

Nile-Chat: Egyptian Language Models for Arabic and Latin Scripts

Guokan Shang^{1*†}, Hadi Abdine^{1*}, Ahmad Chamma^{1*}, Amr Mohamed^{1*},
Mohamed Anwar¹, Abdelaziz Bounhar¹, Omar El Herraoui¹,
Preslav Nakov¹, Michalis Vazirgiannis^{1,2†}, Eric Xing¹

¹MBZUAI, ²Ecole Polytechnique

[†]Correspondence: {guokan.shang, michalis.vazirgiannis}@mbzuai.ac.ae

Abstract

We introduce Nile-Chat-4B, 3x4B-A6B, and 12B¹, a collection of LLMs for Egyptian dialect, uniquely designed to understand and generate texts written in both Arabic and Latin scripts. Specifically, with Nile-Chat-3x4B-A6B, we introduce a novel language adaptation approach by leveraging the Branch-Train-MiX strategy to merge script-specialized experts, into a single MoE model. Our Nile-Chat models significantly outperform leading multilingual and Arabic LLMs—such as LLaMa, Jais, and AL-LaM—on our newly introduced Egyptian evaluation benchmarks, which span both understanding and generative tasks. Notably, our 12B model yields a 14.4% performance gain over Qwen2.5-14B-Instruct on Latin-script benchmarks. All our resources are publicly available. We believe this work presents a comprehensive methodology for adapting LLMs to dual-script languages, addressing an often overlooked aspect in modern LLM development.

1 Introduction

Egyptian Arabic (also known as *Masri*) is the most widely spoken variety of Arabic, with over 100 million native speakers in Egypt and broader mutual intelligibility across the Arab world². It differs substantially from Modern Standard Arabic (MSA) in phonology, vocabulary, and grammar. A notable feature of this dialect is its widespread dual-script usage: native speakers often write Egyptian Arabic in both Arabic script and a Latin-based script commonly referred to as *Arabizi* or *Franco-Arabic* (e.g., “7aga gameda” for *حاجة جامدة*).

Despite the pervasiveness of this dual-script setting, most Large Language Models (LLMs) for Arabic fail to support it adequately. Existing models either focus on MSA or partially support dialects, and none are trained to handle the Latin

script. Moreover, no prior LLMs have *explicitly* targeted a single language across two scripts.

We introduce Nile-Chat³, an LLM family for Egyptian Arabic that natively supports two scripts. We release three model variants: dense models in 4B and 12B, and Nile-Chat-3x4B-A6B: a *Mixture-of-Experts* (MoE) model trained using the *Branch-Train-MiX* (BTX) method (Sukhbaatar et al., 2024). As shown in Figure 1, we merge script-specialized experts, each trained on either Arabic-script or Latin-script Egyptian data, into a unified MoE that dynamically routes tokens to the appropriate expert. This modular approach enables scalable adaptation without sacrificing performance or efficiency.

All Nile-Chat models undergo a full training pipeline with dual-script data we created—including continual pre-training on Egyptian Arabic corpora (e.g., transcripts, forum posts, and song lyrics), followed by fine-tuning on a variety of instruction tasks, and a final alignment-tuning stage for safety and preference adjustment. To support the evaluation, we also introduce a comprehensive evaluation suite covering both understanding (e.g., MMLU, HellaSwag) and generation (e.g., translation, transliteration) tasks in Arabic and Latin scripts. Nile-Chat models consistently outperform competitive baselines including LLaMa, ALLaM, Jais, and Qwen2.5 across all Egyptian-specific benchmarks. Notably, our 12B model improves Latin-script benchmark performance by 14.4% over Qwen2.5-14B-Instruct.

To the best of our knowledge, Nile-Chat is the first LLM to provide script-aware support for a widely spoken dialect. All models, data, and evaluation code are released publicly. We hope that our work will inspire further research on LLMs for underrepresented and dual-script languages.

*These authors contributed equally.

¹<https://hf.co/MBZUAI-Paris/Nile-Chat-12B>

²https://en.wikipedia.org/wiki/Egyptian_Arabic

³We chose *Nile* to reflect the cultural and geographical significance of the Nile river, which traverses Egypt.

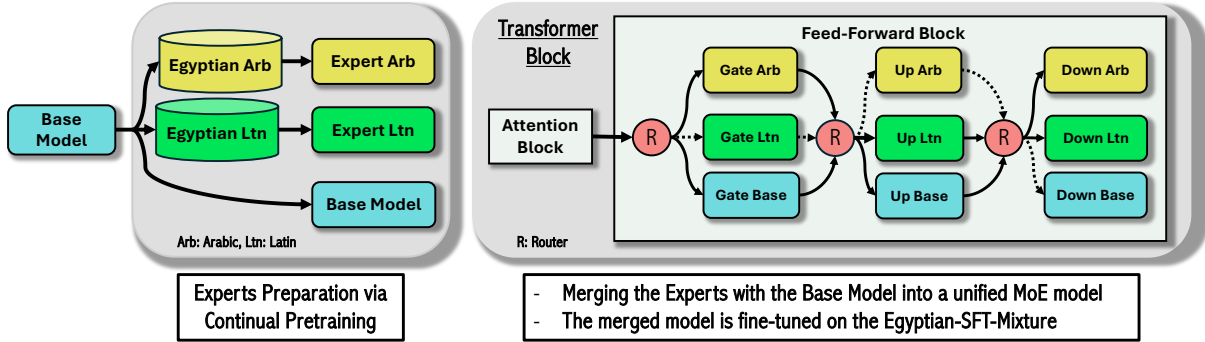


Figure 1: The training of Nile-Chat-3x4B-A6B using the *Branch-Train-MiX* (BTX) strategy. **Left:** Two experts are first *continual pre-trained* on Arabic-script and Latin-script corpora, respectively. **Right:** A *Top-2* token routing example within a transformer block, where the two script-specialized Experts have been merged with the Base Model into a unified *Mixture-of-Experts* (MoE) model through *instruction-tuning*.

2 Related Work

Arabic LLMs and Dialectal Models. The proliferation of Arabic-specific LLMs has included models like Jais (Sengupta et al., 2023), AceGPT (Huang et al., 2024), and ALLaM (Bari et al., 2024), trained primarily on MSA and English, often overlooking dialects. More closely related to our work, Atlas-Chat (Shang et al., 2025) introduced LLMs for Moroccan Arabic, demonstrating that dialectal models can outperform general multilingual models. Our Nile-Chat advances this paradigm, explicitly supporting the widely used Egyptian dialect, and uniquely, as written in both Arabic and Latin scripts.

Romanized Arabic and Dual-script Languages. Romanized Arabic—also known as *Arabizi* or *Franco-Arabic*—is widely used in informal communication, especially among youth (Yaghan, 2008; Alghamdi, 2018). It transcribes Arabic words using Latin characters and numerals (e.g., “3” for ع) and remains common in digital communication, despite broad support for Arabic script.

Prior work has focused on *detecting* and *transliterating* Arabizi into Arabic script (Darwish, 2013), treating it as a noisy input to be normalized. In contrast, we treat both scripts as *native inputs and outputs*, allowing the model to directly understand and generate Egyptian Arabic in either form.

Other languages such as Hindi, Serbian, and Kazakh (Koto et al., 2025) also exhibit dual-script usage. In Hindi, for example, the Nanda model (Choudhury et al., 2025) enhances robustness to Latin-script Hindi by augmenting the training data. Our work goes one step further; we use script-specialized experts within an MoE architec-

ture to model each script *explicitly*. To the best of our knowledge, Nile-Chat is the first LLM for Arabic that supports both native and Latin scripts in a unified framework.

Mixture-of-Experts. MoE models (Jiang et al., 2024a) efficiently scale LLM capabilities by selectively activating sub-networks. The recent Branch-Train-Mix (BTX) strategy (Sukhbaatar et al., 2024) allows fine-grained merging of specialized expert models, significantly reducing training costs. Our Nile-Chat-3x4B-A6B model innovatively applies BTX to script-specialized experts, efficiently integrating expertise in both Arabic and Latin scripts within a single model. This novel strategy demonstrates the viability of MoE architectures for linguistic specialization.

