Fine-Tuning and Agentic Workflows for Lebanese Arabizi Understanding and Generation

Abbass Abdullah, Amir Amine, Hassan Bazzi, Mohammad Khalife, Joseph Azar, and Raphaël Couturier
Université Marie et Louis Pasteur, CNRS
FEMTO-ST Institute, Belfort, France
{abbass.abdullah, amir.amine, hassan.bazzi, mohammad.khalife}@edu.univ-fcomte.fr,
{joseph.azar, raphael.couturier}@umlp.fr

***Abstract*—Arabizi—the Latin-script rendering of spoken Arabic using letters, numerals, and punctuation—dominates informal digital communication across the Arab world. Yet most modern Arabic NLP systems assume Arabic-script input, leaving Arabizi largely unsupported. Lebanese Arabizi is particularly challenging due to extreme spelling variability, numerals-as-phonemes (e.g., 3, 7, 8), and systematic trilingual code-switching with English and French.**

This work presents the first unified study of Lebanese Arabizi as a conversational interface for large language models. We curate a native-speaker dataset from WhatsApp conversations and develop an evaluation protocol that captures style fidelity, instruction-following, code-switching, and orthographic variability. We compare six modeling approaches: parameter-efficient fine-tuning of open-weight LLMs, a modular Arabizi→Arabic→Arabizi pipeline, managed proprietary fine-tuning, large zero-shot models, and an agentic RAG workflow.

Our experiments show that Arabizi-specific fine-tuning substantially improves human-likeness and dialect authenticity over zero-shot baselines. Model scale alone is insufficient for robust Lebanese Arabizi handling. The translational pipeline preserves semantic content better but often degrades stylistic naturalness, while agentic RAG improves knowledge-intensive responses when paired with a fine-tuned generator. These results highlight the need for dialect-aware adaptation and establish a reproducible foundation for Lebanese Arabizi conversational modeling.

***Index Terms*—Arabizi, Lebanese Arabic, dialectal NLP, low-resource languages, large language models, parameter-efficient fine-tuning, retrieval-augmented generation, agentic workflows, code-switching, conversational AI**

I. Introduction

Arabizi is a non-standard Latin-script representation of spoken Arabic that combines letters, numerals, and punctuation to encode Arabic phonemes.[1] It appears everywhere—SMS, WhatsApp, Twitter, Instagram. Despite this prevalence, most Arabic NLP resources and models assume Arabic-script input. They either ignore Arabizi entirely or treat it as noise to be transliterated away. Prior research has focused mainly on transliteration or black-box translation into Arabic or English, often targeting North African dialects. Conversational assistant behavior has received little attention.

Lebanese Arabizi introduces several layers of complexity. There is no orthographic standard—writers spell words phonetically, leading to substantial variation even for common expressions. Online communication frequently mixes Arabizi, English, and French within single utterances, often with pragmatic intent. Numerals like 2, 3, 5, 7, and 8 encode Arabic consonants absent from the Latin alphabet, but the mappings vary across writers. Combined with missing diacritics and inconsistent vowels, this creates high phonological ambiguity where identical surface forms can mean different things depending on context.

The data situation makes this worse. Large corpora exist for Arabic and several dialects in Arabic script, but Lebanese Arabizi remains severely under-resourced. Existing Arabizi datasets are small, domain-specific, and concentrated on non-Levantine dialects. Annotation practices vary widely—sentiment labels, transliteration, dialect tagging—without consistent treatment of numerals or code-switching. Until recently, there were no shared benchmarks for Arabizi. Emerging ones emphasize translation rather than conversational evaluation, which makes reliable cross-model comparison difficult.

Current approaches struggle under these conditions. Arabic and dialectal models are pretrained on Arabic-script text, while multilingual LLMs observe Arabizi only sparsely during pretraining. This leads to brittle zero-shot performance. The small, noisy nature of available Arabizi data makes direct fine-tuning prone to overfitting on narrow domains or idiosyncratic spelling. As a result, prior work has remained fragmented across isolated tasks like transliteration or sentiment analysis, with limited exploration of conversational modeling or systematic comparison of adaptation strategies.

