DEVELOPMENT OF SMART VOICE AGENT
With case study (Libyan Voice Assistant)

Maryam Omar AboSarafa
Department of Programming and Systems Analysis
College of Computer Technology, Tripoli
Tripoli, Libya
eng.maryalbosyifi@cctt.edu.ly

Mohamed Abolgasem Arteimi
Electrical and Computer Engineering Department
Libyan Academy,
Tripoli, Libya
https://orcid.org/0000-0002-1729-2172
arteimi@yahoo.com

Abstract - The paper presents the creation of an end-to-end voice assistant system designed for a lesser-resourced dialect of Arabic, Libyan Tripolitanian, which does not receive local support in commercial ASR and NLP applications. To remediate this lack, we built a demographically balanced and phonemically rich corpus of speech data containing over 13,000 audio samples. It contains both natural and semi-structured utterances and is annotated using the CODA* orthography for dialectal Arabic. Using this dataset, we trained the OpenAI Whisper model with the Hugging Face Transformers, achieving a WER (Word Error Rate) reduction of 2.045 → 0.040. To assist in managing smartphone commands and having simple conversations in Tripolitanian Arabic, the ASR output is passed to a Rasa-based chatbot that is trained on intent-annotated queries. The chatbot was able to perform with 100% intent accuracy and a 0.998 entity F1-score. This modular pipeline is confirmed by evaluation results on standard ASR and NLU metrics. These findings show that it is possible to create high-performance, specific voice interfaces based on training for specific dialect inquiry through domain-adapted training, data augmentation, and system integration. Future expansions include extending the dataset to suit use in speech synthesis in the Libyan dialect and broader Libyan dialect support.

Index Terms - Tripolitanian Arabic, Libyan Dialect, Automatic Speech Recognition, Whisper ASR, Rasa Chatbot, Spoken Dialogue Systems, Code-Switching, Low-Resource Languages, Natural Language Understanding, Dataset Augmentation.

1. INTRODUCTION

Voice assistants are now integrated into smart devices, allowing for voice-based interactions with applications and services [4]. Most of these systems address high-resource languages, such as English, Mandarin, and French. Although Arabic is the fifth most widely spoken language in the world, its dialects are severely underrepresented in voice AI systems, as stated in [5] and [6].

Libyan Arabic accents, including the Tripolitanian dialect in the west of the country, is one of the most neglected. As a matter of fact, this dialect contains phonetic and lexical influences from Classical Arabic, Italian, English, Turkish, and Berber dialect, which leads to its being linguistically distinct and hard for general-purpose ASR systems, as discussed in [7] and [8].

While many modern state-of-the-art ASR systems, including Whisper by OpenAI, presented in [2], perform excellently on multilingual benchmarks, their zero-shot results on Arabic dialects (especially Libyan Arabic) are reduced due to the absence of dialectal training data [9] and [10] This limits access to speech technologies for millions of Libyan speakers and exacerbates digital inequality in AI, as reported in [11].

In this paper, we present a study that makes three important contributions:

- It contains the first Libyan Tripolitanian dialect speech data ever recorded, including commands, stories, conversations, idioms, spontaneous speech, etc.

DOI: 10.5281/zenodo.15505226

- It demonstrates the quality of fine-tuning Whisper on low-resource Arabic dialects and significantly reduces the word error rate (WER).

- It includes a functional spoken-language interface with Rasa for the purpose of day-to-day smartphone commands and performing simple dialogues using the Libyan dialect.

2. Related Work

Arabic speech recognition has conventionally targeted Modern Standard Arabic (MSA) since it is utilization in formal media and corpora exist, as noted in [12]. However, everyday communication in the Arab world is predominantly in local dialects, which are substantially disparate from one another in pronunciation, morphology, and vocabulary. Among them, Libyan Arabic is one of the least represented in computational linguistics, as described in [13] and [14].

2.1 Arabic Dialect Datasets

For dialectal Arabic, a number of noteworthy corpora have been created, including:

- MADAR: The MADAR project offers resources for MSA and 25 dialectal varieties, but notably lacks representation for Libyan Arabic [6].

- The Egyptian Arabic Speech Corpus: The Egyptian Arabic Speech Corpus has been employed in several speech recognition studies to address the challenges of dialectal Arabic, as the authors pointed out in [15].

- The ADI Challenge Dataset [17] and the Arabic Speech Corpus [16], Establish reference data for the classification and identification of dialects.

There are no specific datasets for Libyan Arabic in these resources, which primarily concentrate on Egyptian, Gulf, Levantine, and MSA.

2.2 Whisper and Multilingual ASR

The Whisper model by OpenAI [2] is a transformer-based ASR model trained on 680,000 hours of multilingual, multitask supervised data. It is notable for its zero-shot generalization capabilities. However, researchers have reported that its baseline WER on Arabic dialects (without fine-tuning) is significantly higher compared to high-resource languages [9] and [18].

Recent works have demonstrated that fine-tuning Whisper on domain-specific datasets significantly improves ASR performance in low-resource languages and accents [9] and [20]. To date, however, no prior work has explored Whisper fine-tuning for Libyan Arabic.

This study conducted a zero-shot evaluation of Whisper and Wav2Vec 2.0 on carefully selected one-hour dataset of Tripolitanian Libyan dialect, which has been highlighted by rich phonemic variation and a variety of spoken forms. Results show that Whisper achieves a total word error rate (WER) of 0.494 and character error rate (CER) of 0.195 , significantly exceeding Wav2Vec 2.0, which achieved a total WER of 1.045 and CER of 0.956. Whisper's outputs were consistently more understandable and semantically coherent particularly in spontaneous, dialect-rich utterances.

