ALADAN at IWSLT24 Low-resource Arabic Dialectal Speech Translation Task

Waad Ben Kheder[1], Josef Jon[2, 4], André Beyer[3], Abdel Messaoudi[1],
Rabea Affan[2], Claude Barras[1], Maxim Tychonov[2], and Jean-Luc Gauvain[1]

[1]Vocapia Research, France
[2]Lingea, Czechia
[3]Crowdee, Germany
[4]Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Czechia

Abstract

This paper presents ALADAN's approach to the IWSLT 2024 Dialectal and Low-resource shared task, focusing on Levantine Arabic (apc) and Tunisian Arabic (aeb) to English speech translation (ST). Addressing challenges such as the lack of standardized orthography and limited training data, we propose a solution for data normalization in Dialectal Arabic, employing a modified Levenshtein distance and Word2vec models to find orthographic variants of the same word. Our system consists of a cascade ST system integrating two ASR systems (TDNN-F and Zipformer) and two NMT modules derived from pre-trained models (NLLB-200 1.3B distilled model and CohereAI's Command-R). Additionally, we explore the integration of unsupervised textual and audio data, highlighting the importance of multi-dialectal datasets for both ASR and NMT tasks. Our system achieves BLEU score of 31.5 for Levantine Arabic on the official validation set.

1 Introduction

Speech translation (ST) systems play a crucial role in facilitating communication across languages and dialects, enabling access to information and services for diverse linguistic communities. However, developing accurate ST systems for dialectal Arabic poses significant challenges due to the scarcity of annotated data and the lack of standardized orthography. In particular, dialectal variants such as Levantine Arabic (apc) and Tunisian Arabic (aeb) are severely under-resourced in terms of Automatic Speech Recognition (ASR) and Neural Machine Translation (NMT) datasets.

These limitations present a major bottleneck in the development of high-quality ST systems and many works in previous IWSLT evaluations (Yan et al., 2022; Anastasopoulos et al., 2022; Agarwal et al., 2023; Hussein et al., 2023; Boito et al., 2022) explored various transfer techniques on the acoustic level by fine-tuning pre-trained speech encoders such as the Wav2vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021) for ASR, or neural models such as NLLB-200 (Costa-jussà et al., 2022) and mBART (Liu et al., 2020) for NMT. The use of Modern Standard Arabic (MSA) datasets like MGB2 for dialect transfer (Costa-jussà et al., 2022; Tsiamas et al., 2022) has also been proven effective.

In recent years, more sophisticated ASR architectures such as the Zipformer (Yao et al., 2023) emerged as a more effective alternative to other transformer-based architectures like Conformers (Gulati et al., 2020) and Branchformer (Peng et al., 2022). In NLP, large language models (LLMs) (Achiam et al., 2023; Brown et al., 2020; Touvron et al., 2023; Le Scao et al., 2023; Jiang et al., 2023) have demonstrated strong performance across various tasks in mainstream languages, yet a notable constraint persists in their limited support for low-resource languages and dialects.

Building upon these novelties, we propose an approach that leverages pre-trained models and multi-dialectal resources for dialectal Arabic ST. We adopt a cascade ST system comprising two ASR systems (TDNN-F and Zipformer) and two NMT modules derived from pre-trained models (NLLB-200 and Command-R). Additionally, we develop a generic text normalization methodology for Dialectal Arabic and integrate crowd-sourced NMT data and multi-dialectal datasets like PADIC (Meftouh et al., 2015) to supplement the limited training data. The outcomes for Levantine Arabic (apc) are reported on the IWSLT2024 valid and test2024 sets, while the results for Tunisian Arabic (aeb) are provided for both validation and test set test1 and dev published in IWSLT2022.

Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024), pages 240–250
August 15-16, 2024 ©2024 Association for Computational Linguistics

2 Methods

2.1 Text normalization

Due to the lack of standardized conventions across various dialects, it is necessary to design text normalization procedures in order to mitigate ambiguity and facilitate dialectal data exploitation. In this section, we detail the approach used to normalize transcripts and texts written in Dialectal Arabic. While our research primarily targets Levantine Arabic (apc) and Tunisian Arabic (aeb), we opt for the term "Dialectal Arabic" to denote a broader range of dialects. Our text normalization process includes character-level and word-level normalization to ensure consistency and accuracy in representing linguistic content.

