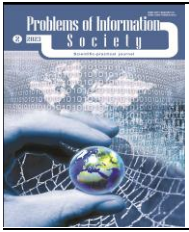

www.jpis.az

14 (2)
2023

Development of a real-time speech recognition system for the Azerbaijani language

Alakbar T. Valizada

Azerbaijan Technical University, H.Javid ave 25, AZ 1073, Baku, Azerbaijan

alakbar.valizada@aztu.edu.az

orcid.org/0009-0001-9880-292X

ARTICLE INFO

http://doi.org/10.25045/jpis.v14.i2.07

Article history:
Received 8 February 2023
Received in revised form
11 Aprel 2023
Accepted 13 June 2023

Keywords:
Hybrid Speech Recognition
N-gram
Kaldi
Agglutinative languages
Real-time speech recognition
WebSocket

ABSTRACT

This paper investigates the development of a real-time automatic speech recognition system dedicated to the Azerbaijani language, focusing on addressing the prevalent gap in speech recognition system for underrepresented languages. Our research integrates a hybrid acoustic modeling approach that combines Hidden Markov Model and Deep Neural Network to interpret the complexities of Azerbaijani acoustic patterns effectively. Recognizing the agglutinative nature of Azerbaijani, the ASR system employs a syllable-based n-gram model for language modeling, ensuring the system accurately captures the syntax and semantics of Azerbaijani speech. To enable real-time capabilities, we incorporate WebSocket technology, which facilitates efficient bidirectional communication between the client and server, necessary for processing streaming speech data instantly. The Kaldi and SRILM toolkits are used for the training of acoustic and language models, respectively, contributing to the system's robust performance and adaptability. We have conducted comprehensive experiments to test the effectiveness of our system, the results of which strongly corroborate the utility of the syllable-based subword modeling approach for Azerbaijani language recognition. Our proposed ASR system shows superior performance in terms of recognition accuracy and rapid response times, outperforming other systems tested on the same language data. The system's success not only proves beneficial for Azerbaijani language recognition but also provides a valuable framework for potential future applications in other agglutinative languages, thereby contributing to the promotion of linguistic diversity in automatic speech recognition technology.

1. Introduction

Speech Recognition System (ASR) has emerged as a pivotal technology in the contemporary digital landscape, finding utility in myriad applications, from transcription services and virtual assistants to the broader domain of human-computer interaction. The escalating demand for real-time speech recognition in an array of fields underscores the importance of engineering ASR systems that can efficiently process and transcribe spoken language with minimal delay. This demand gains additional gravity when considered in the context of under-resourced languages, where ASR technology is often underdeveloped or absent. Among these languages is Azerbaijani, an agglutinative language with intricate morphological structures, which presents distinct challenges to ASR systems due to its complex nature.

Historically, Azerbaijani ASR research has predominantly emphasized offline speech recognition systems, typically custom-built for niche applications such as emergency call centers (Valizada et al., 2021) and taxi call service systems (Rustamov et al., 2019). While these dedicated offline systems have undoubtedly advanced the field and provided valuable insights, the burgeoning demand for real-time ASR systems brings to the forefront the urgency for Azerbaijani speech recognition solutions that offer both accuracy and real-time response.

Addressing this critical need, this paper presents a novel real-time ASR system explicitly developed for the Azerbaijani language. The system is meticulously designed to process streaming speech data and hinges on a WebSocket-based communication framework. This choice of framework fosters efficient transmission of speech data in real-time between the user-end clients and the recognition server, an essential characteristic for any effective real-time ASR system.

Underpinning the system's architecture is a robust hybrid of Hidden Markov Model (HMM) and Time-Delay Neural Network (TDNN) utilized for acoustic modeling. Additionally, a syllable-based n-gram model is incorporated for language modeling. The fusion of these approaches enables the system to aptly handle the complexities inherent to the Azerbaijani language, which often boasts a high degree of unique word forms due to its agglutinative nature.