2.3 Dialectal Arabic Chatbot Development

A popular framework for creating task-oriented chatbots is Rasa [3], which is open-source. It offers powerful tools for entity extraction, intent recognition, and dialogue policy learning, as demonstrated in [21].

Rasa has been used in Arabic NLP for hybrid systems, MSA, and Egyptian Arabic, as shown in [22]. However, rather than transcribed dialectal speech, these systems frequently assume clean text inputs. Additionally, code-switching, orthographic variation, and phonological ambiguity continue to be problems for NLU modules in Arabic dialects [23].

Basic Arabic is supported by commercial systems such as Google Assistant and Apple Siri, but they are only able to handle MSA or Egyptian dialects and are unable to handle dialect-specific intent logic [24] .

As far as we are aware, no previous research has combined a refined Whisper ASR model with a Rasa chatbot to accommodate the Libyan Tripolitanian dialect.

3. Methodology

The three main components of the suggested system are described in this section: developing a speech dataset in the Tripolitanian dialect, optimizing the Whisper ASR model by fine-tuning it, and integrating ASR with a Rasa voice-driven chatbot.

3.1 Creation of Datasets

3.1.1 Collecting Data

Two sources were used in the construction of the Tripolitanian Arabic dataset:

- Using USB microphones in controlled settings, native speakers between the ages of 8 and 66 made both scripted and unplanned recordings.

- The second source is YouTube, from which audio was extracted from videos that are accessible to the general public after using TF-IDF-based search techniques and keyword filtering, as described in [25] to get the samples of Tripolitanian dialect.

Participants were recorded using Audacity software with settings of 16 kHz, mono channel, and 16-bit WAV format, and they were balanced by age, gender, and dialect fluency.

3.1.2 Content of Dataset

The dataset includes more than 13,000 audio samples, which can be classified into the following categories:

- The dataset has perfect coverage of the Libyan Dialect phonemes.

- Lexical Variety: synonyms for the frequently used words, idiomatic expressions, culturally significant terms, expressions, greetings, and names are included.

- Grammar patterns: Different sentence types like questions, commands, and declarative statements, as well as mixed grammar constructions, are comprised.

- Linguistic differences: The code-switching was taken into account, such as mixing Arabic with other languages such as Italian or English.

- Contextual Content: contains examples of typical everyday situations in which the Tripolitanian dialect is used, i.e. Everyday interactions, business meetings, and informal topics.

The CODA* system was used to transcribe each file, as it establishing a single orthographic standard for dialectal Arabic transcription while maintaining spoken variation, as described in [1]. The metadata file includes speaker ID, age, gender, environment, Dialect, speaking style, sampling rate, bit-depth and device.

3.1.3 Augmenting Data

To simulate real-world natural environments, audio augmentation techniques were used by adding various background noises to the real recording files. These noises represent road noise, café ambience, inside a car while driving, crowded people in a small room, and a rainy day. This enhanced model robustness outside of the lab, which is consistent with [26] and [27] practices.

3.2 Fine-Tuning Whisper ASR

3.2.1 Configuration of the Model

Using the Hugging Face Transformers framework and the Seq2SeqTrainer, we enhanced OpenAI's Whisper Medium model by fine-tuning it, as demonstrated in [2] using the Libyan Dialect Dataset. and the following hyperparameters were set: 30 epochs; 8 batch sizes; 4e-5 learning rates; 17 gradient accumulations; 136 effective batch sizes; Linear Warmup is the scheduler, and the CTC Loss + Cross Entropy is the loss function. The model's tokenizer and processor components (Whisper Processor) were initialized from multilingual checkpoints

3.2.2 Preprocessing

Every recording was:

- Trimmed and normalized.

- Tokenized with Whisper's built-in multilingual tokenizer

- Augmented with silence padding and noise injection

- Tracked using Tensor Board and evaluate.load("wer") for real-time monitoring

During early epochs, training logs demonstrated rapid WER decline and consistent loss minimization.

3.3 Integration of Rasa Chatbot

3.3.1 Pipeline NLU

Rasa Open Source 3.x was used to create the chatbot as described in [3]. The pipeline contained:

- Regex Featurizer
- DIETClassifier for multitask intent and entity prediction; CountVectorFeaturizer
- Language: Personalized tokenizer for Arabic that complies with CODA.

3.3.2 Training RASA with Libyan Arabic Data

Training was performed on YAML files annotated with intent, entities, slots and custom actions in Tripolitanian Arabic.

Figure 1 illustrates the training process of a chatbot that was built based on the Rasa framework. First it is parsed

Using training files (nlu.yml, story.yml, and rules.yml) that were pre-prepared in the Libyan dialect.

Then the WhitespaceTokenizer is used to tokenize texts, and RegexFeaturizer, LexicalSyntacticFeaturizer, and CountVectorsFeaturizer are used to extract features. The DIETClassifier is trained using these features that was extracted in the previous step to detect intents and extract entities, and to handle retrieval-based queries, the ResponseSelector was trained. Rasa simultaneously used TEDPolicy and RulePolicy to predict the next action and extracts dialogue patterns from structured stories. Via this combined training approach, the assistant was able to understand Libyan dialect inputs and react contextually.