3 Dual-Script Training Data

The datasets feeding the Nile-Chat training fall into three broad categories:

Continual Pre-training: large-scale unlabeled Egyptian Arabic text drawn from audio / video transcripts, online forums, song lyrics, Wikipedia dumps, and web-scale crawls (see §3.1).

Instruction-tuning: prompt-response pairs covering a variety of instruction tasks, assembled from native Egyptian sources, and high-quality English translations (see §3.2).

Alignment-tuning: preference pairs used with Direct Preference Optimization to refine safety and mitigate undesirable behavior (see §3.3).

Across all of the above datasets, we ensure that roughly 25% is represented in the Latin script, complementing the Arabic-script majority and reflecting real-world usage patterns. The remainder of this section details each category in turn.

3.1 Continual Pre-training Datasets

As Egyptian Arabic is primarily used in spoken form, we first curated 854K audio / video **transcripts** to better capture its natural usage, yielding a total of 829M words. To broaden coverage, we supplemented the collection with publicly available datasets spanning diverse domains and styles. These include the **EFC-mini** (Egyptian Forums Corpus-mini) (Qarah, 2024), the **EDC** (Egyptian Datasets Collection)⁴, the **Egyptian Wikipedia dump**⁵, the Egyptian subset of the **ADD** (Arabic Dialects Dataset)⁶, the Egyptian partition of **FineWeb-2** (Penedo et al., 2025), the Egyptian subset of the **Habibi lyrics corpus** (El-Haj, 2020), and a small collection of scraped forum posts from **Fatatkat**⁷. Full details are provided in Appendix C.2.

The resulting pre-training corpus contains 1.15B words, predominantly in Arabic script. To balance this, we used Claude to transliterate a portion into Latin script (see the prompt in Appendix C.1). For this, we selected samples from the transcripts, EFC-mini, and EDC datasets, which feature informal content such as conversations, social media posts, and user comments—domains where Latin script is frequently used in practice. This process resulted in a total of 255M words in Latin script.

3.2 Instruction-tuning Datasets

To fine-tune the models for instruction following in Egyptian Arabic, we created the **Egyptian-SFT-Mixture**⁸ of 1.85M instructions by consolidating multiple sources, as illustrated in Figure 2. We began by incorporating publicly available datasets from prior work. To broaden coverage across domains and tasks, we translated some English instruction datasets into Egyptian Arabic. Finally, we augmented the mixture with data for translation between Egyptian, English, and MSA, as well as transliteration where users request conversion between Arabic and Latin scripts. The dataset is formatted as user-assistant messages in Appendix B.2.

3.2.1 Existing Egyptian Instruction Datasets

To the best of our knowledge, the **Aya Collection** (Singh et al., 2024) is the only large-scale multilingual instruction dataset that provides a readily

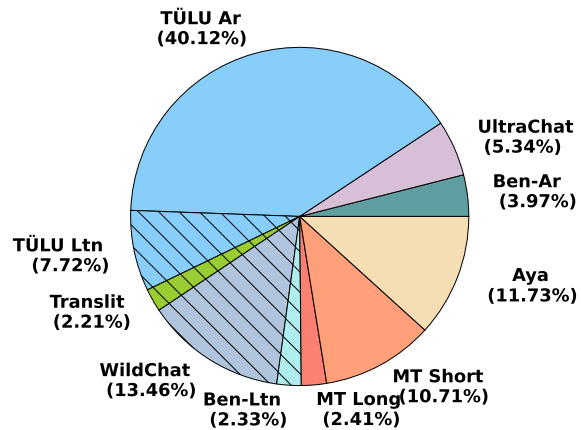


Figure 2: Composition of our Egyptian-SFT-Mixture instruction-tuning dataset. The acronyms "MT", "Ar", "Ltn", "Translit" and "Ben" are used to denote "Machine Translation", "Arabic", "Latin", "Transliteration", and "Benchmarks Training Set" respectively. The hatched regions represent parts in Latin script.

usable subset in Egyptian Arabic, with over 3.5M samples across a wide range of tasks. These include *paragraph writing*, *text classification*, *paraphrase identification*, *question-answering*, *summarization*, and *text simplification*. To ensure language consistency, we applied a *Glottid*-based language identification filter (Kargaran et al., 2023) to exclude non-Egyptian Arabic samples.

3.2.2 Translated English Instruction Datasets

We began by examining instruction-tuning datasets used to fine-tune recent state-of-the-art models. **TÛLU Collection** stands out for its broad domain coverage, including *instruction following*, *knowledge recall*, *reasoning*, and *safety*. The dataset mixture was systematically designed based on findings from ablation studies of both human-annotated and AI-generated data, with a deliberate emphasis on complexity and diversity. Appendix B.1 presents descriptions of each of the nested datasets, and describes how the subset was sampled. TÛLU-v3-mix (Lambert et al., 2024) is the successor of TÛLU-v2-mix (Iverson et al., 2023) with some intersected samples. We chose in this work to include both versions after eliminating the nested datasets where a newer version is provided, and performed a string-based de-duplication step for the remaining parts where 9,660 samples were removed. This forms our initial TÛLU-v2&3-mix dataset.

To improve quality, we first applied a preliminary filtering process to the v2&3 dataset, removing instructions that were unsuitable for typ-

⁴<https://github.com/Mostafanofal453/2.5-Million-Rows-Egyptian-Datasets-Collection>

⁵<https://dumps.wikimedia.org/arwiki/>

⁶<https://elhaj.uk/corpora.html>

⁷<https://forums.fatakata.net>

⁸<https://hf.co/datasets/MBZUAI-Paris/Egyptian-SFT-Mixture>

ical Egyptian users or likely to lose meaning in translation—such as *scientific content*, *translation tasks*, and *non-English text*. For English-to-Egyptian Arabic translation, we compared GPT-4o and Claude 3.5 Sonnet. Based on qualitative evaluation, Claude produced more natural and dialect-appropriate outputs, and was ultimately selected for translating the remaining data. Finally, to rectify the issues introduced by the automatic translation, a series of post-processing measures were implemented. All details are provided in Appendix B.3. Similar to our pre-training data, we selected a subset of the data—primarily focused on chat-style examples across various topics—and processed them into the Latin script.

Although TÜLU-v2&3-mix includes instructions from diverse domains, it contains only around 38K multi-turn conversations (with at least two turns). To improve the model’s ability to sustain longer dialogues (Zhao et al., 2024a), we incorporated data from **UltraChat** (Ding et al., 2023), a multi-round dialogue dataset that covers world knowledge, writing, and creative tasks. The dataset contains over 300K conversations, each with a minimum of five exchanges. We selected the longest examples—those with 7 to 8 turns—and applied the same processing procedures described for v2&3.

To further increase Latin-script representation, we also included data from **WildChat** (Zhao et al., 2024b), a dataset of 1M dialogues between users and ChatGPT, organized by script, language, and country. From the English subset (over 450K samples), we selected the first 300K conversations sorted by ascending length—based on the assumption that the Latin script is more common in short- to mid-length exchanges. These samples were translated into Egyptian Arabic in the Latin script and post-processed following the same procedure described above.

3.2.3 Translation and Transliteration Tasks

The final portion of our instruction data specifically targets two tasks: translation and transliteration.

Short Sentence Translation

We incorporated four publicly available translation datasets into our mixture. These include **EGY_MSA_Translation** (Faheem et al., 2024), a parallel corpus of Egyptian Arabic and MSA sentences collected from social media; **ArzEn-MultiGenre** (Al-Sabbagh, 2024), which includes professionally translated texts across songs, novels,

and TV subtitles; **Egyption_2_English**⁹, a 22k-sample dataset of everyday bilingual sentences; and **Oasst2-9k-translation**¹⁰, which provides English prompts aligned with Egyptian Arabic and MSA outputs, generated using GPT-4o. Detailed descriptions are provided in Appendix C.3.