2.1.1 Character normalization

Multiple character-level normalizations were explored in previous work on the IWSLT22 speech translation task for Tunisian Arabic. Indeed, a good improvement in ASR and ST performance was reported in (Yan et al., 2022) after removing diacritics and single character words, and applying Alif/Ya/Ta-Marbuta normalization. Despite its reported efficiency, the Alif/Ya/Ta-Marbuta normalization can alter certain words, changing their meaning; eg. the words على ("on" in English) and علي ("Ali" in English) become one and the same once this normalization is applied. For this specific reason, this normalization will not be used in our work, and more effort is invested into word-level text normalization in order to fix the most frequent Alif/Ya/Ta-Marbuta -related problems. Moreover, it's important to note that using the "single character words" filtering strategy can be harmful in the case of Levantine Arabic, which has the proclitic ع, a very frequent word, corresponding to a reduced form of the على preposition (meaning "to" or "on"). Removing such words can result in the loss of valuable grammatical information and impact the performance of NMT models.

In our work, we start by applying a similar, but less aggressive normalization, which consists of converting all eastern Arabic numerals to western Arabic numerals and removing all diacritics. Then, we normalize rare characters like the non-Arabic letter ژ and other special characters representing loan sounds such as پ for /p/, ڤ or ڥ for /v/, and گ or ڨ for /g/. It is important to emphasize the fact that the character ڤ typically denotes the sound /g/ in Tunisian and Algerian dialects (usually normalized as ق) but often represents /v/ in other dialects (usually normalized as ف).

Table 1 summarizes the character normalization rules used in our experiments.

Dialect	Normalizations
All dialects	ژ => ر / پ => ب
Levantine Arabic	ڤ or ڥ => ف
Tunisian Arabic	ڤ => ق / ڥ => ف

Table 1: Characters normalization rules for different Arabic dialects.

2.1.2 Word normalization

The second step in text normalization operates at the word level and aims at fixing orthographic inconsistencies (words written in different forms) and limiting transcription errors (misspellings or typos).

Long words normalization: While analyzing the IWSLT "aeb" dataset, we noted a significant prevalence of lengthy words (more than 180 occurrences), often representing compound terms in Arabic or French. In most instances, these elongated words encapsulate entire French sentences and should be normalized to improve readability and reduce the amount of Out-of-Vocabulary (OOV) words. These words are segmented into constituent parts based on their semantic meaning in French as shown in Table 2.

A similar phenomenon can also be observed in Arabic words, corresponding, in most cases, to combined words. A simple method to identify such words is to search for final characters mid-word, namely the "Alif maksura" (ى) and "Ta marbouta" (ة). One example is the word "الجماعةهاذوكم", which is normalized as "الجماعة هاذوكم". This criterion can also reveal spelling mistakes in frequent words like the misspelled word صحح (meaning "correct" or "true") which should be normalized as صحيح.

Orthographic variant normalization: In Dialectal Arabic transcripts, a single word may be written in various forms due to multiple factors. This variability often arises from the phonetic representation of words, where characters with similar pronunciations can be used interchangeably (such as "alif" and "alif maksura" at the end of a word). This phenomenon is also prevalent in foreign words where a word like "Google" can be written as قوقل, غوغل or جوجل depending on the country or the region, which reflects different interpretations of the loan sound /g/. French words containing nasal vowels (like /ɑ̃/, /ɔ̃/, /œ̃/ and /ɛ̃/)

	Example 1	**Example 2**
Original text	انترانمونيبليسنيموان	ألنابافيأتونسيون
Corresponding French	entraînement ni plus ni moins	elle n'a pas fait attention
Normalized text	انترانمون ني بليس ني موان	أل نا با في أتونسيون

Table 2: Two examples of elongated words corresponding to French phrases in the "aeb" IWSLT transcripts (before and after normalization).

can also be written in different ways; the most frequent ones being ـان /a:n/ or ـون /wn/.

To assist in our normalization efforts, we use a combination of orthographic and semantic similarities at the word level, by designing a weighted Levenshtein distance and using it in tandem with a Word2Vec model.

Weighted Levenshtein distance: In a recent work (Hajbi et al., 2024), a method for converting Moroccan Arabizi text to MSA based on a weighted Levenshtein distance was proposed. Inspired by this idea, we develop a weighted Levenshtein distance tailored specifically for Dialectal Arabic. This adjusted metric employs a higher cost when the insertion, removal, or substitution of a character is likely to result in the creation of a new word, particularly when consonants are altered in a word. Conversely, it assigns a lower cost when the insertion, removal, or substitution is attributable to an orthographic variant of the same word.