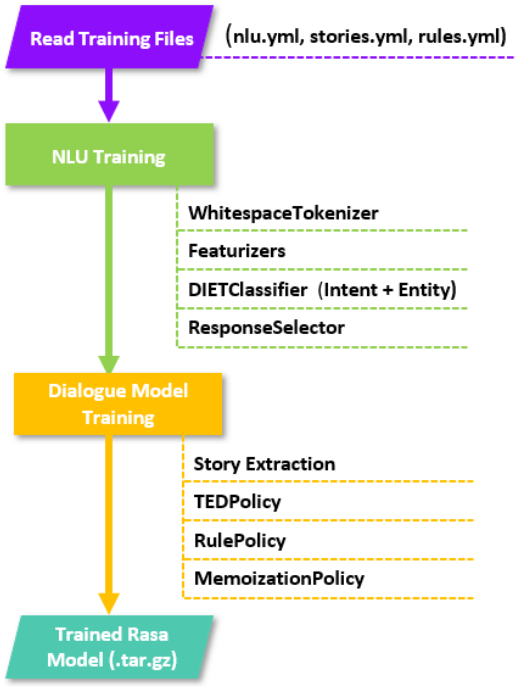

FIGURE 1. RASA CHATBOT TRAINING PIPELINE FOR TRIPOLITANIAN ARABIC. THE WORKFLOW SHOWS PARSING OF TRAINING FILES, FEATURE EXTRACTION, INTENT AND ENTITY RECOGNITION, AND DIALOGUE POLICY LEARNING.

3.3.3 Dialogue Management

For dialogue flow and command execution:

- TEDPolicy was used to learn conversation patterns
- RulePolicy defined fixed-response behavior
- MemoizationPolicy supported short history matching.

Rasa's fallback and NLU thresholds were adjusted for high precision in intent resolution.

3.3.4 VOICE INPUT & CUSTOM ACTIONS

The assistant uses rasa_sdk to open applications such as WhatsApp, Facebook, Messenger, etc., adjust Wi-Fi, Bluetooth, airplane mode, etc., navigate between screens, make calls, and read the battery level, date, and time.

Verifies user requests and reads the system state. The pipeline was expanded to accept ASR transcripts through WebSocket.

Figure 2 shows the complete sequence of interaction in the Libyan voice assistant's Natural Language Understanding (NLU) pipeline. When the user speaks in Tripolitanian (Libyan dialect), the process starts. fine-tuned Whisper ASR model converts the input audio into text. Then the text that resulted from the previous step is passed to the Rasa NLU pipeline, which tokenizes and featurizes it before useing DIETClassifier to classify intents and extract related entities. The Policy Engine.

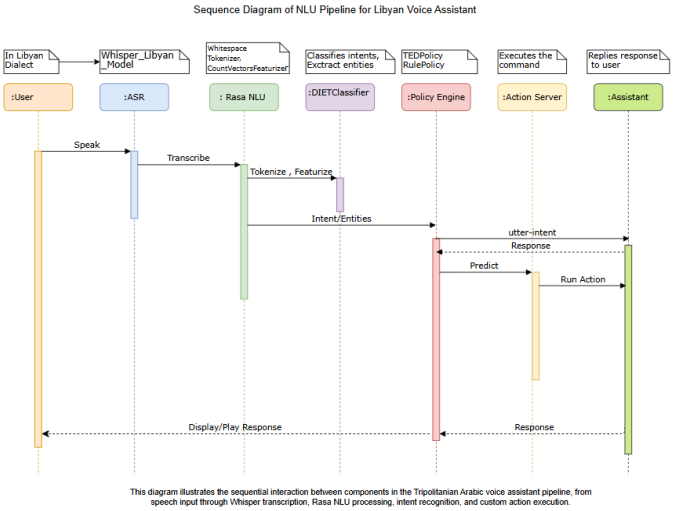

FIGURE 2: END-TO-END INTERACTION FLOW OF THE VOICE AGENT. ILLUSTRATES USER INPUT PROCESSING FROM SPEECH RECOGNITION TO INTENT CLASSIFICATION AND SYSTEM RESPONSE GENERATION.

4. SYSTEM ARCHITECTURE: SMART VOICE AGENT FRAMEWORK

To illustrate the practical deployment of how all modules interoperate, we integrated all by creating a unified system known as the Smart Voice Agent Framework applicable to the Libyan Tripolitanian dialect. Figure 3 provides an overview of the complete framework, which is essentially a diagram showing the flow of speech input leading to the execution of the action and the system response (see Figure 3).

DOI: 10.5281/zenodo.15505226

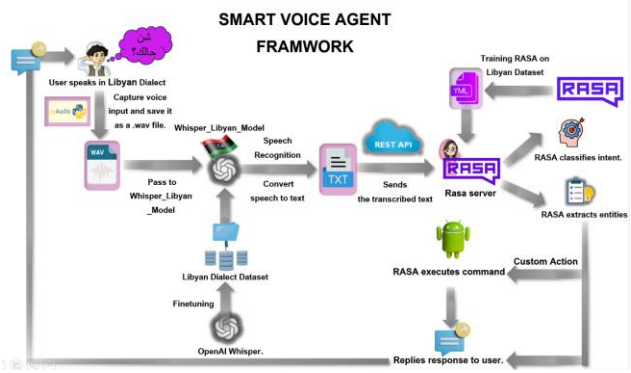

FIGURE 3. SMART VOICE AGENT SYSTEM ARCHITECTURE. PRESENTS THE INTEGRATED FRAMEWORK FOR VOICE INPUT, ASR, RASA NLU, DIALOGUE MANAGEMENT, AND ACTION EXECUTION.