The implementation of this innovative ASR system is rooted in the use of the Kaldi (Povey et al., 2011) and SRILM toolkits (Stolcke, 2002) for model training and subsequent evaluation. This paper outlines the detailed experimental setup, discussing the data preparation process, model training, and evaluation metrics. It further presents the resulting performance metrics, emphasizing the system's effectiveness in delivering real-time speech recognition for the Azerbaijani language.

The paper is systematically organized into subsequent sections to facilitate understanding: Section 2 provides a comprehensive overview of related work in the ASR field, particularly focusing on WebSocket-based communication technology. Section 3 delves into the detailed experimental setup, results, and performance metrics, including crucial aspects such as data preparation and model training. Finally, Section 4 concludes the paper, offering a summarization of the study's findings and delineating potential future work in this exciting field.

2. System Overview

2.1 Architecture

The proposed system consists of the following components:

1) Client-side application: Records user speech and streams audio data to the server using WebSocket.
2) Server-side application: Receives audio data, performs speech recognition using a hybrid model – HMM and Deep Neural Network (DNN) – HMM/DNN, and sends recognition results back to the client.
3) Acoustic model: A hybrid HMM/DNN model trained using the Kaldi toolkit.
4) Language model: An n-gram language model created using the SRILM toolkit.

2.2 WebSocket Implementation

WebSocket is a protocol that enables bidirectional, low-latency communication between a client and a server over a single, long-lived connection (Figure 1) (MDN contributors, 2023). This protocol is particularly suitable for real-time applications such as speech recognition, where data needs to be streamed continuously from the client to the server, and recognition results need to be returned promptly. In this section, we will discuss the server-side and client-side implementations of WebSocket for the proposed real-time speech recognition system for Azerbaijani.

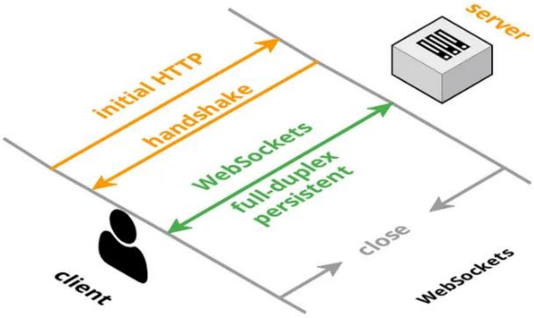

Fig. 1. High-level overview of a WebSocket communication

The proposed system consists of the following components:

Client-Side Implementation

a) Recording audio data: The client application, which can be a web or mobile application, initiates a WebSocket connection to the server by providing the server's URL and the desired connection protocol. Upon establishing the connection, the client application maintains the connection throughout the user's interaction with the system.

The client-side application records the user's speech using the device's microphone and an appropriate audio recording API. The recorded audio data is typically stored in a buffer, which is a temporary storage area in memory.

To segment the audio data into smaller chunks, the client application sets up a timer or an event-based callback mechanism that triggers at regular intervals, say every T milliseconds. When the timer or event is triggered, the client application retrieves the audio data from the buffer corresponding to the specified time interval.

For example, if the buffer size is set to 16,000 samples (corresponding to 1 second of audio at a 16 kHz sampling rate), and the timer interval is set to 250 milliseconds, the client application would retrieve 4,000 samples (250 ms * 16 samples/ms) from the buffer at each interval. These samples represent an audio chunk that is ready for transmission.

*b) **Sending audio data:*** Once the audio chunk has been prepared, the client application sends it to the server over the WebSocket connection. The audio data is typically encoded into a binary format, such as pulse-code modulation (PCM), which preserves the original audio information and ensures efficient transmission.

Each WebSocket message sent to the server includes the binary-encoded audio chunk and any necessary metadata, such as the client's session ID, the current audio timestamp, or a sequence number to maintain the order of the chunks. The metadata can be sent as a separate message or included within the binary message, depending on the implementation.