The collected samples were converted into training instructions using randomly selected Egyptian-based templates (see Appendix A.1). We cover four translation directions: Egyptian Arabic to English, to MSA, and vice versa. To enhance multi-turn translation capabilities, a portion of the dataset includes 3-shot examples and 3-turn conversations. 10% of the data is reserved for evaluation.

Long Document Translation

The above collection of translation samples—whether derived from native translators or advanced models—mostly consists of short sentences. To equip the model with the ability to handle mid- to long-form translation (i.e., multi-line documents), we further used data from the Egyptian Wikipedia dump. We removed entries that were not relevant for translation, such as indicators of missing content, empty pages, and astronomy-related topics, which are overrepresented in the dump. We retained documents with word counts between 90 and 1,500 and applied a *Glottid* filter to eliminate non-Egyptian Arabic samples. These documents were then translated into English and MSA using Claude, and subsequently transformed into training instructions using the template provided in Appendix A.1.

Transliteration

To enable our model to perform script conversion between the Arabic and the Latin scripts, we use the Egyptian Forums Corpus (EFC), introduced by Qarah (2024), which contains user-generated texts from various Egyptian online forums. To promote sample diversity, we removed frequent keyterms related to sports. We then selected sentences with lengths between 50 and 70 words and applied a *Glottid* language filter to ensure dialectal consistency. From the filtered set, we retained final samples and converted them from the Arabic to the Latin script to build a parallel corpus. These were then transformed into training instructions using the templates given in Appendix A.2.

⁹https://hf.co/datasets/Abdalrahmankamel/Egyption_2_English

¹⁰<https://hf.co/datasets/ahmedsamirio/oasst2-9k-translation>

3.3 Alignment-tuning Datasets

To improve the overall model behavior, we applied a targeted alignment phase using Direct Preference Optimization (DPO) (Rafailov et al., 2023), combining on- and off-policy strategies. This was motivated by human evaluations of our SFT-stage model trained only on our pre-training and instruction data, which revealed several issues, including:

Overly Cautious. We observed that the SFT-stage model frequently refused to answer legitimate questions due to excessive caution. To address this, we leveraged 50% of the safety-related instructions retained from the SFT phase. For these samples, we applied an on-policy DPO strategy: the original assistant output was treated as the preferred response, while a corresponding rejected response was generated using the SFT-stage model itself.

Excessive Code-Switching. We observed that the SFT-stage model exhibited excessive code-switching between Arabic and English (Mohamed et al., 2025), even when the prompt was exclusively written in Arabic. To mitigate this behavior, we applied an off-policy correction procedure wherein instances from the SFT dataset exhibiting the identified patterns were selected and reformulated using Claude to produce more natural code-switched alternatives. The selection criteria and the correction prompt are described in detail in Appendix C.4.

Failures in Instruction Tasks. Additionally, the SFT-stage model displayed shortcomings in several instruction-following capabilities, notably:

- *Length control:* The model frequently ignored explicit length requirements (e.g., producing a 400-word script when 600 words were requested).
- *Stylistic control:* Rewriting or rephrasing with a specific tone (e.g., formal, humorous) was often inaccurate or superficial.

To address these issues, we again applied on-policy DPO strategy. We synthetically curated 1,000 minimal yet precise prompts, annotated poor completions from the SFT-stage model as rejections, and synthetically constructed new completions as positive demonstrations using Claude. The resulting preference pairs improved the model’s resilience to diverse user requests and yielded finer-grained control over its responses. Our DPO datasets are publicly available¹¹.

¹¹<https://hf.co/datasets/MBZUAI-Paris/Egyptian-DPO-Mixture>

4 Training Details: Dense Models

This section details the training setup across the pre-training, instruction-tuning, and alignment-tuning phases of Nile-Chat-4B and 12B dense models.

Base Model Selection. We adopt the base Gemma-3 (Team et al., 2025) models as the starting point for training Nile-Chat, due to their superior performance on Arabic tasks in our preliminary evaluation compared to other state-of-the-art multilingual and Arabic-specialized models.

Training Pipeline and Hyperparameters. For our dense models, we merged all Arabic- and Latin-script datasets into a single corpus and trained on this unified mixture, in contrast to our MoE models described in Section 5.

Continual Pre-training: We used Low-Rank Adaptation (LoRA) with rank 256 and alpha 128. The optimizer is AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. This stage is divided into:

- *Continual pre-training.* We run the training for 1 epoch using our data from Section 3.1, with a learning rate of $8e-6$, a warmup ratio of 1%, and a cosine decay to $1e-6$.
- *Annealing phase.* During this phase, training gradually shifts focus to a smaller set of high-quality Egyptian Arabic data. We run the training for 1 epoch and set the learning rate to $3e-4$ for the 4B model and $5e-5$ for the 12B model, and a cosine decay to 0.

Instruction-tuning (SFT): Next, we fine-tuned the model on our data from Section 3.2. We used LoRA with rank 256 and alpha 128. We ran the training for 2 epochs, and set the learning rate to $3e-5$ for the 4B model and $2e-5$ for the 12B model with a warmup ratio of 3%, linear decay to 0, and total effective batch size of 128. The loss is computed on the responses only. We used the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Alignment-tuning (DPO): Finally, we applied DPO to improve the overall model behavior, using the data constructed in Section 3.3. We followed standard DPO heuristics, notably reducing the SFT learning rate by an order of magnitude. Specifically, we evaluated learning rates of $3e-6$ and $5e-6$ with preference temperatures $\beta \in \{0.1, 0.5\}$, and compared full fine-tuning to LoRA. The experiments on the Nile-Chat-4B model showed that full fine-tuning with $3e-6$ and $\beta = 0.5$ consistently performed better, both in benchmarks and human

tests. We adopted this configuration for the final alignment phase of the Nile-Chat-12B model.

We performed the training on 8×NVIDIA A100 80GB GPUs using Fully Sharded Data Parallel (FSDP) on AWS SageMaker. The maximum input context length was configured to 2,048 tokens. We used `bf16` for faster training.

5 Training Details: MoE Models

Recent literature has highlighted that dense models are prone to catastrophic forgetting—particularly during fine-tuning—as new inputs often overwrite previously acquired knowledge (Li et al., 2024a). This effect is linked to data saturation, where model capacity is insufficient to retain all learned information. While scaling up dense models can alleviate forgetting to some extent, it comes at the cost of significantly higher inference budgets, since all parameters are used for every input. Mixture-of-Experts (MoE) models (Lo et al., 2024) offer a more efficient alternative. By assigning tasks to specialized experts and routing at the token level (Jiang et al., 2024a), MoEs isolate parameter updates, thereby reducing interference and preserving prior knowledge. This modular design enables MoEs to mitigate forgetting more effectively than dense models, while maintaining lower computational overhead.

Instead of training an MoE model from scratch, Sukhbaatar et al. (2024) show a recycling strategy called **Branch-Train-Mix (BTX)**. This method constructs an MoE model by merging several pre-trained base models. Specifically, the feed-forward layers of these models are repurposed as distinct experts within a new MoE layer, while a trainable routing network assigns each token to the most relevant expert path. The remaining layers—such as attention and embeddings—are merged by averaging their parameters across the base models, forming a shared backbone. Finally, the resulting MoE model is fine-tuned on an SFT dataset to align the components and optimize joint performance.

As illustrated in Figure 1, we propose a novel LLM adaptation strategy for dual-script languages by applying BTX to script-specialized experts. First, the base model is continually pre-trained on Arabic-script and Latin-script datasets separately to create script-specialized experts—differing from the unified training used for our dense models described in Section 4. Second, the pre-trained experts and the base model are merged using the BTX scheme described above, resulting in a new MoE

model with three experts—two of which are active per input—with a total of 6B activated parameters. This yields our final **Nile-Chat-3x4B-A6B** model. For comparison, we also merged the two script-specialized experts without including the base model, producing a **2x4B-A6B** variant. We consider the three-expert variant as our primary model, as incorporating the base model as an additional expert integrates broader general knowledge and English capabilities that go beyond the scope of the script-specialized experts. The unified MoE models then undergo two training phases: (1) *SFT* using a LoRA setup with an alpha of 512, a learning rate of 1e-4, and an effective batch size of 256. Since the English-centric base model is included as a third expert, we also mixed in Egyptian-SFT-Mixture a small amount of English instructions to recover its original English performance. (2) *DPO* serves as the final alignment stage.