The Smart Voice Agent Framework enables users to control smartphone features and receive text feedback using natural Tripolitanian Arabic. It uses a tuned Whisper ASR to do recognition of dialectal speech and a Rasa-based NLP engine to understand and execute commands. A modular, scalable system that is ideal for mobile deployment and smart assistant scenarios.

4.1 Framwork description

User Input Capture: A user speaks in the Libyan dialect (e.g., "شن حالك؟") then the system uses pyAudio to record the audio and save it as a .wav file at 16kHz, mono-channel, and 16-bit encoding.

Speech Recognition: That recorded audio is fed into the Whisper_Libyan_Model, a version of OpenAI's Whisper that has been fine-tuned on a Tripolitanian Arabic dataset. it transcribes the audio to text.

REST API Transmission: The transcribed content is passed to the Rasa server through a REST API, thereby making the NLP module independent of the ASR engine.

Natural Language Understanding (Rasa Server): The server runs input with DIETClassifier for intent identification, Regex extractor for entity extraction, Dialogue policies (TEDPolicy, RulePolicy) for controlling flow.

Custom Action Execution: Custom actions are executed to perform system tasks, such as app launch and Bluetooth toggling through rasa_sdk and Android API integration, after identifying intent and entities, If user input requires it.

Response Generation: The assistant sends a contextual response back to the user in text output, either in reply to the user input as a conversation or to confirm the action taken.

4.2 Innovation & Relevance

This framework is the first of its kind for Libyan Dialect. It:

- Combines state-of-the-art multilingual ASR with intent-based NLP
- Handles code-switched inputs
- Is adaptable for voice-controlled mobile applications
- Achieves real-world deployment potential for low-resource dialects

5. EXPERIMENTAL RESULTS

In this section, we provide an evaluation of the performance of our fine-tuned Whisper ASR model and the Rasa-based voice assistant using established performance metrics, visualization, and error analysis methods.

5.1 Evaluation Metrics

- ASR Metrics

$$\text{WER} = \frac{S + D + I}{N}$$

Where:

- S: number of substitutions
- D: deletions
- I: insertions
- N: total words in the reference

Word Error Rate (WER), a commonly used metric in speech recognition, is the primary measure that is used to evaluate Whisper's performance, this metric was calculated using the evaluate library from Hugging Face as described in [28].

- NLU Metrics

The chatbot evaluation report includes:

- Intent Accuracy

$$\text{Accuracy} = \frac{TP}{TP + FN}$$

- Entity Extraction F1-Score

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- True Positives (TP): Correctly classified instances.
- False Negatives (FN): Instances that should have been classified as a given class but were misclassified.
- False Positives (FP): Instances incorrectly classified as another class.

DOI: 10.5281/zenodo.15505226

- Recall measures how many actual positive cases were correctly predicted by the model.
- Precision measures how many of the items that the model labeled as positive are actually correct.

TEDPolicy Accuracy measures the correct prediction of dialogue actions, while the confusion matrix visualizes per-intent classification errors. The study uses rasa test and rasa evaluate CLI utilities to evaluate chatbots, as demonstrated in [3] .

5.2 ASR Performance (Whisper)

TABLE 1. RESULTS OF WHISPER FINE-TUNING

Metric	Baseline	Fine-Tuned
Word Error Rate (WER)	2.045	0.040
Training Loss (Final)	0.2098	0.0015
Validation Loss (Final)	0.1439	0.0012

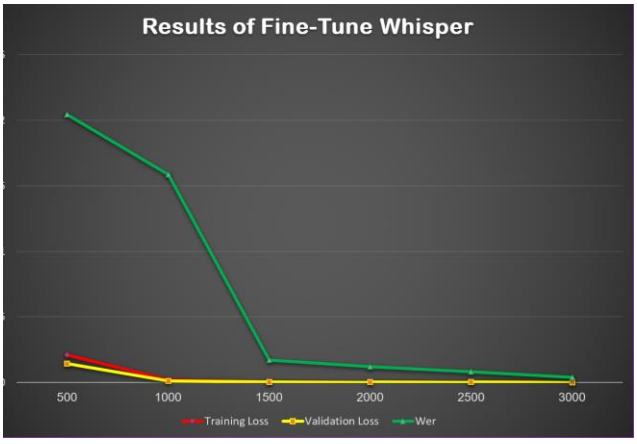

FIGURE 4. WHISPER FINE-TUNING PERFORMANCE METRICS. SHOWS WER, TRAINING, AND VALIDATION LOSS IMPROVEMENTS OVER TRAINING STEPS DURING ASR MODEL FINE-TUNING.

The study reveals that fine-tuning Whisper on a task-specific dialectal dataset can reduce WER by 98%, confirming previous low-resource ASR optimization findings, as shown in [29] and [30].

