a source sentence and a grammar change and you have to output the correct change!
<Assistant>: Ok!
<USER>: Source Sentence: [SOURCE$_1$] / Grammar Change: [CHANGE$_1$]
<Assistant>: [TARGET$_1$]
<USER>: Source Sentence: [SOURCE$_{\text{sample}}$] / Grammar Change: [CHANGE$_{\text{sample}}$]
<Assistant>: [TARGET$_{\text{sample}}$]

Figure 2: An example of a 1-shot prompt for a sample, with [LANGUAGE] being replaced by the specific language under consideration. During training, we predict and compute the loss based on the [TARGET$_{\text{sample}}$] sequence. However, during testing, [TARGET$_{\text{sample}}$] is left blank and must be predicted.

$$p_{\text{gen}} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x + b_{ptr})$$

where vectors $w_{h^*}^t, w_s, w_x$ and the scalar b_{ptr} are all learnable parameters, while σ represents the sigmoid function. The probability p_{gen} serves as a soft switch, enabling the model to decide whether to generate a character from the vocabulary or to copy a character from the source sequence by sampling from the attention distribution a_t:

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i^t,$$

where $P_{\text{vocab}}(w)$ represents the probability distribution across all characters in the vocabulary, while $P(w)$ additionally adds all characters present in the source sequence.

Training We adopt a two-step training approach for our model: Initially, we train a model on the combined training sets of all three languages for 100 epochs, incorporating early stopping. Additionally, we employ hyperparameter tuning through 100 trials; see Appendix A.1. Subsequently, in preparation for our ensemble approach, we select the top 10 models and conduct further finetuning on each model using the dataset of the target language. This process is repeated independently for all three languages. Each change feature is assigned a distinct feature token, and we include language tags for each individual dataset, treating them as a change feature.

[2] https://github.com/AmericasNLP/americasnlp2024/tree/master/ST2_EducationalMaterials/data

	Bribri			Guarani			Maya			Avg.
	Acc.	BLEU	ChrF	Acc.	BLEU	ChrF	Acc.	BLEU	ChrF	Acc.
Dev Set										
`(1) LSTMs+Mixtrals+GPT4s`	**30.19**	**51.96**	**67.60**	**53.16**	**61.98**	**88.53**	**70.46**	**85.14**	**93.75**	**51.27**
`(2) LSTMs+Mixtrals`	**30.19**	**51.96**	**67.60**	49.36	58.09	86.33	**70.46**	**85.14**	**93.75**	50.00
`(3) LSTMs`	24.10	50.30	61.47	41.77	43.28	77.65	70.47	85.13	93.59	45.45
`(4) Mixtrals`	22.17	47.28	66.80	44.30	54.78	84.60	61.74	80.67	91.60	42.74
Test Set										
`(1) LSTMs+Mixtrals+GPT4s`	17.50	44.20	70.09	**34.62**	**49.60**	**84.93**	38.39	66.81	83.70	30.17
`(2) LSTMs+Mixtrals`	17.50	44.20	70.09	23.08	35.95	79.71	38.39	66.81	83.70	26.32
`(3) LSTMs`	8.54	32.50	61.24	12.64	20.01	71.61	27.74	58.59	79.29	16.31
`(4) Mixtrals`	**19.38**	**46.93**	**73.02**	23.90	36.94	79.48	**53.87**	**77.68**	**90.94**	**32.38**

Table 2: Our results on the development set (upper part) and the official results on the test set (lower part).

3.2 Mixtral 8x7B (Instruct)

Our second model is the Mixtral 8x7B (Instruct),[3] a LLM finetuned on instructional data (Jiang et al., 2024).

Architecture The Mixtral 8x7B model is a sparse mixture of experts language model (Shazeer et al., 2017), employing the same decoder-only transformer architecture as Mistral 7B (Jiang et al., 2023). However, it distinguishes itself by having each layer composed of 8 feedforward blocks, referred to as *experts*. At every token and layer, a router network selects two experts, which may vary at each timestep, to process the current state and combines their outputs. Consequently, while each token theoretically has access to 47B parameters, only 13B active parameters are utilized during inference. We leverage the instruction-tuned version.