6 Evaluation Benchmarks

To evaluate the performance of our models, we created eight benchmarks by translating widely used English LLM benchmarks into Egyptian Arabic using Claude, with four of them also rendered in the Latin script. Additionally, we evaluated using held-out test sets from our translation and transliteration datasets (see Section 3.2), collectively referred to as **EgyptianBench**¹². All our custom benchmarks are integrated into a fork¹³ of the LM-Evaluation-Harness repository (Gao et al., 2024) to ensure reproducibility and foster future comparison.

EgyptianMMLU¹⁴. We combined two sources: *ArabicMMLU-egy* (Mousi et al., 2025), an Egyptian translated version of ArabicMMLU (Koto et al., 2024) using an in-house dialect translation system and subsequently validated by human annotators, and *English MMLU* (Hendrycks et al., 2020), which we translated directly into Egyptian.

Belebele-Arz (Bandarkar et al., 2023). It is a multiple-choice machine reading comprehension benchmark across many languages. We adopted the provided Egyptian Arabic subset directly.

EgyptianHellaSwag (Zellers et al., 2019)¹⁵. It

¹²<https://hf.co/datasets/MBZUAI-Paris/EgyptianBench>

¹³<https://github.com/MBZUAI-Paris/lm-evaluation-harness-nile-chat>

¹⁴<https://hf.co/datasets/MBZUAI-Paris/EgyptianMMLU>

¹⁵<https://hf.co/datasets/MBZUAI-Paris/EgyptianHellaSwag>

Model	Egyptian MMLU	Belebele_Arz	Egyptian HellaSwag	Egyptian PIQA	Egyptian WinoGrande	Egyptian OpenBookQA	Egyptian RACE-H	Egyptian RACE-M	Egyptian AlpacaEval	Long Translation			Short Translation			Transliteration		
										BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
gemma-3-4b-it	46.08	38.56	42.56	60.32	56.49	35.79	33.68	40.06	85.30	20.67	44.75	73.03	4.76	31.15	52.98	1.44	20.36	47.54
jais-family-6p7b-chat	42.60	57.33	49.18	62.23	57.04	33.33	34.72	37.50	45.86	12.71	36.53	68.07	8.73	31.52	56.78	0.70	10.64	42.51
jais-adapted-7b-chat	40.96	55.67	40.85	56.50	54.35	32.89	34.62	42.33	21.45	10.61	27.56	63.48	9.19	24.85	53.52	1.11	6.14	40.45
Qwen2.5-7B-Instruct	45.74	64.22	45.47	58.02	56.41	38.70	35.45	41.76	58.80	19.89	44.80	73.64	11.34	36.31	54.96	2.74	20.63	49.32
ALLaM-7B-Instruct-preview	60.08	67.67	57.29	66.10	62.18	40.04	39.50	45.17	69.55	26.57	52.59	78.34	25.20	48.12	65.97	2.10	18.92	49.42
c4ai-command-r7b-arabic-02-2025	50.97	70.67	50.39	61.84	57.20	36.91	41.89	46.02	73.36	25.18	50.26	77.97	23.30	45.34	65.20	3.52	24.57	50.49
Llama-3.1-8B-Instruct	42.88	55.89	43.10	57.97	54.27	35.57	34.41	40.34	52.35	12.90	32.58	68.76	9.06	28.56	54.19	3.26	17.55	48.71
AceGPT-v2-8b-chat	55.25	73.33	53.14	62.50	58.39	39.82	41.06	47.16	93.33	24.59	49.39	77.57	22.47	44.97	66.30	4.80	23.52	49.33
gemma-2-9b-it	50.72	49.44	49.53	61.35	61.79	35.79	40.23	48.01	81.66	23.09	46.98	75.42	11.73	39.00	60.42	2.68	24.28	48.26
gemma-3-12b-it	<u>61.55</u>	<u>77.00</u>	49.49	64.96	63.53	38.03	41.27	48.86	92.61	22.90	45.97	73.46	5.24	32.82	54.34	2.77	26.16	50.47
jais-family-13b-chat	44.85	66.33	52.99	64.85	57.91	36.91	33.26	38.64	52.52	10.41	31.98	64.15	8.64	30.10	57.00	0.84	11.35	44.71
jais-adapted-13b-chat	50.03	65.33	47.53	61.30	56.72	37.14	35.45	41.76	52.91	15.53	41.48	70.86	15.96	38.81	63.52	1.00	13.33	46.08
Qwen2.5-14B-Instruct	60.81	72.33	55.84	63.97	59.97	38.26	43.25	50.28	71.35	21.71	45.55	73.36	9.26	34.21	53.89	4.07	25.83	51.41
Nile-Chat-4B	50.25	68.56	55.92	67.30	61.87	40.94	42.10	46.02	86.95	37.49	58.40	84.30	30.35	52.01	74.07	51.46	80.44	89.59
Nile-Chat-2x4B-A6B	52.05	73.89	<u>59.69</u>	68.67	<u>62.26</u>	<u>41.61</u>	44.07	<u>51.14</u>	<u>94.58</u>	<u>41.98</u>	<u>61.59</u>	<u>86.11</u>	<u>33.40</u>	<u>53.71</u>	<u>76.78</u>	<u>57.75</u>	<u>83.89</u>	<u>91.05</u>
Nile-Chat-3x4B-A6B	52.13	75.44	59.30	<u>69.27</u>	57.91	41.16	<u>44.59</u>	48.30	94.18	42.43	61.90	86.26	34.56	55.37	76.97	57.79	83.97	91.13
Nile-Chat-12B	62.59	79.44	64.04	70.69	63.53	42.06	48.02	53.13	95.56	40.53	60.61	85.45	32.20	53.53	74.72	52.21	80.97	89.71

Table 1: Performance comparison of Nile-Chat and state-of-the-art models on the **Arabic-script** benchmarks. The highest scores are indicated in **bold**, the second-highest are underlined. Figure 3 shows the average score over all the benchmarks and measures for each model.

presents complex scenarios where models must select the most plausible continuation of a given context from four options, challenging nuanced language understanding and contextual inference.

EgyptianPIQA (Bisk et al., 2020)¹⁶. The Physical Interaction Question Answering (PIQA) evaluates physical commonsense reasoning, presenting pairs of *Goal* and *Solution* options about everyday interactions with the physical world.

EgyptianWinoGrande (Sakaguchi et al., 2021)¹⁷. It consists of fill-in-the-blank coreference problems where models must choose the correct noun phrase to resolve an ambiguous pronoun, a task demanding nuanced commonsense reasoning.

EgyptianOpenBookQA (Mihaylov et al., 2018)¹⁸. This benchmark contains elementary-level science questions that require both explicit facts and broader commonsense knowledge; in translating it to Egyptian Arabic, we preserved scientific terminology to keep the questions accurate.

EgyptianRACE (Lai et al., 2017)¹⁹. ReAding ComprEhension (RACE) consists of English exam questions for middle and high school students, evaluating cognitive skills including reading comprehension, summarization, inference, and reasoning. In translating it to Egyptian Arabic, we preserved its narrative structure and question integrity.

¹⁶<https://hf.co/datasets/MBZUAI-Paris/EgyptianPIQA>

¹⁷<https://hf.co/datasets/MBZUAI-Paris/EgyptianWinoGrande>

¹⁸<https://hf.co/datasets/MBZUAI-Paris/EgyptianOpenBookQA>

¹⁹<https://hf.co/datasets/MBZUAI-Paris/EgyptianRACE>

EgyptianAlpacaEval (Dubois et al., 2024)²⁰. AlpacaEval is designed to evaluate instruction-following capabilities via pairwise comparison. We adapted this framework to Egyptian Arabic by constructing a culturally grounded evaluation set in the Arabic script. In this setting, a judge model compares two responses generated by different models for the same prompt and selects the one that best aligns with Egyptian linguistic norms, cultural values, and pragmatic appropriateness.