Agentic workflows offer an alternative by decomposing the problem into specialized components—retrieval for factual grounding, generation for dialectal fluency. This is particularly appealing for Arabizi, where data scarcity makes it hard to train a single monolithic model that handles reasoning, factuality, and non-standard orthography jointly. Unlike prior work that treats Arabizi primarily as a translation problem, we explore it as a conversational interface. We directly compare parameter-efficient fine-tuning, modular translation pipelines, large zero-shot models, and agentic retrieval-augmented workflows under a unified Lebanese Arabizi evaluation protocol.

Why does this matter? Lebanese Arabizi is used daily by millions of speakers. Current AI systems remain optimized for English and Arabic script, forcing users to switch languages to interact effectively. Addressing Lebanese Arabizi is essential

[1] We use "Arabizi" to denote Latin-script writing of Arabic dialects, and "Lebanese Arabizi" for the Lebanese variety.

for building inclusive, user-aligned language technologies that support natural communication.

We make four contributions:

1) We curate Lebanese Arabizi conversational data and define an evaluation protocol capturing style fidelity, instruction-following, code-switching, and orthographic variability.
2) We perform parameter-efficient fine-tuning (LoRA/QLoRA) across multiple open-weight LLM families and scales, evaluating their effectiveness on Lebanese Arabizi tasks.
3) We implement a modular translational workflow and compare it against direct Arabizi generation under identical test conditions, analyzing trade-offs between pivoted translation and end-to-end modeling.
4) We evaluate large-scale models and agentic retrieval-augmented pipelines, analyzing how model scale and contextual grounding affect knowledge-intensive Arabizi queries while preserving stylistic fidelity.

II. Literature Review

A. Datasets for Dialectal and Lebanese Arabic NLP

1) Existing Arabic-Script Resources: Large-scale pre-trained resources like MARBERT and CAMeLBERT have significantly advanced Arabic dialect modeling, including dialect identification and morphosyntactic tagging [1]–[3]. Related work on dialectal benchmarks and evaluation protocols demonstrates the effectiveness of Arabic-script pretraining for downstream tasks [4], [5].

However, these datasets and models operate exclusively in Arabic script. They provide no coverage for Arabizi, which relies on Latin characters and numerals. This limits their applicability for modeling informal Lebanese digital communication.

2) The Need for Native Data: Most publicly available Lebanese or Levantine datasets (e.g., MADAR, OpenSubtitles) are either translated or crowdsourced by non-native speakers [4], [6]. They often fail to capture colloquial idioms, pragmatic markers, and code-switching patterns that characterize real usage.

Recent work on LebEval shows that small, carefully curated datasets authored by native speakers can outperform much larger but noisier corpora [7]. Similar findings appear in other low-resource dialect adaptation studies [5]. This motivates our focus on collecting naturally occurring WhatsApp conversations rather than relying on synthetic or translated data.

B. Pre-trained Models and Adaptation

1) Arabic Encoders (MARBERT, CAMeLBERT): Dialect-specialized encoders like MARBERT consistently outperform multilingual models on Arabic dialect tasks [1]. CAMeLBERT demonstrates strong performance on morphosyntactic tagging across Arabic varieties [2].

Despite their success, these models are trained exclusively on Arabic-script data. They cannot handle Arabizi without a transliteration or translation step, which introduces noise and stylistic distortion [3].

2) Instruction-Tuned Models (Aya-23): Instruction-tuned multilingual models like Aya23-8B have shown promising results for Lebanese translation through adapter-based fine-tuning [7]. Similar trends appear in other instruction-following LLMs adapted for Arabic tasks [8].

However, existing adaptations operate solely in Arabic script. We extend this line of research by applying parameter-efficient adaptation directly to Arabizi, enabling native generation without an intermediate normalization or translation stage.