Training We employ, what Li et al. (2023) call, supervised in-context learning (SICL), which differs itself from conventional in-context learning (ICL) by integrating in-context examples directly into the training phase (Min et al., 2022; Chen et al., 2022). We concatenate the task instruction, labeled in-context examples, and the target sequence to predict. Subsequently, we finetune the model to predict the target sequence, see Figure 2 for an example. In contrast, ICL generate predictions without adjusting model parameters.

To enhance both training and inference efficiency, we implement 4-bit quantization with LoRA (Dettmers et al., 2023). We train multiple LoRA adapters by varying the number of examples per prompt (k) and the number of epochs (m). Specifically, we experiment with $k = 5, 10, 20$ and $m = 10, 20$, resulting in a total of 6 models per language. Each LoRA adapter, applied onto the query and value projection matrices in the self-attention module, possesses a rank of 8. For each sample, examples are selected based on their overlap with the same or similar changes, with the top-k most similar examples chosen. Additionally, the order of the top-k examples is randomized for each epoch. We employ a learning rate of 1e-4 alongside a cosine learning rate scheduler, with a weight decay of 0.1.

3.3 GPT-4

In addition to Mixtral 8x7B, we incorporate GPT-4 using ICL. GPT-4, another LLM, is configured with $k = 20$ examples. We maintain consistency in example selection and prompt style with Mixtral 8x7B (Instruct). Specifically, we leverage the *gpt-4-turbo-2024-04-09* version of GPT-4.

3.4 Ensembling Strategy

Our four final systems consist of different ensembles constructed from the previously mentioned models, leveraging majority voting to reach a final decision, with the best-performing model on the development set breaking ties. To introduce more diversity for the LLMs, we generate two inference prompts: While one prompt organizes the top-k examples in ascending order, the other arranges them in descending order. Consequently, for each language, we have 10 LSTM, 12 Mixtral, and 2 GPT-4 predictions. For each system, we choose the best combination of models by evaluating their performance on the development set.

System 1 This system incorporates predictions from the LSTM, Mixtral 8x7B, and GPT-4 models. It is denoted by `(1) LSTMs+Mixtrals+GPT4s`.

[3] Model taken from `https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1`

System 2 This system comprises predictions from the LSTM and Mixtral 8x7B models, labeled as `(2) LSTMs+Mixtrals`.

System 3 This system solely relies on predictions from the LSTM models, identified as `(3) LSTMs`.

System 4 This system only considers the Mixtral models and is denoted by `(4) Mixtrals`.

4 Results

The primary metric for evaluating the shared task performance is accuracy (acc.), supplemented by BLEU (Papineni et al., 2002) and ChrF (Popović, 2015) as additional metrics. We present the results for the development set and test set in Table 2.

4.1 Development Set Results

The ensemble of all models demonstrates the highest performance, achieving an average accuracy of 51.24 and attaining the top scores across all languages. Notably, the only difference between `(1) LSTMs+Mixtrals+GPT4s` and `(2) LSTMs+Mixtrals` is in the Guarani language, where the addition of ChatGPT improves performance. When considering only LSTM models, we still achieve an average accuracy of 45.45, compared to 42.74 for Mixtral models.

4.2 Test Set Results

On the test set, we observe a significant difference from the reported development set results. The Mixtral ensemble performs best, achieving an accuracy of 32.38, approximately 10 points lower than its development set performance. Surprisingly, the LSTM ensemble performs notably worse, with an average accuracy of only 16.31. This decline in performance cascades through all other ensembles incorporating LSTM models: `(2) LSTMs+Mixtrals` achieves an average accuracy of 26.32, while `(1) LSTMs+Mixtrals+GPT4s` reaches an average of 30.17.

Nevertheless, our `(1)` system achieves the highest performance on Guarani among all shared task systems, while `(4) Mixtrals` attains the highest accuracy on Maya (tied with another team). Overall, our `(4) Mixtrals` system secures second place among all systems based on average accuracy.

Development & Test Set Discrepancy The LSTMs, constructed at the character-level and trained from scratch with a limited training set, might encounter numerous unknown characters. Analyzing the case-sensitive character overlap between the language specific training, development, and test sets reveals a substantial disparity. For instance, in the case of Bribri, we observe that, while 21% of samples in the development set contain unseen characters, this figure rises to 65.4% in the test set. Similarly, for Guarani, the proportion increases from 11.4% in the development set to 22.3% in the test set. Conversely, for Maya, while there are no unseen characters in the development set, they account for 15.5% of samples in the test set.