1. **Initialization:** All insertion, deletion and substitution costs ($\mathrm{cost}_I(.)$, $\mathrm{cost}_D(.)$ and $\mathrm{cost}_S(.,.)$ respectively) are initialized to 1.
2. **Weights modification:**
   The costs are then modified as follows:

$$\begin{cases} \mathrm{cost}_I(\mathrm{v}_i) = \mathrm{cost}_D(\mathrm{v}_i) = 0.1, \ \forall \mathrm{v}_i \in \mathbf{SV} \\ \mathrm{cost}_I(\mathrm{c}_i) = \mathrm{cost}_D(\mathrm{c}_i) = 1.5, \ \forall \mathrm{c}_i \in \mathbf{C} \\ \mathrm{cost}_S(\mathrm{c}_i, \mathrm{c}_j) = 1.5, \forall (\mathrm{c}_i, \mathrm{c}_j) \in \mathbf{C}, \ i \neq j \\ \mathrm{cost}_S(\mathrm{c}_i, \mathrm{c}_j) = 0.3, \forall (\mathrm{c}_i, \mathrm{c}_j) \in \mathbf{C}_1, \ i \neq j \\ \mathrm{cost}_S(\mathrm{a}_i, \mathrm{a}_j) = 0.3, \forall (\mathrm{a}_i, \mathrm{a}_j) \in \mathbf{A}, \ i \neq j \end{cases}$$

Where:

- **SV** = {و, ي, ا}; semi-vowels + Alif.
- **A** = {ا, أ, إ, آ, ٱ}; different variants of Alif.
- **C** = all Arabic letters, excluding semi-vowels (**SV**) and variants of Alif (**A**).
- $\mathbf{C}_1$ = {(س, ص), (ت, ط), (د, ذ), (ض, ظ)}; pairs of consonants used interchangeably in certain Arabic dialects (mainly emphatic consonants (Habash et al., 2012)).

It's important to mention that these cost values are determined empirically and can be further optimized to suit specific dialects.

By using this modified metric, the similarity between words such as بركينغ and باركنغ is diminished (two variants of the word "parking"), while the distance between باركينغ and ماركينغ is increased ("parking" vs. "marking").

In practice, relying solely on the weighted Levenshtein distance proves insufficient for effectively identifying orthographic variants of a word. This limitation arises primarily from the large size of the search space requiring the computation of distances between all pairs of words for each dialect, alongside the labor-intensive manual filtering requisite for determining the appropriate normalizations.

To address this challenge, we augment this string distance-based approach with a "semantic" proxy. This supplementary technique leverages a Word2Vec model to identify semantically similar words, thereby reducing the size of the search space prior to the application of string distance computation.

Word2vec model: Word2vec is a group of models which aim to represent words in a continuous vector space where words with similar meanings or contexts are closer to each other. This is achieved by learning representations of words based on the context in which they appear in a large corpus of text. Word2Vec identifies similar words by computing the cosine similarity (or other distance metrics) between their corresponding vectors. This model can either be implemented as a Continuous Bag of Words (CBOW) (Mikolov et al., 2013a) where a word is predicted given its context, or as a Skip-gram model (Mikolov et al., 2013b) where the context is predicted given a word. In our work, we use a CBOW model with a 100-dimensional word embeddings and a window size of 5. The similarity between the embeddings is computed as the cosine similarity (range = $[-1, 1]$).

The following algorithm is used to find the orthographic variants of each word:

For each word **w** in the vocabulary **V**:

1. Use the Word2vec model to find the 50 clos-

est words v_k to w using the cosine distance between their embeddings $emb_{\mathbf{v}_k}$:
$\{\mathbf{v}_k \mid \cos(emb_{\mathbf{w}}, emb_{\mathbf{v}_k}) > 0.3,\ \forall \mathbf{v}_k \in \mathbf{V}\}$

2. Compute the weighted Levenshtein distance lev_W and keep the words $\mathbf{v}_k$ close to $\mathbf{w}$: $\{\mathbf{v}_k \mid \text{lev}_W(\mathbf{w}, \mathbf{v}_k) < 3\}$

Following this process, the largest clusters of words are identified and manually checked. These are some examples of apc/aeb clusters:

- The Tunisian word for "anyway": حاسيلو، حاصلو، حاسيول، حاصولو، حاصيله، حاصلو، حاصيلو.
- The Syrian word for "the computer": الكويبوتير، الكمبيوتر.
- The French word "normalement": ،نرملن، نرمامون، نرماللمون نورموملون، نورمامون، نورماملن، نرملمون، نورملومن، نورملمان، نورماماملون، نورماملون.