5.3 Rasa Chatbot Performance

TABLE 2. RESULTS OF RASA PERFORMANCE

Metric	Score
Intent Classification Acc.	100%
Entity F1-Score	0.998
TEDPolicy Accuracy	98.9%
Best Intent F1	0.995
Worst Intent F1	0.886

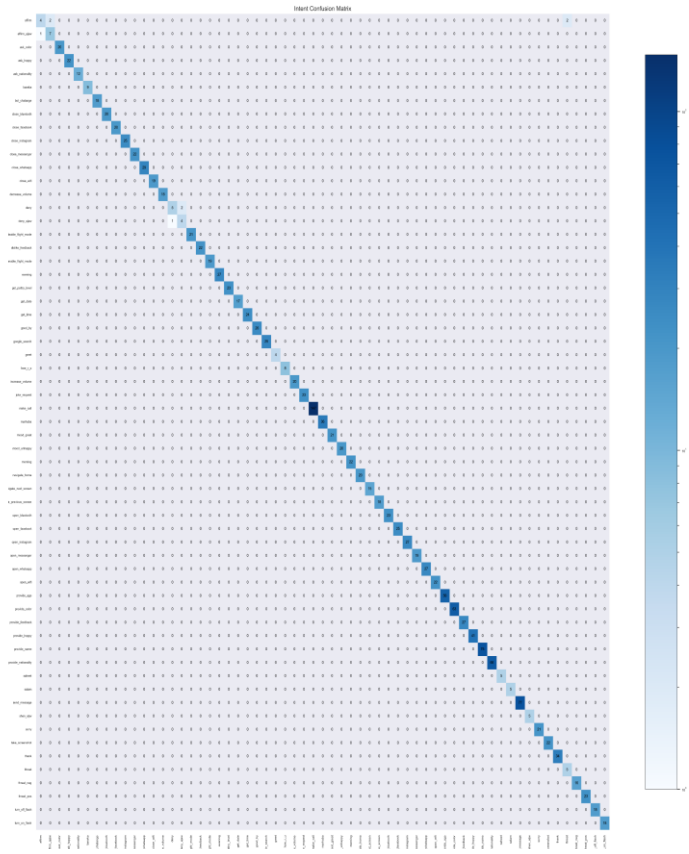

FIGURE 5. CONFUSION MATRIX FOR RASA INTENT CLASSIFICATION. VISUALIZES CLASSIFICATION ACCURACY AND MISCLASSIFICATIONS AMONG LIBYAN DIALECT INTENTS.

In order to evaluate the assistant's contextual dialogue management skills, we evaluated multi-turn situations which needed the chatbot to track user intent through turns and resolve ambiguity by asking clarifying questions as in Figure 6. For example:

USER→ **"شني الجو؟" :**In the dialect of Libya, it could mean "How are you?" or "How's the weather"?

Bot : "حيرتني راهو ماعرفتكش تنشد على جوي وإلا على الجو الجو؟! " → "تسأل فيا على جوي؟"You confused me. I don't know if you're asking about me or the weather?! Are you asking about me"?

User" → "طبعا ننشد على جوك" :Of course I am asking about you" or"→ "تي لا ننشد على عالجو" No, I'm asking about the weather", Based on the user's answer, the bot either identifies the "affirm_ajaw" intent then answers with his

condition in the first case "عالكيف بالك تبي نساعدك في حويجة" or identifies the "deny_ajaw" intent and executed a custom action to open the weather app and replays the message : "أهو فتحتلك التطبيق باش تشوف الجو" → "So I opened the application for you to check the weather".

These interactions demonstrate the assistant's multi-turn context tracking, and disambiguation capabilities, enabled by TEDPolicy and custom action handling within the Rasa framework.

FIGURE 6. RESOLVING AMBIGUITY BY PROMPTING FOR CLARIFICATION. CHATBOT HANDLES AMBIGUOUS USER INPUT BY ASKING CLARIFYING QUESTIONS.

6. DISCUSSION

Experimental results confirm the merits of task-specific fine-tuning and domain-aware NLU modeling for low-resource dialectal ASR systems. Trends in performance and how the design choices influence the accuracy of the system and also its generalizability, are discussed in the following section.

6.1 Whisper ASR Fine-Tuning Analysis

Figure 4 provides an overview of the complete WER, Training Loss and Validation Loss, the training process showed consistent reductions in the loss values during the interval from step 500 to step 3000. At step 500, the training loss was measured at 0.2098 and the validation loss at 0.1439, still indicating that the model was in a phase of adaptation to the new data domain. At step 3000, on the other hand, training loss went down to 0.0015, and the validation loss exponentially reduced to 0.0012. This indicates successful convergence and the very limited possibility for overfitting (see Figure 4).

Over time, the WER decreased dramatically, going from 2.045 at step 500 to 0.1725 at step 1500 and then to 0.0400 at step 3000. This pattern suggests that the model quickly learned phonological patterns specific to Tripolitanian Arabic by adapting to the dialect-specific data [30].

This is replicating what is seen in similar low-resource adaptation experiments, in which noise augmentation and domain-matched data yield rapid loss reduction and steep WER decline, as shown in [29,26]. Also, the very small gap between training and validation loss is a sign of very good generalization: the model is performing equally well on seen and unseen data.

Moreover, a total final WER of 0.04 (4%) as shown in Table 1, indicates the model makes just 4 errors per 100 words in transcription—a performance that was rated very high for any practical ASR system in any realistic conditions, especially in dialectal and noisy conditions, as pointed in [2,18].

When a detailed examination of Whisper's output was performed, some transcription issues were found. The transcription of numbers is one notable problem. Even when contextual cues indicate a linguistic rather than numerical representation, in most cases, Whisper converts spoken numbers into digits instead of words, such as:

[t-con-0029.wav]
REF: تلاتين عام → Thirty Years
HYP: 30 عام
WER: 0.50, CER: 0.60

Reflecting the replacement of the spoken number with digits.

[t-con-0055.wav]
REF: انا مواليد حداش ستة ألفين وأربعة → I was born on Eleven/six/Two thousand and four
HYP: 2004-6-11 انا مواليد
WER: 0.66, CER: 0.33

In like manner, time references such as حداش ستة ألفين وأربعة" was transcribed as "11-6-2004", indicating Whisper's preference for numeric formats over lexical ones.