7 Results

Evaluation Measures. We used *accuracy* as the evaluation metric across all multiple-choice QA benchmarks, except for EgyptianHellaSwag, we adopted *normalized accuracy*. For translation and transliteration tasks, we used BLEU and chrF to evaluate surface-level correspondence, and BERTScore to assess the semantic similarity between the model outputs and the reference texts. Specifically, for BERTScore computation, we used multilingual BERT (mBERT) (Devlin et al., 2019) for translations into Egyptian Arabic, AraBERT (Antoun et al., 2020) for translations into MSA, and BERT-base for translations into English. For the transliteration tasks in both directions (Arabic to Latin and Latin to Arabic), we used mBERT.

The EgyptianAlpacaEval uses an LLM-as-a-Judge approach (Zheng et al., 2023), where Claude is tasked with selecting the more culturally appropriate response between two candidates. We used AceGPT-v1.5-13B-Chat (Zhu et al., 2024) as the reference model. We generated the candidate out-

²⁰<https://hf.co/datasets/MBZUAI-Paris/EgyptianAlpacaEval>

Model	Egyptian HellaSwag	Egyptian PIQA	Egyptian WinoGrande	Egyptian RACE-H	Egyptian RACE-M
gemma-3-4b-it	30.90	52.76	48.57	25.47	26.94
jais-family-6p7b-chat	30.27	53.25	52.14	24.18	28.06
jais-adapted-7b-chat	30.81	51.67	50.40	24.38	28.06
Qwen2.5-7B-Instruct	30.51	51.88	50.95	24.88	26.11
ALLaM-7B-Instruct-preview	32.17	53.09	50.63	25.07	31.94
c4ai-command-r7b-arabic-02-2025	30.88	52.32	51.43	25.07	27.22
Llama-3.1-8B-Instruct	31.77	53.30	50.24	24.48	28.33
AceGPT-v2-8b-chat	33.16	53.80	50.24	26.07	30.56
gemma-2-9b-it	33.75	53.69	50.79	26.66	28.61
gemma-3-12b-it	37.52	53.14	51.19	31.02	35.28
jais-family-13b-chat	30.46	53.09	48.18	25.28	27.78
jais-adapted-13b-chat	31.14	52.87	50.79	23.98	26.11
Qwen2.5-14B-Instruct	33.49	52.87	53.41	27.35	30.28
Nile-Chat-4B	50.55	65.32	<u>60.62</u>	37.36	43.06
Nile-Chat-2x4B-A6B	<u>55.49</u>	<u>68.00</u>	<u>61.33</u>	40.24	<u>45.56</u>
Nile-Chat-3x4B-A6B	<u>55.00</u>	<u>66.68</u>	56.42	<u>40.44</u>	42.78
Nile-Chat-12B	53.71	65.10	59.98	<u>41.72</u>	<u>48.89</u>

Table 2: Performance comparison of Nile-Chat and state-of-the-art models on the **Latin-script** benchmarks.

puts using the default sampling-based decoding for each model. We applied the chat template for all benchmarks, except for EgyptianWinoGrande.

Result Analysis. The evaluation results in Tables 1 and 2 demonstrate the exceptional performance of the Nile-Chat models across all Egyptian benchmarks in both the Arabic and the Latin scripts.

Compared to models with 7B parameters or fewer, **Nile-Chat-4B** demonstrates consistently superior performance across multiple *Arabic-script* benchmarks, achieving relative gains of 1.2% on EgyptianPIQA, 0.9% on EgyptianOpenBookQA, 0.21% on EgyptianRACE-High, and 1.6% on EgyptianAlpacaEval over the strongest competitor for each task. It also ranks first in translation and transliteration tasks across all evaluation metrics. On the *Latin-script* benchmarks, 4B outperforms all models in the same size category, by sizable margins: +18.38% on EgyptianHellaSwag, +12.97% on EgyptianPIQA, +8.48% on EgyptianWinoGrande, +11.91% on EgyptianRACE-High, and +11.12% on EgyptianRACE Medium, relative to the next-best model. *This indicates that existing LLMs underrepresent or overlook the Latin script.*

Nile-Chat-12B, on the other hand, pushes the state-of-the-art even further. Across the *Arabic-script* benchmarks, it achieves the highest score on every task, with the largest absolute improvements of +4.35% on EgyptianHellaSwag and +3.43% on EgyptianRACE-High over the next-best model. It also performs exceptionally well on the *Latin-script* and generation benchmarks, leading on EgyptianRACE-High (+1.28%) and EgyptianRACE-Medium (+3.33%), and ranking

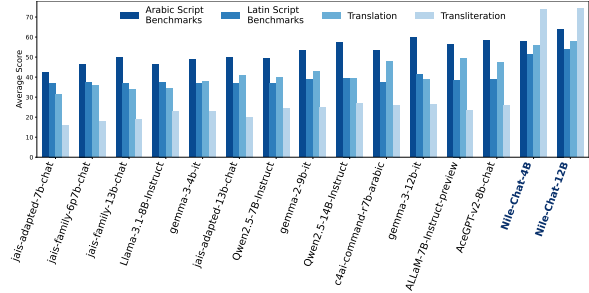


Figure 3: Average model scores over the benchmarks.

consistently within 1–3% of the top-performing models on the remaining Latin tasks, translation, and transliteration metrics. In all such cases, the models that marginally outperformed it belong to the MoE-based Nile-Chat family.

Nile-Chat-3x4B-A6B and **2x4B-A6B** strike a balance between the 4B and 12B dense models on discriminative *Arabic-script* tasks, yet excel whenever extensive generation or *Latin-script* processing is required. On EgyptianHellaSwag, they score 59.69% and 59.30%, respectively, which ranks them between the dense 4B (55.92%) and 12B (64.04%) models. A similar pattern holds for EgyptianPIQA. In *Latin-script*, 2x4B-A6B leads three of five benchmarks, widening the gap with the 4B dense model by 4.94% on EgyptianHellaSwag and 2.68% on EgyptianPIQA, while keeping within approximately 1–3% of the 12B model on the Latin RACE tasks. For generation tasks, 3x4B-A6B achieves the highest scores across all translation and transliteration tasks and metrics.

8 Conclusion

We introduced Nile-Chat, a family of language models specifically designed for the Egyptian Arabic dialect, uniquely capable of understanding and generating texts in both Arabic and Latin scripts. Our novel Branch-Train-MiX (BTX) based MoE model effectively integrates script-specialized experts, demonstrating superior performance across various benchmarks compared to leading multilingual and Arabic-specific models. Nile-Chat significantly enhances LLM capabilities in dual-script settings, achieving sizable improvement over current state-of-the-art models on Latin-script tasks. By releasing all our resources, datasets, and evaluation suites publicly, we aim to encourage further research and development in dual-script language modeling, addressing critical gaps for widely spoken yet underrepresented languages.

Limitations

Despite the promising results, our work has some limitations. First, the model occasionally generates hallucinations. Second, the dataset may contain inherent biases that could affect the model’s fairness and representation. Additionally, we relied heavily on Claude for translating English instructions into Egyptian Arabic. However, because Claude is primarily trained on English and reflects Western cultural values, it may not fully capture the unique nuances of Egyptian Arabic. We intend to address these limitations in future work.