2.2 ASR

Different architectures were tested in previous IWSLT evaluations for Dialectal Arabic and most of them opted for end-to-end architectures such as a Conformer encoder + CTC (Yan et al., 2022; Boito et al., 2022), a Branchformer encoder + a Transformer decoder (Hussein et al., 2023) showing the superiority of transformer-based architectures for this task. In recent years, the Zipformer architecture (Yao et al., 2023) was introduced as a more effective end-to-end model where, differently from Conformer that processes the sequence at a fixed frame rate of 25Hz, models use a U-Net-like structure and learn temporal representation at different resolutions in a more efficient way. The Zipformer architecture achieves state-of-the-art performance while capturing long-range dependencies and contextual information.

In crafting our ASR system, we prioritized compactness and speed in the selection of architectures. The ASR module developed in this work and used for the speech translation (ST) task comprises two distinct systems (TDNN-F and Zipformer), whose outputs are combined using the Recognizer Output Voting Error Reduction (ROVER) algorithm (Fiscus, 1997) for enhanced performance. Combining a TDNN-F model and an end-to-end model like Zipformer can be a powerful strategy to leverage the strengths of both approaches and achieve improved ASR performance. The TDNN-F model excels in its modular design, allowing for fine-tuning of each component independently while Zipformers streamline the ASR pipeline by implicitly learning relevant features from raw data and capturing long-range dependencies more effectively. Moreover, this system combination can mitigate some known limits of end-to-end architectures, such as high deletion errors, especially when dealing with long utterances (Chiu et al., 2021; Fox et al., 2024).

2.3 ST

We classify the systems we have experimented with based on their specificity into two categories: MT-only and prompt-driven LLMs.

2.3.1 MT-only

We experimented with multiple encoder-decoder Transformer models (see Section 3.3).

2.3.2 Prompt driven LLMs

Context Traditional MT models in the majority operate only on sentence level, without regard for a larger surrounding context. LLMs however are usually trained on longer chunks of text and can innately use the information in the context.

A simple way to translate a whole conversation using an LLM would be to use a prompt along the lines ``*Translate the following conversation into English:* conversation"

One practical problem with this approach is that the task evaluation is still sentence-level, meaning we need to keep the same sentence boundaries across source and translated conversations. In our experience, this was difficult to achieve reliably with all the LLMs we have experimented with. There are multiple possible approaches to obtain the same segmentation as in the input:

- Sentence splitter -- Unreliable and introduces another tool into the pipeline
- Asking the model to keep the same number of lines: only works for documents with a small number of lines and even then, the model still moves the content across sentence boundaries
- Using separator token, e.g. [s] in the input to separate sentences and asking the model to keep it in the corresponding position in the translation -- similar issues as above
- Only translating one sentence at a time, but providing the whole context (source document and previously translated prefix) in the prompt

We have chosen to use the last option based on the experimental results. Our final prompt for context-aware NMT is shown in Listing 1.

2.4 Finetuning

We finetune some of the models on datasets described in Section 3.1. For traditional MT mod-

```
We need to translate a single line from conversation in Tunisian Arabic into English.
This is the conversation: {src_context}
The start of the conversation is already translated into English: {prev_context}
Translate the following line from {src_lang} to {tgt_lang}.
Be very literal , and only translate the content of the line , do not add any
explanations: {src_line}
```

Listing 1: The final context-aware prompt we used in our submission.

els, we finetune all the weights. For LLMs, we use QLoRA (Dettmers et al., 2023). The hyperparameters are described in Section 3.3.

2.5 Reranking

We rerank the outputs of multiple systems using Minimum Bayes Risk (MBR) decoding (Goel and Byrne, 2000; Kumar and Byrne, 2004; Freitag et al., 2021), with COMET22-DA (Rei et al., 2022) as the objective metric. MBR allows for the use of reference-based metrics for reranking even in cases where the reference is unavailable, by instead using the initial translation candidates as pseudo-references. For the final submission, we used a method introduced by Jon and Bojar (2023), which combines MBR decoding with a genetic algorithm to combine and mutate the translation candidates to create better quality translations.

3 Experiments

This section describes our experimental settings, used data and results.

3.1 Data

In this subsection, we list the datasets we used for training and evaluating our systems.

3.1.1 ASR data

Table 3 summarizes the audio data used to build our ASR models. To improve the robustness of our ASR system, these data are augmented using speed perturbation, additive noise and reverberation.

3.1.2 NMT data

Table 4 summarizes the textual data used to train the MT models and fine-tune the LLMs.