These outputs resulted in high word and character error rates. But, in some cases, Whisper accurately transcripts the spoken numbers as a word:

[t-num-0025.wav]
REF: خمسة وعشرين
HYP: خمسة وعشرين
WER: 0.00, CER: 0.00

Another challenge includes the pronunciation of the letter "ق" (qāf), which is transcribed as [g] in Libyan dialect, and it is represented by the CODA* system as "ق" not "ج". Whisper occasionally misinterprets this phoneme, producing outputs consistent with non-Libyan dialect pronunciations, as seen in:

t-fhq-0122.wav where "وقاعد" ("and still") was rendered as "وجاعد".

[t-fhq-0122.wav]
REF: وقاعد حتى الخدمة مخدمش → It still doesn't work yet.
HYP: وجاعد حتى الخدمة مخدمش
WER: 0.25, CER: 0.05

Also, in some cases, Whisper correctly maintains the Libyan dialect pronunciation of (qāf), such as in "قلتلك".

[t-fhq-0276.wav]
REF: اكتر من اللي قلتلك عليهم → More than those I told you about.
HYP: اكتر من اللي قلتلك عليهم
WER: 0.00, CER: 0.00

6.1.1 Avoiding Overfitting and Underfitting

- When training and validation losses are both high, underfitting is typically identified. In this instance, underfitting was ruled out because the model achieved values close to zero on both.
- When validation loss is high but training loss is low, overfitting usually appears, as demonstrated in [19].

However, both validation and training loss as appeared in Figure 4, decreased together in work results—this indicates that the model is learning patterns instead of simply memorizing training data. These indicate a robust and well-regularized model pipeline, which is made possible by:

- Limited but domain-rich training data.
- Silent trimming and segmentation.
- Augmented data diversity.
- A carefully configured training process.

6.1.2 Impact of Code-Switching and Noise Robustness

The authors in [2,9] demonstrated that one of the critical design elements is being able to recognize code-switched, The fine-tuned Whisper model was able to be fluent in recognizing hybrid expressions through its recognizing ability, and since the model was learned from multilingual text, it improved speaker utility for Libyans who regularly code-switched Arabic with English or Italian loanwords, as shown in [7].

Notably, Whisper demonstrates strong performance in handling code-switching between Arabic and English, which is common in Libyan spoken language. For example:

[t-cmd-0115.wav]
REF: دير search على سعر الدولار → Search for the dollar price
HYP: دير search على سعر الدولار → Search for the dollar price
WER: 0.00, CER: 0.00

In the above case, phrase "دير search على سعر الدولار" ("Do a search for the dollar price") was transcribed perfectly, achieving perfect WER and CER scores. These findings align with Whisper's multilingual capabilities as reported by Radford et al. (2023), who noted its robustness in multilingual and code-switched scenarios

Moreover, the dataset was used in 7 different noise environments (such as street noise, café noise, classroom noise, and reverb) to enhance the model's ability to transcribe speech in everyday settings [26,27].

While the complete dataset cannot yet be released due to privacy and licensing constraints, a curated subset of the Tripolitanian Arabic dataset along with example YAML training files for Rasa will be made available upon request. To support reproducibility and encourage further research on Libyan Arabic ASR.

6.2 Rasa NLU & Dialogue Evaluation

Strong performance was also shown by the NLU and dialogue models of the Rasa assistant. According to the findings of TEDPolicy and DIETClassifier as appeared in Table 2. The analysis at the intent level, based on the F1 evaluation and confusion matrix:
Class C3 obtained F1 = 0.995.
Moderate variance could not deter Class C8 from scoring highly with an F1 = 0.981.
The lowest F1 = 0.886 related to Class C4 (confirm and deny intents), which, it seems, was impressed by the lack of possibilities and examples of them. This is because the number of examples is limited and contains only affirmation and denial intentions that limit their further development.

However, they do not change the accuracy of intention recognition in the context of its usage, such as yes or no confirming or negating the user's emotional state (Are you happy?).

The NLU pipeline of Rasa's test with cross-validation folds for precision-recall evaluation was used. This

analysis has aligned with research on low-resource NLU warnings pointing out the availability of double meanings in the semantics of confirmation/clarification intents, which makes them error-prone, as demonstrated in [23] and [22].

To go beyond aggregate metrics like F1-score, an intent-level confusion matrix was created using 60 Libyan dialect intents. The majority of the intents were classified with high precision and minimal confusion, as seen in Figure 5. However, some misclassifications were observed, most prominently the overlap between threat, affirm, and affirm_ajaw, also between deny and deny_ajaw. This was due to semantic similarity and limited variation in example utterances like" → "باهي, إيه" Ok, yes" for affirm" → "باهي باهي",Woe to you!" for threat, and إيه" " → "عالجوYes, about the weather" for affirm_ajaw. For example, out of 8 samples labeled as affirm, 2 were misclassified as affirm_ajaw and 2 as threat.

Excellent separation can be observed in the custom intents that related to app actions, such as open_facebook, open_instagram, close_whatsapp, and open_whatsapp, suggesting strong intent disambiguation performance.

The confusion matrix indicated areas for semantic disambiguation improvement while confirming the DIETClassifier's robustness .This analysis supports the power of the NLU model for practical deployment in domain-specific contexts and enriching low-resource intents with more contextually diverse training data.