III. Methodology

A. Lebanese Arabizi Dataset

1) Motivation and data scarcity: Lebanese Arabizi lacks large, standardized datasets. Existing Arabic-script resources for other dialects do not transfer due to Lebanese-specific vocabulary and trilingual Arabic/English/French mixing. This scarcity motivates a dataset strategy centered on careful collection, filtering, and style-faithful curation.

2) Data Sources and Preparation: Our primary source is WhatsApp group chats, which provide realistic conversational structure and code-switching. We also explored YouTube comments and Twitter/X, but these proved noisy—Arabic-script dominance, non-Lebanese dialects, emoji-only content, and low yield after filtering.

We parse WhatsApp exports into structured messages (timestamp, anonymized speaker ID, text) with light normalization (whitespace cleanup) while preserving Arabizi variability like numerals-as-phonemes and elongations. We remove messages that are Arabic-only, English-only, emoji/reaction-only, numbers-only, or contain links/media placeholders. After filtering, the unlabeled corpus contains hundreds of thousands of messages, scalable to millions with additional exports.

From these chats, we derive three dataset formats. For translation-based modeling (M2), we align Arabizi messages with Arabic-script equivalents, translating Arabizi-first messages into Arabic with light manual correction. This yields up to $\sim$10k Arabizi$\rightarrow$Arabic pairs.

For instruction tuning (M1, M3), direct chat-to-pair conversion proved unreliable. We use LLM-assisted generation followed by strict manual Lebanese-style correction, ensuring natural vocabulary, pragmatics, and code-switching. This produces 4,520 high-quality instruction–response samples.

3) Dataset components: The dataset is intentionally multi-format to support different training paradigms (M1-M5). Table I summarizes dataset components and usage.

- **D1: Unlabeled RAW chat corpus.** Filtered WhatsApp messages.
- **D2: Arabizi$\leftrightarrow$Arabic parallel data.** Used for translational pipelines, enabling reasoning in Arabic script while preserving Lebanese meaning.
- **D3: Instruction–response pairs.** High-quality Arabizi input–output pairs for supervised fine-tuning.

TABLE I
SUMMARY OF LEBANESE ARABIZI DATASET COMPONENTS AND USAGE.

Component	Format	Purpose	Scale
D1: RAW WhatsApp corpus	Unlabeled messages	Retrieval store for M4 (conversational context)	$\sim 10^5$–10^6 msgs
D2: Arabizi→Arabic	Parallel pairs	Translation module for M2 using Arabic as pivot	up to $\sim$10k pairs
D3: Arabizi I/O SFT	Input–output pairs	Supervised fine-tuning for M1 and compact SFT references (M3)	4,520 pairs

B. Methods Overview

We evaluate six methods under a unified Lebanese Arabizi evaluation protocol:

- **M1: Direct Arabizi Fine-tuning** (single-model Arabizi→Arabizi generation via LoRA/QLoRA).
- **M2: Translational Pipeline** (Arabizi→Arabic reasoning→Arabizi), where translation modules are fine-tuned and reasoning is performed in Arabic.
- **M3: Managed Proprietary Fine-tuning** (Gemini Flash 2.5 via Vertex AI) as a cloud-hosted reference.
- **M4: Large Open-weight Instruction Model (Gemma 27B)** used zero-shot without Arabizi-specific fine-tuning.
- **M5: Large Open-weight Instruction Model (LLaMA 70B)** used zero-shot without Arabizi-specific fine-tuning.
- **M6: Agentic Retrieval-Augmented Generation (RAG) with a Fine-tuned Arabizi Generator**.

*1) **M1: Direct Arabizi Response Generation**:* In M1, a single instruction-tuned LLM is adapted to accept Lebanese Arabizi input and generate Arabizi responses directly. This end-to-end setting requires the model to handle both non-standard Arabizi orthography and response generation. We apply LoRA/QLoRA rather than full fine-tuning to reduce computational cost.