Constrained datasets: IWSLT22 (LDC2022E01) consists of "aeb" speech, reference transcript and eng translations, containing 202k sentence pairs. The **UFAL parallel dataset** (Krubiński et al., 2023) contains multi-lingual parallel sentences (including "eng", "arb" and "apc").

Dataset	Dur.
Public supervised data	
GALE (BN/BC)	2800h
Tunisian Arabic (CTS) / IWSLT22	160h
Moroccan Arabic (CTS) / Appen [1]	30h
Levantine Arabic (CTS) / LDC [2]	250h
Internal supervised data	
Levantine Arabic (CTS)	365h
Egyptian Arabic (CTS)	135h
Algerian Arabic (CTS)	300h
Tunisian Arabic (Youtube)	20h
Moroccan Arabic (Youtube)	20h
Unsupervised data	
Tunisian Arabic (Radio)	150h
Total	4230h

Table 3: List of datasets used to train the ASR module.

Dataset	Dialect(s)	# sents.
UFAL	arb, apc	120k
LDC2012T09[3]	arz, apc	176.1k
IWSLT22[4]	aeb	202.4k
PADIC-ENG	arb, aeb, arq, apc, ary	44,8k
MADAR-ENG	25 cities	12k
Interviews	apc	4.8k
Global Voices	arb	63k
Crowd-sourced	apc	9.5k

Table 4: Datasets used for NMT finetuning.

Crowd-sourced data: We collaborate with our ALADAN partner, Crowdee[5], a micro-task crowdsourcing platform, to construct a parallel dataset for Levantine Arabic (apc) to English (eng) NMT. To ensure the high quality of the dataset, we design a linguistic assessment test consisting of 40 questions in Levantine Arabic. These questions cover various aspects, including Arabic grammar and multiple-choice translation exercises between "apc" and "eng".

In these tasks, transcripts from our internal Levantine Arabic CTS dataset (mentioned in Table 3) dataset are used as input, and the resulting dataset

[5] Crowdee—https://www.crowdee.de/

contains 9.5k parallel sentences.

PADIC-ENG: PADIC (Meftouh et al., 2015) is a multi-dialect dataset containing 6400 parallel sentences encompassing six distinct dialects: two Algerian variants, along with Palestinian, Syrian, Tunisian, and Moroccan Arabic, in addition to MSA. We translated the MSA side into English using the NLLB-1.3B model.

MADAR-ENG: MADAR (Bouamor et al., 2018) is a 25-way multiparallel dataset collected in 25 Arabic-speaking cities. We also translated the MSA side into English and paired the translation with source sides from cities located in Levantine or Tunisian Arabic-speaking regions.

Interviews: We scraped a website containing interviews in English with refugees and their experience with the integration in their new countries[6], resulting in 4.8k collected sentences. We translated the text into "apc" using the NLLB-1.3B model and used the resulting dataset as a backtranslation finetuning data. We have selected this website based on the domain similarity with the validation data.

LDC2012T09 contains dataset parallel sentences translated from Egyptian Arabic (arz), North Levantine Arabic (apc) and South Levantine Arabic (ajp) to English (eng). It was developed by Raytheon BBN, LDC, and Sakhr Software and provided to our project consortium for the purposes of the shared task free of charge by LDC.

Global Voices dataset was collected by the CASMACAT project. The Arabic-English part consists of 63k parallel sentences.

Apc-valid is provided by the organizers.[7]

3.2 ASR

3.2.1 ASR models

(A) TDNN-F: The first system, is based on the Factorized Time-Delay Neural Network (TDNN-F) architecture as outlined in (Povey et al., 2018). This model consists of 15 layers with approximately 28 million parameters. The ReLU layer dimension is set to 1920, with linear bottlenecks of dimensions $\{320, 240\}$. This acoustic model is coupled with an n-gram language model.

(B) Zipformer: The second system adopts an End-to-End architecture utilizing the Zipformer design, and more specifically the "Zipformer-M" configuration described in (Yao et al., 2023).

(C) Zipformer+TDNN-F: The output of the two developed ASR systems (A) and (B) are combined using the ROVER algorithm.

3.2.2 Training procedure

First, we train generic models (TDNN-F and Zipformer) using all available data to take advantage of the acoustic and linguistic similarities between different Arabic dialects. These pre-trained multi-dialect models are then fine-tuned using "apc" (or "aeb") -only data.

The TDNN-F model is pre-trained for 10 epochs (on all data) using lr=1e-3, then fine-tuned using LF-MMI-based transfer learning (Ghahremani et al., 2017) for 8 epochs using lr=2e-5, a primary lr-factor of 0.1 and a lr-factor of 1.0 for the last layer. The Zipformer model is pre-trained for 80 epochs (on all data) using the lr=4e-3 then ran 50 epochs for fine-tuning by using dialect-only data and lr=5e-3.