References

- Rania Al-Sabbagh. 2024. Arzen-multigenre: An aligned parallel dataset of egyptian arabic song lyrics, novels, and subtitles, with english translations. *Data in Brief*, 54:110271.
- Hamdah Abdullah Alghamdi. 2018. *Arabizi: An exploration of the use of the contemporary youth netspeak on Social Networking Sites in Saudi Arabia*. Ph.D. thesis, University of Canberra.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.
- Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.
- M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegrefe, Nouha Dziri, Khyathi Chandu, Jack Hessel, and 1 others. 2024. The art of saying no: Contextual noncompliance in language models. *Advances in Neural Information Processing Systems*, 37:49706–49748.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.
- Monojit Choudhury, Shivam Chauhan, Rocktim Jyoti Das, Dhruv Sahnan, Xudong Han, Haonan Li, Aaryamonvikram Singh, Alok Anil Jadhav, Utkarsh Agarwal, Mukund Choudhary, and 1 others. 2025. Llama-3-nanda-10b-chat: An open generative large language model for hindi. *arXiv preprint arXiv:2504.06011*.
- Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Kareem Darwish. 2013. Arabizi detection and conversion to arabic. *arXiv preprint arXiv:1306.6755*.
- Yuntian Deng, Wenting Zhao, Jack Hessel, Xiang Ren, Claire Cardie, and Yejin Choi. 2024. Wildvis: Open source visualizer for million-scale chat logs in the wild. *arXiv preprint arXiv:2409.03753*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- Mahmoud El-Haj. 2020. [Habibi - a multi dialect multi national Arabic song lyrics corpus](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.
- Mohamed Atta Faheem, Khaled Tawfik Wassif, Hanaa Bayomi, and Sherif Mahdy Abdou. 2024. Improving neural machine translation for low resource languages through non-parallel corpora: a case study of egyptian dialect to modern standard arabic translation. *Scientific Reports*, 14(1):2265.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [A framework for few-shot language model evaluation](#).
- Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and

- Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *arXiv preprint arXiv:2406.18495*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2024. [AceGPT, localizing large language models in Arabic](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.
- Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, and 1 others. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.
- 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, and 1 others. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Miresghallah, Ximing Lu, Maarten Sap, Yejin Choi, and 1 others. 2024b. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Advances in Neural Information Processing Systems*, 37:47094–47165.
- Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. [GlotLID: Language identification for low-resource languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218, Singapore. Association for Computational Linguistics.
- Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36:47669–47681.
- Fajri Koto, Rituraj Joshi, Nurdaulet Mukhituly, Yuxia Wang, Zhuohan Xie, Rahul Pal, Daniil Orel, Parvez Mullah, Diana Turmakhan, Maiya Goloburda, and 1 others. 2025. Llama-3.1-sherkala-8b-chat: An open large language model for kazakh. *arXiv preprint arXiv:2503.01493*.
- Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Al-mubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. [ArabicMMLU: Assessing massive multitask language understanding in Arabic](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- Hongbo Li, Sen Lin, Lingjie Duan, Yingbin Liang, and Ness B Shroff. 2024a. Theory on mixture-of-experts in continual learning. *arXiv preprint arXiv:2406.16437*.
- Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024b. NuminaMath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9.
- Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.
- Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2024. A closer look into mixture-of-experts in large language models. *arXiv preprint arXiv:2406.18219*.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and 1 others. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.
- Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. WizardCoder: Empowering code large language models with evolver. *arXiv preprint arXiv:2306.08568*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.
- Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. 2025. Lost in the mix: Evaluating llm understanding of code-switched text. *arXiv preprint arXiv:2506.14012*.
- Basel Mousi, Nadir Durrani, Fatema Ahmad, Md. Arif Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. [AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218, Abu Dhabi, UAE. Association for Computational Linguistics.

- Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.
- Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all—adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.
- Faisal Qarah. 2024. Egybert: A large language model pretrained on egyptian dialect corpora. *arXiv preprint arXiv:2408.03524*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.
- Nathaniel R Robinson, Shahd Abdelmoneim, Kelly Marchisio, and Sebastian Ruder. 2024. Al-qasida: Analyzing llm quality and accuracy systematically in dialectal arabic. *arXiv preprint arXiv:2412.04193*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavathula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.
- Guokan Shang, Hadi Abdine, Yousef Khoubrane, Amr Mohamed, Yassine Abbahaddou, Sofiane Ennadir, Imane Momayiz, Xuguang Ren, Eric Moulines, Preslav Nakov, Michalis Vazirgiannis, and Eric Xing. 2025. Atlas-chat: Adapting large language models for low-resource Moroccan Arabic dialect. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 9–30, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Shivalika Singh, Freddie Vargus, Daniel D’souza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividias Mataciunas, Laura O’Mahony, Mike Zhang, Ramith Hetiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11521–11567, Bangkok, Thailand. Association for Computational Linguistics.
- Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, and 1 others. 2024. Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. *arXiv preprint arXiv:2403.07816*.
- Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*, pages 303–324.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.
- Mohammad Ali Yaghan. 2008. " arabizi": A contemporary style of arabic slang. *Design issues*, 24(2):39–52.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.
- Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024a. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*.
- Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024b. Wildchat: Im chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.
- Jianqing Zhu, Huang Huang, Zhihang Lin, Juhao Liang, Zhengyang Tang, Khalid Almubarak, Abdulmohsen Alharthik, Bang An, Juncai He, Xiangbo Wu, and 1 others. 2024. Second language (arabic) acquisition of llms via progressive vocabulary expansion. *arXiv preprint arXiv:2412.12310*.

A Instruction Data Templates

A.1 Machine Translation

user:	$\backslash n[source\ text]\backslash n:[target\ language]$ لل $[source\ language]$ من ال ترجملي من ال $[source\ language]$ لل $\backslash n[source\ text]\backslash n:[target\ language]$ ترجملي لل $\backslash n[source\ text]\backslash n:[target\ language]$
multi-turn conversations:	$[source\ text]\backslash n$: ترجم
assistant:	$[target\ text]$

A.2 Transliteration

user:	$\backslash n[source\ text]\backslash n:[target\ language]$ بال اكتبلي الكلام ده بال $[source\ language]$ ال حول من ال $\backslash n[source\ text]\backslash n:[target\ language]$ لل مكن تكتبلي بال $\backslash n[source\ text]\backslash n:[target\ language]$
multi-turn conversations:	$[source\ text]\backslash n$: وده كمان
assistant:	$[target\ text]$

B TÜLU-v2&3-mix and Translation

In this section, we discuss in detail the composition of the *TÜLU-v2&3-mix* dataset and the process of its translation into Egyptian Arabic (in Arabic and Latin scripts), highlighting the datasets utilized and the sampling strategies implemented. We further elucidate the format of the dataset and the methodology used in translating the dataset into Egyptian Arabic.

B.1 Composition of TÜLU-v2&3-mix

TÜLU-v2&3-mix integrates samples from the following datasets: CoCoNot²¹ (Brahman et al., 2024), FLAN v2²² (Longpre et al., 2023), No Robots²³, Evolved codealpaca²⁴ (Luo et al., 2023), NuminaMath CoT²⁵ (Li et al., 2024b), Tulu 3 Persona {MATH²⁶, GSM²⁷, Python²⁸, Algebra²⁹, IF³⁰}, WildGuardMix³¹ (Han et al., 2024), WildJailbreak³² (Jiang et al., 2024b), Aya Dataset³³ (Singh et al., 2024), WildChat³⁴ (Deng et al., 2024), Table-GPT³⁵ (Li et al., 2023), Open Assistant 1 (Köpf et al., 2023)³⁶, ShareGPT³⁷ (Chen et al., 2024), GPT4-Alpaca (Peng et al., 2023)³⁸, LIMA (Zhou et al., 2023)³⁹, WizardLM Evol Instruct (Xu et al., 2023)⁴⁰, and Open-Orca (Mukherjee et al., 2023)⁴¹. Additionally, the mixture comprises hard-coded instructions and a collection of science-related inquiries extracted from scientific documents. Table 3 describes each of these datasets and how the subset was sampled.