We fine-tune Gemma-2 12B using LoRA adapters on an instruction-tuned backbone. We leverage ShareGPT-format Arabizi conversations alongside our curated dataset. Training instances follow an instruction-style conversational format, with assistant outputs written in Arabizi while preserving natural English/French code-switching. This approach requires only a single model call at inference but typically demands higher-quality Arabizi supervision to ensure stable instruction following.

*2) **M2: Translational Model (Three-Stage Pipeline)**:* We evaluate a modular translational pipeline that preserves Arabizi as the user interface while shifting answer generation into Arabic:

$$\hat{x}_{ar} = T_{\text{Az2Ar}}(x_{az}), \tag{1}$$

$$\hat{y}_{ar} = G_{\text{Ar}}(\hat{x}_{ar}), \tag{2}$$

$$\hat{y}_{az} = T_{\text{Ar2Az}}(\hat{y}_{ar}). \tag{3}$$

where x_{az} is the input Arabizi text, $\hat{x}_{ar}$ is the translated Arabic text, $\hat{y}_{ar}$ is the generated Arabic response, and $\hat{y}_{az}$ is the final Arabizi output. T_{Az2Ar} and T_{Ar2Az} are respectively the Arabizi-to-Arabic and Arabic-to-Arabizi translation modules, while G_{Ar} is the Arabic response generator. We use `Gemma-2 12B` out of the box as the Arabic answering model G_{Ar}, keeping it frozen to preserve its native Arabic capabilities. The Arabizi–Arabic and Arabic–Arabizi translation modules are fine-tuned with LoRA adapters. Strict system prompts enforce single-script output at each stage. Translation training relies on a mixture of manually curated and filtered synthetic Arabizi–Arabic pairs.

At inference, all stages share a unified decoding configuration. Because outputs cascade across stages, translation errors may propagate. We therefore analyze both stage-level fidelity and end-to-end Arabizi output quality.

C. *M3: Vertex AI Fine-tuning (Gemini Flash 2.5)*

We fine-tune Gemini Flash 2.5 using Google Vertex AI's managed supervised fine-tuning pipeline on a compact single-turn Arabizi dataset. This provides a proprietary reference point for instruction-following quality under a controlled cloud setting.

D. *M4–M5: Large Zero-shot Baselines*

We evaluate Gemma 27B Instruct and LLaMA 70B Chat/Instruct in a zero-shot setting. We prompt them directly in Lebanese Arabizi without fine-tuning, translation, or retrieval. These baselines test whether model scale alone enables robust handling of Arabizi orthography and dialectal variation.

E. *M6: Agentic Retrieval-Augmented Generation (RAG) with GPT-4o-mini*

We design a lightweight agentic RAG workflow where GPT-4o-mini serves as the base generator and retrieval provides external grounding. The retrieval corpus consists of curated Lebanese Arabizi books, which offer longer and more stylistically consistent context than chat logs.

Given an input query, the agent determines whether grounding is required, retrieves relevant passages, compresses them into a concise context, and generates the final Lebanese Arabizi response conditioned on the retrieved content.

IV. EXPERIMENTAL SETUP

A. *Vertex AI*

We fine-tune Gemini Flash 2.5 (chat/instruction) using Google Cloud Vertex AI supervised fine-tuning.

B. *Kaggle*

All experiments run in Kaggle Notebooks equipped with an NVIDIA T4 GPU (16 GB VRAM). Using 4-bit quantization with LoRA, this setup enables fine-tuning models with up to approximately 12B parameters. Models M1 and M2 are fine-tuned in this environment. We rely on the `Unsloth` library for efficient training.

C. *RunPod*

For higher-memory experiments, we use a rented RunPod server with two NVIDIA A6000 GPUs (76 GB total VRAM). This supports fine-tuning and inference beyond Kaggle limitations while providing full control over the software stack. Models M4 and M5 are trained and evaluated here. All experiments reuse the same HuggingFace/PEFT-based stack as Kaggle runs to ensure architectural consistency.