3.2.3 Results

Table 5 summarizes the WERs achieved by our ASR systems after applying the normalization procedure detailed in Section 2.1. This normalization significantly improved WERs for "apc" and "aeb" by 10% and 18%, respectively. The combined model achieved even greater improvements, demonstrating the complementarity of the two models and outperforming all WERs reported in (Agarwal et al., 2023) for "aeb".

	apc	**aeb**	
	apc-valid	**dev**	**test1**
(A) TDNN-F	26.5	39.9	40.8
(B) Zipformer	25.8	33.7	34.3
(C) Zipformer +TDNN-F	**23.6**	**32.7**	**33.1**

Table 5: WER (%) of ASR models on IWSLT24 Levantine Arabic (apc) validation and IWLST22 Tunisian Arabic (aeb) dev/test sets.

3.3 ST

We compare lower-cased BLEU (Papineni et al., 2002), ChrF (Popović, 2015) and COMET22-DA (Rei et al., 2022) scores of multiple systems on *apc-valid*, both on human transcriptions and in cascaded setting with our ASR systems.

3.3.1 Baselines

We have compared multiple open-source MT models (Costa-jussà et al., 2022; Kudugunta et al., 2023) and LLMs (Mesnard et al., 2024; Jiang

[6] https://socialscienceworks.org

[7] https://github.com/ufal/IWSLT2024_Levantine_Arabic_data

Type	Model	Human			ASR		
		BLEU	chrF	COMET	BLEU	chrF	COMET
MT	eTranslation	15.9	41	0.615	14.1	38.9	0.595
	GoogleTranslate	**29.9**	**55.7**	**0.780**	**26.3**	**51.6**	**0.747**
	MADLAD-10B	18.4	42.4	0.711	15.9	39.0	0.678
	NLLB200-1.3B	21.1	47.5	0.739	18.7	44.6	0.716
	NLLB200-600M	20.7	47.2	0.745	18.7	44.1	0.715
	NLLB200-3.3B	21.1	47.4	0.728	18.1	43.9	0.700
	Opus-MT	10.5	36.5	0.595	10.1	35.7	0.579
LLM	Jais-13B	21.9	45.5	0.755			
	Bloom-z	13.9	36.2	0.703			
	1-shot	15.1	37.5	0.716			
	Aya-101	16.1	42.3	0.711			
	1-shot	17.6	42.9	0.714			
	ALMA	7.1	32.0	0.587			
	1-shot	7.8	30.3	0.593			
	Mistral	8.5	35.4	0.620			
	1-shot	8.1	36.9	0.608			
	Gemma	6.7	31.8	0.563			
	1-shot	6.8	27.7	0.561			
	Command-R full+context	**29.5**	**54.1**	**0.805**			
	Command-R 4bit	24.3	49.8	0.778	20.7	46.2	0.737
	1-shot	25.6	50.4	0.785	21.7	46.6	0.749
	context	26.9	51.9	0.793	22.9	47.8	0.765
	1-shot + context	26.1	51.7	0.797	24.2	49.0	0.771

Table 6: Baseline models for ST. The first row displays the origin of the transcribed source file: *Human* are the transcriptions provided by the task organizers, *ASR* are the outputs of our Zipformer+TDNN-F ASR model. Missing values for +*context* in LLMs means that the given model was not able to provide the translation in the line-by-line format necessary for the evaluation. We did not evaluate most of the LLMs on the ASR transcriptions, since we already ruled these models out from the further experiments.

et al., 2023; Üstün et al., 2024; Sengupta et al., 2023; Muennighoff et al., 2022) in both sentence-to-sentence and context-aware translation. In the prompt-driven LLMs, we used a simple prompt in the form ``*Translate the following sentence from Levantine Arabic to English:* {source_sentence}" for sentence-to-sentence translation.

We evaluated the context-aware approach only with the LLMs and we used the prompt shown in Listing 1. We sample with temperature $t = 0.2$ based on preliminary experiments for the decoding. We compare 0-shot and 1-shot scenarios, with a short example taken directly from the valid set, so the model sees one short excerpt from the validation set with the correct translation.

The results are shown in Table 6. We see that the models vary greatly, with the best scores obtained by the commercial engine in the case of sentence-level, traditional MT models, and Command-R in the case of LLMs. The only LLM that responded well to our context-aware prompt was Command-R, for the other models, the output was not usable.