	Bribri	Guarani	Maya	**Avg.**
Ensemble vs. (Best) Single Model				
Mixtral (Single)	17.45	40.50	57.71	38.55
Mixtrals (Ensemble)	**22.17**	**44.30**	**61.74**	**42.74**
ICL vs. SICL				
Mixtral (ICL)	7.08	18.99	35.57	20.55
Mixtral (SICL)	**14.15**	**36.7**	**57.71**	**36.19**
Random Prompt Order				
Mixtral (Fix)	8.49	35.44	54.36	32.76
Mixtral (Random)	**14.15**	**36.70**	**57.71**	**36.19**

Table 3: Ablation study on the development set for `(4) Mixtrals`, our best system.

5 Ablation Study

In this section, we conduct a brief ablation study on our best-performing system, `(4) Mixtrals`. The results on the development set are presented in Table 3.

Ensemble vs. (Best) Single Model We demonstrate that assembling the Mixtral models into an ensemble boost performance by approximately 4.19 average accuracy points compared to the single best Mixtral model.

ICL vs. SICL For this and the following comparison, we fix the number of examples to $k = 20$ and epochs to $m = 10$. We observe that ICL, which does not adjust parameters, demonstrates an average accuracy of only 20.55, a notable 15.64 lower than SICL.

Random Order per Epoch: Finally, we investigate the impact of randomly varying the order of the k examples in the prompt per epoch on performance. We find that maintaining a fixed order (consistent during inference) leads to decreased performance across all languages, with an average accuracy decrease of 3.43.

6 Conclusion

We presented the systems of the Meenzer Team by JGU for the AmericasNLP 2024 shared task on the creation of educational resources. We trained character-level pointer-generator LSTMs as well as Mixtral 8x7B models finetuned through SICL. In addition, we used GPT-4 models via in-context learning. We secured second place with an ensemble of the finetuned Mixtral 8x7B models and reached the highest accuracy on Maya. Additionally, we achieved the highest performance on Guarani using an ensemble of LSTM, Mixtral, and GPT-4 models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Luis Chiruzzo, Pavel Denisov, Samuel Canul Yah, Lorena Hau Ucán, Marvin Agüero-Torales, Aldo Alvarez, Silvia Fernandez Sabido, Alejandro Molina Villegas, Abteen Ebrahimi, Robert Pugh, Arturo Oncevay, Shruti Rijhwani, Rolando Coto-Solano, Katharina von der Wense, and Manuel Mager. 2024. Findings of the AmericasNLP 2024 shared task on the creation of educational materials for indigenous languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Chengzu Li, Han Zhou, Goran Glavaš, Anna Korhonen, and Ivan Vulić. 2023. On task performance and model calibration with supervised and self-ensembled in-context learning. *Preprint*, arXiv:2312.13772.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

	Hyperparameter	values
Optimization	Batch size	{2, 4, . . . , 128}
	Learning rate	[$1e^{-5}$, 0.01]
	β_1	[.8, .999]
	β_2	[.98, .999]
	Label smoothing	[0, .2]
	Scheduler	{reduceonplateau, warmupinvsqrt, (none)}
	Warmup samples*	{0, 10, . . . , 1000}
	Factor*	[.1, .9]
	Min. learning rate*	[$1e^{-7}$, .001]
	Learning rate patience*	{1, 2, . . . , 5}
Architectural	Embedding Size	{16, 32, . . . , 512}
	Hidden layer size	{64, 128, . . . , 2048}
	Encoder & Decoder layers	{1, 2}
	Feature Attention heads	{1, 2}
	Dropout	[0, .5]

Table 4: LSTM hyperparameter space. Continuous distributions are denoted by intervals [. . .], while discrete ones show step sizes 1, 2, . . . , max. We uniformly sample from these, except for the learning rate, which follows a log uniform distribution. Hyperparameters and the distributions we sample from. * marks conditional hyperparameters, relevant only with chosen schedulers.

A Appendix

A.1 Hyperparameter Grid

We report in Table 4 the hyperparameter grid for our LSTMs.