V. Experiments

A. Research Questions

Our experiments address four questions:

- **RQ1: Fine-tuned model performance.** How well does a fine-tuned model perform on Lebanese Arabizi data?
- **RQ2: Pipeline vs direct generation.** Does the translational pipeline (M2) improve meaning preservation relative to direct fine-tuned generation (M1)?
- **RQ3: Effect of model scale.** How does increasing model size (M4, M5) affect performance on Lebanese Arabizi tasks compared to a smaller fine-tuned model (M1)?
- **RQ4: Fine-tuned models vs. agentic workflow.** How do our fine-tuned models (M1-M5) compare to the agentic RAG workflow (M6) on knowledge-intensive queries while preserving Arabizi style?

B. Evaluation Protocol

All methods are evaluated under a unified protocol combining (i) automatic overlap metrics (BLEU/ROUGE) on a held-out Arabizi test set and (ii) qualitative human evaluation with native Lebanese speakers assessing dialect authenticity, Arabizi naturalness, and perceived human-likeness.

1) Automatic Metrics: For each test prompt and reference Arabizi response, we compute n-gram overlap using:

- **BLEU-4**: corpus-level, precision-oriented overlap.
- **ROUGE-1/2/L**: recall-oriented overlap.

We compute scores under two conditions: (i) raw text, and (ii) light-normalized text (lowercasing, whitespace and punctuation normalization, controlled collapsing of character elongations, while preserving numerals). Reporting both distinguishes semantic divergence from orthographic variation.

2) Human Evaluation: We conduct a subjective, blinded evaluation performed by the authors, all native Lebanese speakers. We assess whether model outputs resemble natural Lebanese Arabizi. Evaluation focuses on: (i) perceived human-likeness (AI vs Lebanese writer) and (ii) overall linguistic quality and dialect authenticity.

VI. Results

This section reports quantitative and qualitative results across all evaluated methods (M1–M6).

Model	BLEU-4 (Precision)	ROUGE-L (Recall)
M1	5.20	8.50
M2	5.17	14.40
M3	N/A†	N/A†
M4	N/A†	N/A†
M5	N/A†	N/A†

TABLE II
BLEU-4 and ROUGE-L scores for Lebanese Arabizi.

†Not evaluated with BLEU/ROUGE due to unreliability for Lebanese Arabizi.

A. Quantitative Results: BLEU and ROUGE

We report BLEU-4 and ROUGE-L scores on a held-out Lebanese Arabizi test set for the fine-tuned generator (M1) and the translation-based pipeline (M2). Results appear in Table II.

Scores are computed under both raw text and light normalization (lowercasing, whitespace and punctuation normalization, controlled elongation collapsing) to partially reduce penalties from superficial spelling variation. Even under these conditions, absolute BLEU and ROUGE values remain low. M1 achieves BLEU-4=5.20 and ROUGE-L=8.50, while M2 achieves BLEU-4=5.17 and ROUGE-L=14.40. These scores permit limited relative comparison, but their magnitude reflects the high orthographic variability of Lebanese Arabizi rather than poor semantic quality.

BLEU and ROUGE are designed for standardized orthography and sentence-level translation. In Lebanese Arabizi, they systematically penalize spelling variation, paraphrasing, pragmatic rewording, and code-switching. They often assign near-zero overlap scores to fluent, human-like responses. For this reason, we do not apply BLEU or ROUGE to the remaining systems (M3–M5).

Consider the everyday greeting *kifak* ("how are you?"). It has no canonical spelling and appears in many variants: *kifak*, *keefak*, *kefak*, *kifakkk*, *kifk*. These often combine with elongation, capitalization, or code-switching (*keefak bro?*). This one-to-many mapping from meaning to form severely limits string-based evaluation reliability.

B. Qualitative Examples

Table III shows representative prompt–response examples. For each prompt, we include the reference and outputs from all models (M1-M6).