3.3.2 Finetuning

We selected one model from each category (MT, LLM): NLLB and Command-R, due to their best scores and good instruction-following capabilities in the case of the latter. We finetuned them on MT datasets listed in Section 3.1. The results on *apc-valid* are shown in Table 7.

For Command-R finetuning, we used the 4-bit quantized model (due to hardware limitations) and QLoRA with r values of 8, 16, 32 (at higher values we ran into memory issues), α equal to either $r/2$, r or $2r$ and learning rates set to either $1e-4$, $5e-5$ or $1e-5$. We did not see significant differences in metrics scores between these configurations. We ran the finetuning for 5000 updates with a batch size of 48, on a single A100 80GB GPU. Even though the number of updates only covers about 15% of the whole finetuning dataset, we did not see any improvements from continued training.

We also experimented with multiple decoding algorithms, namely sampling with temperature (Hinton et al., 2015; Ackley et al., 1985), contrastive search (Su and Collier, 2022; Su et al., 2022), locally typical sampling (Meister et al., 2023), and beam search (Graves, 2012). We did not find any significantly better configuration than sampling with $t = 0.2$.

		Human			ASR		
#	Model	BLEU	chrF	COMET	BLEU	chrF	COMET
1	**NLLB-1.3B**	21.1	47.5	0.739	18.7	44.6	0.716
2	+UFAL-APC	21.4	44.4	0.723	17.7	40.5	0.686
3	+IWSLT22	25.1	52.2	0.741	21.3	47.9	0.702
4	+3-transcribed	23.3	50.1	0.746	19.2	46.1	0.710
5	+LDC2012T09	27.8	53.6	0.764	23.8	49.8	0.725
6	+Interviews	21.8	49.9	0.740	19.2	46.8	0.707
7	+CrowdSourced	27.4	53.2	0.759	24	49.3	0.722
8	+GlobVoic	21.8	48.2	0.746	19.7	45.2	0.716
9	+MADAR-MT	19.5	44.1	0.734	17.3	41.2	0.701
10	+PADIC-ENG	26.6	52.7	0.757	23.3	49.1	0.718
11	*+2+3+4+5+10*	-	-	-	29.1	53.1	0.753
12	+3+4+5+6+10	30.1	56	0.777	26.4	52	0.737
13	+3+4+5+6+7+10*	30.6	56.2	0.780	27	52.2	0.742
14	**Command-R 4bit**	26.9	51.9	0.799	22.4	50.2	0.743
15	*+3+4+5+6+10*	34.4	58.1	0.805	30	53.4	0.771
16	+3+4+5+6+7+10*	33.8	57.9	0.806	30.1	53.4	0.768
17	15+MBR	34.5	58.4	0.812	31.1	54.6	0.781
18	*15+MBR-GA*	-	-	-	31.5	55	0.782

Table 7: Fintuning of NLLB-1.3B and Command-R-4bit models. Models marked with asterisk were trained after the end of the shared task and are not a part of the submission. The first row displays the origin of the transcribed source file: *Human* are the transcriptions provided by the task organizers, *ASR* are the outputs of our Zipformer+TDNN-F ASR model. Rows 18, 15, and 11 show our primary, first contrastive, and second contrastive submissions, respectively.

3.4 Final submission

Our primary submission consists of 26 best validation BLEU checkpoints from the finetuned Command-R model from row 15, combined using MBR decoding and a genetic algorithm (Jon and Bojar, 2023; Jon et al., 2023; row 18 in Table 7). We did not carry out the MBR-GA combining for the translations of the reference human transcriptions due to the computational requirements of the process. Our first contrastive submission is the translation from the single best LLM system we trained before the end of the competition (row 15). The second contrastive submission is the best NLLB model trained before the deadline, shown in row 11.

3.5 Conclusion

In this paper, we introduced a generic data normalization method for dialectal Arabic text using a modified Levenshtein distance metric and Word2vec word embeddings, improving ASR performance by up to 18%. We demonstrated the benefits of multi-dialectal modeling and combining models, achieving WERs of 23.6 on the "apc" validation set, 32.7 on the "aeb" dev set, and 33.1 on the "aeb" test1 set. In the MT part, we compared various MT models and LLMs, highlighting the superior performance of LLMs due to their larger context windows. By gathering additional training datasets, we demonstrated the effectiveness of traditional finetuning for NMT models and QLoRA finetuning for LLMs. Combining multiple finetuned models yielded a BLEU score of 31.5 on the "apc" validation set.