6.3 Usability Evaluation Using CUQ

This study used the Chatbot Usability Questionnaire (CUQ), a specialized tool with 16 Likert-scale items that aims to evaluate key usability factors in conversational interfaces (Holmes et al., 2019)., to evaluate the usability of the created chatbot. The questionnaire was completed by 79 participants, 46% were female and 54% were male. Participants' ages ranged from 18 to 63. Students, working professionals, and retirees made up an equal portion of the participants. Each user ranged from a beginner to an expert in terms of technical knowledge. The generalizability of usability scores is supported by this diversity, obtained CUQ scores that ranged from 60.94 to 93.75, with a mean score of 86.85 and a standard deviation of 4.33.

Although the CUQ does not yet have officially established benchmark thresholds, its structure is closely aligned with the well-established System Usability Scale (SUS). According to Lewis and Sauro (2018), SUS scores above 80 are typically interpreted as indicative of high usability, aligning with an A− grade on industry grading curves. The chatbot demonstrated strong perceived usability among the users based on the observed CUQ scores.

According to a score distribution histogram, as in Figure 7, the majority of participants clustered in the 85–90 range, with only one outlier scoring lower than 70 and the highest concentration (n = 25) in the 88–90 range .

Taken together, these results highlight the effectiveness of chatbots in providing user-friendly, intuitive, and reliable interaction. It is also consistent with previous literature that emphasizes the importance of clarity, trust, and conversational flow in chatbot design.

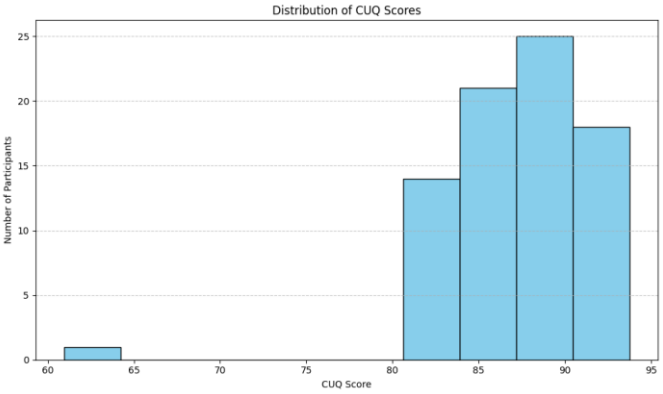

Figure 7. Histogram of Distribution of CUQ scores among 79 participants

7. Conclusion

This work has introduced an original and complete end-to-end voice assistant framework designed for the Tripolitanian language of Libyan Arabic that targets one of the most under-served sections of Arabic NLP and speech technology. Our contributions are in three main areas:

- Dialect-Specific Dataset Creation: We constructed a Libyan speech dataset that is selectively curated, phonetically rich, and generally evenly distributed among the demographics. The dataset contains over 13,000 utterances and is annotated using the CODA* annotation system and augmented to be noise robust and provide diversity.
- Whisper Model Fine-Tuning: Using Hugging Face's Seq2SeqTrainer, we optimized OpenAI's Whisper Medium model, lowering the WER by 98%, from 2.045 to 0.040. Both clean and noisy speech situations showed the same decrease in inaccuracy.
- Voice-Enabled Chatbot Integration: We trained a Rasa-based assistant on our determined domain-specific intents and entities. The assistant achieved 100% accuracy on intent classification and a 0.998 F1-

score for entity extraction. It also provides the ability for natural-level spoken commands, including code-switching, that is encoded in Arabic, English, and Italian.

This framework not only serves as a proof-of-concept for the feasibility of building voice-first AI systems for low-resource dialects but also provides a replicable reference point for researchers and developers working in the parallel context of low-resource languages and dialects.

8. Future Work

A promising direction for future work involves the development of a Libyan Arabic speech synthesis system, which would be based on the current dataset used in this study. the project aims to support the generation of natural and intelligible synthetic speech in the Libyan dialect. This development aims to improve the chatbot's responses to become voice responses. And also, future directions include:

- We aim to add Libyan Arabic dialects from Misrata, Benghazi, and Sabha to the dataset in future versions. This includes recording spontaneous, multi-turn dialogue in (cafés, street interviews), multi-turn dialogue. Over 100,000 utterances from more than 100 speakers with various sociolinguistic backgrounds is our goal. And use it to train (ASR) and chatbot modules.
- Real-Time Mobile Deployment: Optimizing the inference speed and model size for on-device execution.
- Further enhance the general conversation portion and leverage new technologies that enhance dialogue for smoother, more contextually aware conversations beyond short conversations.
- Public Dataset Release: Releasing a subset of the speech dataset and training configurations to promote open research on Libyan Arabic.
- To evaluate real-time deployment results, future research will include more UX metrics, such as task completion rate and spoken interaction logs.