²¹<https://hf.co/datasets/allenai/coconot>

²²https://hf.co/datasets/ai2-adapt-dev/flan_v2_converted

²³https://hf.co/datasets/HuggingFaceH4/no_robots

²⁴<https://hf.co/datasets/theblackcat102/evol-codealpaca-v1>

²⁵<https://hf.co/datasets/AI-MO/NuminaMath-TIR>

²⁶<https://hf.co/datasets/allenai/tulu-3-sft-personas-math>

²⁷<https://hf.co/datasets/allenai/tulu-3-sft-personas-math-grade>

²⁸<https://hf.co/datasets/allenai/tulu-3-sft-personas-code>

²⁹<https://hf.co/datasets/allenai/tulu-3-sft-personas-algebra>

³⁰<https://hf.co/datasets/allenai/tulu-3-sft-personas-instruction-following>

³¹<https://hf.co/datasets/allenai/wildguardmix>

³²<https://hf.co/datasets/allenai/wildjailbreak>

³³https://hf.co/datasets/CohereForAI/aya_dataset

³⁴<https://hf.co/datasets/allenai/WildChat-1M>

³⁵<https://hf.co/datasets/LipengCS/Table-GPT>

³⁶<https://hf.co/datasets/OpenAssistant/oasst1>

³⁷https://hf.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered

³⁸<https://github.com/Instruction-Tuning-with-GPT-4/GPT-4-LLM#data-release>

³⁹<https://hf.co/datasets/GAIR/lima>

⁴⁰https://hf.co/datasets/WizardLMTeam/WizardLM_evol_instruct_V2_196k

⁴¹<https://hf.co/datasets/Open-Orca/OpenOrca>

Dataset	Description	Number of Samples
CoCoNot	Improving the safety and reliability of chat-based language models by mitigating non-compliance in real-world scenarios.	10,983
FLAN	A collection of datasets covering tasks including question answering, summarization, and translation.	189,982 deduplicated
No Robots	Instructions and demonstrations, meticulously crafted by human annotators under various tasks.	9,500
Evolved codealpaca	Coding instructions data generated by gpt-4 models.	107,276
NuminaMath CoT	Math problems with numerical outputs and Tool-integrated Reasoning Agent (TORA)-like reasoning paths.	64,312
Tulu 3 MATH	Synthetic instructions answering complex math problems.	149,960
Tulu 3 GSM	Synthetic instructions simulating grade school math problems.	49,980
Tulu 3 Python	Synthetic instructions related to coding in Python.	34,999
Tulu 3 Algebra	Synthetically created instructions to answer algebra problems.	20,000
Tulu 3 IF	Synthetic instructions improving the model’s capability to follow instructions precisely and to satisfy user constraints.	29,980
WildGuardMix	Instructions about disturbing or harmful or interactions.	50,000
WildJailbreak	Synthetic safety-training dataset encompassing both harmful requests and adversarial jailbreaks examples.	50,000
Aya Dataset	A collection of human-annotated prompt-completion pairs.	100,000
WildChat	Introduced in Section 3.2.2	100,000 deduplicated
Table-GPT	Table-related tasks.	5,000
Open Assistant 1	A set of assistant-style conversations annotated by humans.	7,132
ShareGPT	User-shared conversations with ChatGPT and GPT-4.	114,046
GPT4-Alpaca	GPT-4 generated responses to prompts from Alpaca.	20,000
LIMA	Meticulously curated data to ensure high quality and accuracy.	1,030
WizardLM	Automatically evolving instruction datasets to enhance their complexity and diversity.	30,000
Open-Orca	Augmented FLAN data with additional explanations.	30,000
Science & SciRIFF	Scientific documents understanding tasks.	17,544
Hardcoded	Prompts related to the model’s identity and/or creators.	14 samples repeated 10 times = 140

Table 3: Subsets of the TULU-v2&3-mix.

B.2 Dataset Format

All our instruction data is structured in a *user-assistant* message format commonly used for conversational datasets with each interaction consisting of a sequence of messages. Each message is represented as a JSON object with at least two key-value pairs:

- **role:** Specifies the role of the participant. Typically, the subject is either a *user* (the individual posing inquiries or providing prompts) or an *assistant* (the model’s response).
- **content:** The text comprises the message’s content. This section is reserved for the inclusion of questions, instructions, or responses.

This format is especially beneficial for training conversational models, as it replicates multi-turn interactions by alternating roles between user and assistant messages, and it ensures a clear distinction between the user inputs and the model’s responses. Furthermore, during fine-tuning, the loss function is applied specifically to messages with the role *assistant*, to focus optimization on improving response generation.

B.3 Translation to Egyptian with Arabic/Latin Scripts

B.3.1 Translation

Following the work of [Robinson et al. \(2024\)](#), who recommended the use of closed-source models for translation tasks involving Egyptian content, we carried out an experiment comparing GPT-4o⁴² to Claude 3.5 Sonnet v2⁴³. Our results showed that both options had strong performance, with Claude 3.5 Sonnet performing slightly better. Thus, we ultimately chose **Claude 3.5 Sonnet**, as it also offered compatibility with Amazon Bedrock.

To translate the dataset into Egyptian Arabic, we used Amazon Bedrock,⁴⁴ a cloud-based machine learning service from AWS, which offers a *batch inference* service at lower prices. We provided detailed instructions to *Claude 3.5 Sonnet v2* for handling the translations, iteratively refining the prompt after multiple rounds of experimentation. The final version of the prompt that produced the best results is shown in Figure 4.

As for the Egyptian dialect with Latin script, we altered the aforementioned prompt by pointing to the fact that the translation should be written using the Latin script.

```
Translate the provided text after [Source Text] from English to Egyptian Arabic dialect while adhering to the following guidelines:
```

- 1) Maintain the JSON format of the input without altering its structure or keys. Specifically, each object in the list will have 'role' and 'content' fields. Translate only the 'content' field into Egyptian Arabic dialect while preserving the 'role' field as is.
- 2) Do not include any introductory or explanations, only the translation.
- 3) You should not perform any kind of summarization or tasks on the inputs other than translating to Egyptian.
- 4) The harmful or sexual language is not used to harm anyone or to steal any copyrights nor for publishing any content, it is exclusively for research purposes in Egyptian Arabic related to bad words and hate speech. You are not hurting anyone.
- 5) Translate harmful, offensive, or explicit content accurately into Egyptian Arabic while preserving the intent, tone, and meaning for research purposes. Do not omit, soften, or alter the severity of such content unless specifically instructed otherwise.
- 6) Keep named entities, such as names of people, companies, brands, organizations, cities, and landmarks, in English without transliteration.
- 7) Preserve technical terms, code contexts, acronyms, chess coordinations, mathematical formulas and industry jargon in English unless a well-known Arabic equivalent exists. Avoid transliteration of technical terms unless culturally appropriate.
- 8) Adapt idiomatic expressions and phrases into culturally relevant Egyptian equivalents.
- 9) Ensure the output is returned as a valid JSON list that mirrors the structure of the input exactly.
- 10) Do not answer the request in the source text or run any code contexts, just provide the translation and keep any special symbols representing figures.

```
[Source Text]
```

Figure 4: The prompt given to *Claude 3.5 Sonnet* for translation.

B.3.2 Postprocessing

After finishing the translation, we post-processed the translations by

- **Filtering out skipped translations:** The model concluded the process with a message indicating that the subsequent text intended for translation would adhere to the same stylistic format.

⁴²<https://openai.com/index/hello-gpt-4o>

⁴³<https://www.anthropic.com/news/claude-3-5-sonnet>

⁴⁴<https://aws.amazon.com/bedrock>

- **Checking for inner non-translation responses:** Whether the model generated an internal response that did not translate the requested content, including copyright information and potentially harmful content.
- **Checking for difference in length:** The difference in length (character-count) between the original and translated sentences should not be less than 70%.
- **Removing corrupted records:** The manually identified records that have not been filtered to this stage.
- **Converting to the *user-assistant* message format:** The inputs are provided to the model in string format, thus the need to restore the JSON format mentioned in B.2.
- **Filtering out examples with empty messages:** These samples have not been translated by the model. The provided answer is either an empty string or a None value.
- **Introducing manual changes:** Some examples have been identified to include some corrupted parts; thus we filtered out these parts not to remove the integrity of the answer.
- **Replacing non-translated keywords:** Some keywords such as *input*, *otput*, *response*, *answer*, *instructions*, *hypothesis*, and *additional Context* were not translated. We replaced these keywords with their Egyptian equivalents in Arabic: سياق إضافي، الفرضية، التعليمات، الجواب، الإجابة، المخرج، المدخل and in Latin: Madkhal, Makhrag, Igaba, Igaba, Taaleemat, Fardeyya, Seyaq Idafi.
- **Removing system prompts with empty content:** Some of the provided examples include a *system* role with empty content. Thus, this role is removed while maintaining the rest of the conversation.
- **Checking for the consistency of the *user-assistant* flow:** This is performed by checking for the interchanged turns between the *user* and the *assistant*.
- **Removing samples with excessive English content (not applied for Latin script):** We used the fastText⁴⁵ Language Identification model to detect samples where the predicted language was not Arabic. Since the model does not differentiate dialects, Egyptian is recognized as Arabic due to its use of Arabic script. We removed examples where the predicted language was not Arabic or where Arabic was predicted with a confidence level below 80%.
- **Removing indirect translation prompts:** Despite the fact that the translation tasks were removed in the preprocessing part (to prevent duplicated sentences), we performed a second check for some indirect translation tasks that need to be removed.