Acknowledgements

This work was funded by the the European Defence Fund (EDF) 2021 project ALADAN (Ai-based LAnguage technology development framework for Defence ApplicatioNs; Grant ID: 101102545). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. We would like to express our gratitude to LDC for kindly providing the data used in this study.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David H. Ackley, Geoffrey E. Hinton, and Terrence J.

Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cogn. Sci.*, 9:147--169.

Milind Agarwal, Sweta Agarwal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, et al. 2023. Findings of the iwslt 2023 evaluation campaign. Association for Computational Linguistics.

Antonios Anastasopoulos, Loc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, et al. 2022. Findings of the iwslt 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98--157. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449--12460.

Marcely Zanon Boito, John Ortega, Hugo Riguidel, Antoine Laurent, Lo□c Barrault, Fethi Bougares, Firas Chaabani, Ha Nguyen, Florentin Barbier, Souhir Gahbiche, et al. 2022. On-trac consortium systems for the iwslt 2022 dialect and low-resource speech translation tasks. *arXiv preprint arXiv:2205.01987*.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877--1901.

Chung-Cheng Chiu, Arun Narayanan, Wei Han, Rohit Prabhavalkar, Yu Zhang, Navdeep Jaitly, Ruoming Pang, Tara N Sainath, Patrick Nguyen, Liangliang Cao, et al. 2021. Rnn-t models fail to generalize to out-of-domain audio: Causes and solutions. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 873--880. IEEE.

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

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

Jonathan G Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347--354. IEEE.

Jennifer Drexler Fox, Desh Raj, Natalie Delworth, Quinn McNamara, Corey Miller, and Migüel Jetté. 2024. Updated corpora and benchmarks for long-form speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13246--13250. IEEE.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2021. Minimum bayes risk decoding with neural metrics of translation quality.

Pegah Ghahremani, Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur. 2017. Investigation of transfer learning for asr using lf-mmi trained neural networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 279--286. IEEE.

Vaibhava Goel and William J Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Computer Speech & Language*, 14(2):115--135.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *Preprint*, arXiv:1211.3711.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Nizar Habash, Mona T Diab, and Owen Rambow. 2012. Conventional orthography for dialectal arabic. In *LREC*, pages 711--718.

Soufiane Hajbi, Omayma Amezian, Nawfal El Moukhi, Redouan Korchiyne, and Younes Chihab. 2024. Moroccan arabizi-to-arabic conversion using rule-based transliteration and weighted levenshtein algorithm. *Scientific African*, 23:e02073.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451--3460.

Amir Hussein, Cihan Xiao, Neha Verma, Thomas Thebaud, Matthew Wiesner, and Sanjeev Khudanpur. 2023. Jhu iwslt 2023 dialect speech translation system description. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 283--290.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Josef Jon and Ondřej Bojar. 2023. Breeding machine translations: Evolutionary approach to survive and thrive in the world of automated evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2191--2212, Toronto, Canada. Association for Computational Linguistics.

Josef Jon, Martin Popel, and Ondřej Bojar. 2023. CUNI at WMT23 general translation task: MT and a genetic algorithm. In *Proceedings of the Eighth Conference on Machine Translation*, pages 119--127, Singapore. Association for Computational Linguistics.

Mateusz Krubiński, Hashem Sellat, Shadi Saleh, Adam Pospíšil, Petr Zemánek, and Pavel Pecina. 2023. Multi-parallel corpus of north levantine arabic. In *Proceedings of ArabicNLP 2023*, pages 411--417.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset. *Preprint*, arXiv:2309.04662.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169--176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726--742.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on padic: A parallel arabic dialect corpus. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 26--34.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102--121.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

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

Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe. 2022. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *International Conference on Machine Learning*, pages 17627--17643. PMLR.

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

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743--3747.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation*

(WMT), pages 578--585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Alham Fikri Aji, Zhengzhong Liu, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *arXiv preprint arXiv:2210.14140*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ioannis Tsiamas, Gerard I Gállego, Carlos Escolano, José Fonollosa, and Marta R Costa-jussà. 2022. Pretrained speech encoders and efficient fine-tuning methods for speech translation: Upc at iwslt 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 265--276.

Brian Yan, Patrick Fernandes, Siddharth Dalmia, Jiatong Shi, Yifan Peng, Dan Berrebbi, Xinyi Wang, Graham Neubig, and Shinji Watanabe. 2022. Cmu's iwslt 2022 dialect speech translation system. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 298--307.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2023. Zipformer: A faster and better encoder for automatic speech recognition. *arXiv preprint arXiv:2310.11230*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.