References

[1] M. Habash, “CODA: Conventional Orthography for Dialectal Arabic,” Proc. LREC, 2018.
[2] OpenAI, “Whisper: Robust Speech Recognition via Weak Supervision,” OpenAI Technical Report, 2023. [Online]. Available: https://openai.com/research/whisper
[3] Rasa Technologies, “Rasa Open-Source Documentation,” 2024. [Online]. Available: https://rasa.com/docs/
[4] K. Warden, "The Rise of Voice Assistants," IEEE Spectrum, 2021.
[5] H. Alshamrani et al., “Arabic Speech Recognition Challenges,” Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 6, pp. 590–595, 2020.
[6] M. Bouamor et al., "MADAR Arabic Dialect Corpus," LREC, 2018.
[7] J. Owens, A Grammar of Libyan Arabic: Sarha, Harrassowitz, 1993.
[8] A. Abu-Melhim, The Arabic Dialect of Libya, Lincom Europa, 2002.
[9] S. Gandhi, “Fine-Tune Whisper for Multilingual ASR,” Hugging Face Blog, 2023. [Online]. Available: https://huggingface.co/blog/fine-tune-whisper
[10] Y. Zhang et al., “Pushing the Limits of Semi-Supervised ASR,” arXiv:2010.10504, 2020.
[11] M. Habash et al., “Obstacles for Arabic NLP,” in Proc. 58th Annu. Meet. Assoc. Comput. Linguistics (ACL), 2020, pp. 7740–7755.
[12] A. Ali et al., “Speech Recognition for Arabic: A Comparative Study,” IEEE TASLP, vol. 22, no. 12, 2014.
[13] M. Hsu et al., “A Call for Dialect-Aware Arabic NLP,” EMNLP Workshop, 2022.
[14] V. Ritt-Benmimoun, Ed., Tunisian and Libyan Arabic Dialects: Common Trends – Recent Developments – Diachronic Aspects, Zaragoza, Spain: Prensas de la Universidad de Zaragoza, 2017.
[15] M. Elmahdy, A. Ali, and T. Schultz, “Investigating Arabic Dialects in Egyptian ASR,” Proceedings of the International Workshop on Spoken Language Translation (IWSLT), pp. 1–8, 2014. [Online]. Available: https://aclanthology.org/2014.iwslt-papers.1.pdf
[16] N. Halabi, “The Arabic Speech Corpus,” University of Southampton, 2016. [Online]. Available: https://arabicspeechcorpus.com/
[17] A. Malmasi et al., “Arabic Dialect Identification Challenge 2017,” VarDial, 2017.
[18] Sehar, N. U., Khalid, A., Adeeba, F., & Hussain, S. (2025, January 1). Benchmarking Whisper for Low-Resource Speech Recognition: An N-Shot Evaluation on Pashto, Punjabi, and Urdu. ACL Anthology. https://aclanthology.org/2025.chipsal-1.20/
[19] Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics Conference Series*, *1168*, 022022. https://doi.org/10.1088/1742-6596/1168/2/022022
[20] L. Tagliasacchi et al., “ASR in Low-Resource Languages,” ICASSP, 2022.
[21] T. Bocklisch et al., “Rasa: Open Source Conversational AI,” NeurIPS Workshop, 2017.
[22] Alruily, M. (2022). ARRASA: Channel Optimization for Deep Learning-Based Arabic NLU Chatbot

Framework. *Electronics*, *11*(22), 3745. https://doi.org/10.3390/electronics11223745
[23] N. Habash, et al., (2020). A Panoramic Survey of Natural Language Processing in the Arab World. 10.48550/arXiv.2011.12631.
[24] M. Bouamor et al., "Arabic NLP in Commercial Voice Assistants," LREC, 2020.
[25] Valk, J., & Alumäe, T. (2020, November 25). VoxLingua107: a Dataset for Spoken Language Recognition. arXiv.org. [2011.12998] VoxLingua107: a Dataset for Spoken Language Recognition
[26] D. Park et al., "SpecAugment: A Simple Data Augmentation Method for ASR," Interspeech, 2019.
[27] Y. Shi et al., Noisy training for deep neural networks in speech recognition. EURASIP Journal on Audio, Speech, and Music Processing. 2015. 10.1186/s13636-014-0047-0.
[28] Hugging Face, "Evaluate: Metrics for Machine Learning," GitHub Repository, 2023. [Online]. Available: https://github.com/huggingface/evaluate
[29] D. Amodei et al., "Deep Speech 2: End-to-End ASR," ICML, 2016.
[30] A. Gulati et al., "Conformer: Convolution-Augmented Transformer for ASR," INTERSPEECH, 2020.[5] K. St. Amant, "Virtual office communication protocols: A system for managing international virtual teams," in *Proc. IEEE Int. Professional Commun. Conf.*, 2005, pp. 703–717.
[31] Radford, A., et al. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. OpenAI. https://cdn.openai.com/papers/whisper.pdf
[32] Preprints.org. (2022). Wav2vec2.0 vs. Whisper: A Comparative Study. https://www.preprints.org/manuscript/202212.0426/v1download
[33] Anidjar, O. H., Marbel, R., & Yozevitch, R. (2024). Whisper Turns Stronger: Augmenting Wav2Vec 2.0 for Superior ASR in Low-Resource Languages. arXiv:2501.00425. https://arxiv.org/abs/2501.00425
[34] Talafha, B., et al. (2023). N-Shot Benchmarking of Whisper on Diverse Arabic Speech. Interspeech 2023. https://www.isca-archive.org/interspeech_2023/talafha23_interspeech.pdf
[35] Samuel Holmes, Anne Moorhead, Raymond Bond, Huiru Zheng, Vivien Coates, and Michael Mctear. 2019. Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces? In Proceedings of the 31st European Conference on Cognitive Ergonomics (ECCE 2019), Maurice Mulvenna and Raymond Bond (Eds.). ACM, New York, NY, USA, 207-214.
[36] Lewis, J. R., & Sauro, J. (2018). Item benchmarks for the system usability scale. Journal of Usability Studies Archive, 13(3), 158–167. (PDF) Item Benchmarks for the System Usability Scale