VII. Discussion

Parameter-efficient fine-tuning (M1) consistently improves Lebanese Arabizi fluency and stylistic consistency over zero-shot baselines. This holds across different base LLM families and model scales, including both small base checkpoints and large instruction-tuned models (M4–M5). The gains highlight the importance of dialect-specific adaptation.

Among all evaluated systems, the strongest overall qualitative performance came from M4. Given our dataset size and computational resources, M4 produced the most natural, coherent, and human-like Lebanese Arabizi outputs. This suggests

TABLE III
REPRESENTATIVE PROMPT–RESPONSE EXAMPLES ACROSS DIFFERENT SYSTEMS.

Prompt	System	Answer (Arabizi)
Sabah l 5er!	Reference	*Sabah l noor!*
	M1 (FT)	*3ala ra7*
	M2 (Pipeline)	*Sabah el 5ayr!*
	M3 (Gemini FT)	*Sabah el noor w el wared*
	M4 (Gemma 27B, zero-shot)	*Sabah l noor!*
	M5 (LLaMA 70B, zero-shot)	*sabah l 5er w 7elo*
	M6 (RAG)	*Sabah el 5eyr! Kif fini asa3dak?*
sho btensa7ne e7dar series?	Reference	*Fik tehdar The Office aw Peaky Blinders*
	M1 (FT)	*ehh ya3ne shouf hek*
	M2 (Pipeline)	*te7dar shi awe eh*
	M3 (Gemini FT)	*Ya ahla! Hasab enta shu bet-heb, bass fi ktir eshya helwe hasab el "mood"*
	M4 (Gemma 27B, zero-shot)	*hdar 'The Office', btensa7 fik tabe3a*
	M5 (LLaMA 70B, zero-shot)	*l 3am jdid l 2osba mni7 l 7eml.*
	M6 (RAG)	*W ana ba7eb el series, wenta?*
3nde exam bokra	Reference	*good luck! enta adda*
	M1 (FT)	*bokra exam w enta bt2alel*
	M2 (Pipeline)	*exam bokra ya3ni shayy*
	M3 (Gemini FT)	*Yalla bel tawfiq!*
	M4 (Gemma 27B, zero-shot)	*good luck habibte enta adda w 2dood*
	M5 (LLaMA 70B, zero-shot)	*Ya rab ma3ak!*
	M6 (RAG)	*aywa! good luck bel exam! w enta 3aref tedros?*
sho btensa7ne ekol	Reference	*shawarma*
	M1 (FT)	*akhar ma b2olak*
	M2 (Pipeline)	*eh ya3ne akel la t3a22idha*
	M3 (Gemini FT)	*Hayde kam fikra la aklet mofide la bokra*
	M4 (Gemma 27B, zero-shot)	*ma bense ekol abadan, 3am emza7*
	M5 (LLaMA 70B, zero-shot)	*ma3akne shawarma 3a ra7et ba3d ma2a*
	M6 (RAG)	*Shou 3am t2ol? Bade ekol shou?*

that sufficiently large instruction-tuned models can internalize useful dialectal patterns even with limited resources.

The translational pipeline (M2) reliably preserves semantic content but often reduces stylistic naturalness. Script conversion and cascading translation effects result in outputs that are correct yet less idiomatic.

Agentic retrieval-augmented generation (M6) shows clear potential, particularly for knowledge-intensive prompts. When paired with a fine-tuned generator and supplied with sufficient contextual grounding, M6 improves factual clarity while maintaining Lebanese Arabizi style.

VIII. GENERALIZATION AND TRANSFERABILITY

Although our experiments focus on Lebanese Arabizi, the proposed methods are not dialect- or language-specific.

Our approaches readily generalize to other Arabic dialects. Many dialects—Egyptian, Moroccan, Gulf Arabic—are better represented in online data than Lebanese Arabizi. Given such availability, our approaches can directly ingest existing datasets, convert them into the same instruction-following format, and apply parameter-efficient fine-tuning without architectural modification.