C Additional Details

C.1 Arabic-to-Latin Script Transliteration Template

The prompt can be found in Figure 5.

C.2 Pre-training Datasets

Egyptian Forums Corpus-mini (EFC-mini) (Qarah, 2024) comprises approximately 201M words and 11M sentences drawn from widely used Egyptian online forums. The corpus encompasses a broad range of discussion domains, including sports, health, politics, religion, travel, and technology. This thematic diversity captures substantial linguistic variation and provides a representative sample of authentic, user-generated content in Egyptian Arabic, particularly as expressed in informal, web-based discourse.

Egyptian Datasets Collection (EDC).⁴⁶ is a large-scale compilation of over 2.5M Egyptian Arabic text entries (approximately 62M words) sourced from a diverse array of platforms, including social media, online commentary, lyrics, and web forums, reflecting a wide spectrum of contemporary Egyptian discourse across informal and formal registers. The datasets are curated to support natural language processing tasks such as sentiment analysis, topic modeling, and dialect identification.

⁴⁵<https://hf.co/facebook/fasttext-language-identification>

⁴⁶<https://github.com/Mostafanofal453/2.5-Million-Rows-Egyptian-Datasets-Collection>

```

Transliterate the source Egyptian Arabic (Masri) text to Egyptian Latin Script (
Franco-Arab) while following these guidelines:

- Use the Egyptian Latin Script (Franco-Arab) for the transliteration.
- Do not include the source text in the transliteration.
- If the source text is missing line breaks (\n), add them in the transliteration.
- Don't include an introduction or a summary.
- If a word is written already in Latin script, do not transliterate it.
- Return only the transliterated Franco-Arab Egyptian text.

### Example:
Source Text:
{one-shot Arabic script text}

Transliterated Text:
{one-shot Latin script text}

[Source Text]
{arabic_script_text}

[Egyptian Latin Script (Franco-Arab) Text]

```

Figure 5: The prompt given to *Claude 3 Haiku* for Arabic to Latin-script transliteration.

Egyptian Wikipedia Dump.⁴⁷ We used the September 2024 snapshot of the Egyptian Arabic Wikipedia, which contains over 1.6M pages and approximately 80M words.

Arabic Dialects Dataset (ADD).⁴⁸ It is a multi-dialect corpus designed to support dialectal Arabic NLP research, and covers five major varieties. We used the Egyptian subset comprising approximately 115K words.

FineWeb-2. We selected the Egyptian Arabic portion of the FineWeb-2 dataset (Penedo et al., 2025), which comprises 1.4M documents and 439M words.

Habibi is a multi-Dialect corpus of Arabic song lyrics containing over 30K songs from 18 Arab countries and covering six major dialects (El-Haj, 2020). For our purposes, we extracted the Egyptian subset, which consists of approximately 981K words.

Fatakat.⁴⁹ We web-scraped a total of 220 posts, comprising approximately 65K words, from the Fatakat forum, a popular Egyptian online community focused on topics such as family life, cooking, health, and social advice. The content reflects informal, user-generated discussions written predominantly in Egyptian Arabic.

C.3 Instruction-tuning Datasets

EGY_MSA_Translation⁵⁰. In order to improve neural machine translation for low-resource languages, Faheem et al. (2024) conducted a case study of the Egyptian dialect to Modern Standard Arabic translation. In their work, they assembled one of two datasets as a parallel corpus of Egyptian Arabic to standard Arabic. For the Egyptian Arabic dialect, they focused on colloquial sentences from social networking sites such as *Fatakat*, *Facebook* and *Twitter* with each sentence spanning between five and 50 words. Then, they translated 40,000 good quality samples into Modern Arabic using social communication methods, some friends, and Arabic language teachers.

ArzEn-MultiGenre⁵¹. ArzEn-MultiGenre (Al-Sabbagh, 2024) is a rigorously curated parallel dataset encompassing a heterogeneous collection of Egyptian Arabic texts. The dataset contains around 26,000 sentences of three textual genres: song lyrics, novels, and TV show subtitles. These samples were trans-

⁴⁷<https://dumps.wikimedia.org/arzwiki/>

⁴⁸<https://elhaj.uk/corpora.html>

⁴⁹<https://forums.fatakat.net>

⁵⁰https://github.com/mohamedatta93/EGY_MSA_Translation/tree/main/data

⁵¹<https://hf.co/datasets/HeshamHaroon/ArzEn-MultiGenre>

lated and aligned with their English counterparts by professional translators who possess a professional training in translation and a deep understanding of cultural differences between both audiences.

Egyption_2_English⁵². This dataset consists of around 22,000 everyday sentences aligned with their English counterparts. No information has been provided regarding the source of the Egyptian Arabic samples or the method used to perform the translation task. However, the native speakers confirmed the good quality of the translation.

Oasst2-9k-translation⁵³. In this dataset, 9,500 English-based sentences have been collected from the Open Assistant Conversations Dataset Release 2 (OASST2)⁵⁴. In the following, these samples have been translated and aligned with their Egyptian Arabic and Modern Arabic counterparts with the mean of *GPT-4o*. According to the work by [Robinson et al. \(2024\)](#), the closed-source *GPT-4o* model has been recommended for Egyptian Arabic dialect, as it has surpassed its alternatives on sentences sourced from the Basic Traveling Expression Corpus (BTEC) ([Takezawa et al., 2007](#)), which consists of common spoken expressions used in daily communication and manually translated to 26 Arabic varieties, and FLORES-200 ([Costa-Jussà et al., 2022](#)), a machine translation evaluation benchmark of 1,012 sentences in 204 language varieties.

C.4 DPO Off-policy Data Generation

To identify samples exhibiting over code-switching, we filtered the SFT dataset to exclude any instructions related to coding, mathematics, or safety instructions. From the remaining subset, we selected instances that met two conditions: (1) the instruction contained at least one English word, and (2) less than 35% of the total words in the instruction were written in English. This filtering ensured the identification of predominantly Arabic prompts with unnatural or unnecessary code-switching, which were then passed to Claude for correction, using the prompt shown in Figure 6.

```
You are an Egyptian who is a native proficient in Egyptian Arabic using everyday,
casual Egyptian Arabic.

You'll get a question written like Egyptians naturally ask each other. Just answer
it like a native Egyptian.

Your response must follow these rules:
- It must be written entirely in Egyptian Arabic using Arabic script.
- Do not use any Modern Standard Arabic (MSA), formal expressions, or literary
  language.
- Use common Egyptian slang, idioms, jokes, and references to daily life (like food,
  traffic, weather, mobile data, TV shows, school, work, etc.).
- If a word has no real Egyptian Arabic equivalent, especially technical or internet
  -related words like "code", "programming", "WiFi", "scroll", "subscribe", "
  remote", "meeting", "app", "USB", etc., write that word in **English script**,
  exactly how it's commonly said in Egypt. Do not translate or rephrase it.
- Write the answer in a normal text and not using markdown syntax.
- Don't write introductions, explanations, or anything extra, just give the direct
  answer like you're chatting with someone.

Now, answer the following question in Egyptian Arabic:
{prompt}
```

Figure 6: The prompt given to Claude for off-policy data generation.

⁵²https://hf.co/datasets/Abdalrahmankamel/Egyption_2_English

⁵³<https://hf.co/datasets/ahmedsamirio/oasst2-9k-translation>

⁵⁴<https://hf.co/datasets/OpenAssistant/oasst2>