The RAG component in M6 is inherently dialect-agnostic. By retrieving dialect-specific resources (grammar references, usage guides, conversational examples) and adjusting the system prompt, the agent can operate as a different Arabizi dialect model rather than being limited to Lebanese conventions. This enables flexible control over dialectal style without retraining the base model.

The framework is not limited to Arabic. Both the fine-tuning procedure (M1–M5) and the RAG pipeline (M6) are language-independent. They can be applied to other low-resource or non-standardized languages by substituting the underlying datasets and retrieval corpus. The same mechanisms used to adapt across Arabic dialects naturally extend to cross-language adaptation.

Lebanese Arabizi represents a particularly challenging case due to the near absence of publicly available datasets and the lack of standardized orthography. The fact that our approach yields coherent and human-like outputs under these constraints suggests it should transfer effectively to languages and dialects with greater data availability and stronger written conventions.

IX. Conclusion

This paper investigated Lebanese Arabizi not as a peripheral noise pattern to be normalized, but as a first-class conversational medium for interacting with large language models. Through a unified dataset, evaluation protocol, and systematic comparison of six modeling strategies, we showed that parameter-efficient fine-tuning on native Lebanese Arabizi data yields consistent gains in fluency, stylistic coherence, and perceived human-likeness relative to zero-shot prompting. This holds even when compared against much larger instruction-tuned models, indicating that model scale alone does not substitute for dialect- and orthography-aware adaptation.

We also showed that a modular translational pipeline using Arabic as a reasoning pivot improves meaning preservation in many cases. However, it introduces cascading errors and stylistic artifacts that reduce Arabizi naturalness. In contrast, an agentic RAG workflow, when coupled with a fine-tuned Arabizi generator, enhances factual grounding on knowledge-intensive queries while maintaining dialectal style. Across all settings, traditional overlap metrics like BLEU and ROUGE remain poorly correlated with human judgments. This underscores the need for evaluation methodologies tailored to highly variable, non-standard orthographies.

Our results support a hybrid view of Arabizi modeling. Direct fine-tuning is essential for surface-form fluency and stylistic authenticity. Modular and agentic components provide complementary benefits for reasoning and knowledge access. Future work will explore larger and more diverse native-speaker Arabizi corpora, richer human evaluation protocols, and tighter integration of retrieval and reasoning modules. The broader goal is enabling inclusive, user-aligned AI systems that meet speakers where they naturally communicate.

A. Acknowledgment

The authors wish to thank the Mésocentre de Bourgogne-Franche-Comté for providing computing facilities, and also GENCI for support under allocation AD010613582R3.

References

[1] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "Arbert & marbert: Deep bidirectional transformers for arabic," *arXiv preprint arXiv:2101.01785*, 2021.

[2] G. Inoue, S. Khalifa, and N. Habash, "Morphosyntactic tagging with pre-trained language models for Arabic and its dialects," *arXiv preprint arXiv:2110.06852*, 2021.

[3] A. A. Alsuwaylimi, "Arabic dialect identification in social media: A hybrid model with transformer models and bilstm," *Heliyon*, vol. 10, no. 17, p. e36280, 2024.

[4] H. Bouamor, S. Hassan, and N. Habash, "The MADAR arabic dialect corpus and lexicon," in *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2019, pp. 3387–3396.

[5] O. F. Zaidan and C. Callison-Burch, "Arabic dialect identification," *Computational Linguistics*, vol. 40, no. 1, pp. 171–202, 2014.

[6] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and tv subtitles," in *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, 2016, pp. 923–929.

[7] S. Yakhni and A. Chehab, "Fine-tuning llms for low-resource dialect translation: The case of lebanese," *CoRR*, vol. abs/2505.00114, 2025.

[8] Z. Wang *et al.*, "A survey on large language model based agents," *arXiv preprint arXiv:2401.02028*, 2024.