*c) **Receiving recognition results:*** The client application listens for incoming messages from the server, which contain the recognition results. Upon receiving a message, the client application decodes the JavaScript Object Notation (JSON) object and updates the user interface to display the recognized speech in real-time.

Server-Side Implementation. On the server side, the application receives the incoming WebSocket messages containing the audio chunks and metadata. The server is responsible for decoding the messages, extracting the audio data, and appending it to the appropriate audio buffer associated with the client's session ID.

As the audio chunks are received in the correct order (based on the sequence number or timestamp), the server application can concatenate them to reconstruct the original audio stream. The server maintains an audio buffer for each client session, ensuring that the audio data is correctly processed for each user.

When a sufficient amount of audio data has been accumulated in the buffer, the server processes the buffered audio using the ASR engine, which performs the speech recognition task. The length of the buffered audio can be determined by a fixed duration (e.g., 1 second), or adaptively based on the system's requirements and performance.

Once the ASR engine completes the recognition task, the server prepares the recognition results as a JSON object and sends it back to the corresponding client over the WebSocket connection, allowing the client to receive the results in real-time.

This process of segmenting audio data into chunks, transmitting them over WebSocket, and server-side collection and processing ensures that the real-time speech recognition system for Azerbaijani can efficiently and accurately recognize speech in a real-time scenario.

Acoustic model. The hybrid HMM/DNN architecture used in this study combines the strengths of HMMs and DNNs for acoustic modeling.

*a)**HMM Component***

The HMM component is responsible for modeling the temporal structure of speech using a sequence of hidden states, $= q_1, q_2, \dots, q_T$, and observations, $O = o_1, o_2, \dots, o_T$. *O* observations represent the sequence of *T* acoustic feature vectors extracted from the speech signal at regular time intervals. *Q* hidden states represent the sequence of *T* underlying phonetic units or triphones corresponding to the observed acoustic features. The hidden states are not directly observable, and the goal of the HMM-based ASR system is to determine the most likely sequence of hidden states given the observed acoustic features (Rabiner, 1989).

The HMM is defined by the following parameters:

- State transition probabilities (*A*): $a_{ij} = P(q_t = j | q_{t-1} = i)$ represents the probability of transitioning from state *i* to state *j*.
- Emission probabilities (*B*): $b_j(o_t) = P(o_t | q_t = j)$ represents the probability of observing feature vector o_t at time *t*, given the state *j*.
- Initial state probabilities (*π*): $\pi_i = P(q_1 = i)$, represents the probability of starting in state *i*.

The likelihood of a given observation sequence, *O*, can be calculated using the forward algorithm, which computes the forward probability, $\alpha_t(j)$, as follows:

$$\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = j | \lambda), \quad (1)$$

where *λ = (A, B, π)* are the HMM parameters.

b) ***DNN Component***

While HMMs are effective in modeling the temporal aspects of speech, they are limited in their ability to model the complex relationships between the acoustic features and the phonetic units. DNNs have proven to be highly effective in modeling these complex relationships due to their ability to learn hierarchical representations of the input features (Hinton et al., 2012).

In the hybrid HMM/DNN architecture, the DNN is used to model the observation likelihoods *$b_j(o_t)$* for each HMM state *j*. Specifically, the DNN is trained to predict the posterior probabilities *$P(q_t=j|o_t)$* of the HMM states given the acoustic features *o_t*. Once trained, the DNN can be used to estimate the observation likelihoods for the HMM by applying Bayes' rule:

$$b_j(o_t) = P(o_t|q_t = j) = P(q_t = j|o_t) * P(o_t)/P(q_t = j), \quad (2)$$

where *$P(o_t)$* and *$P(q_t=j)$* are the priors for the observations and states, respectively.

By combining the strengths of HMMs and DNNs, the hybrid HMM/DNN architecture allows for more accurate modeling of the complex relationships between the acoustic features and phonetic units in speech signals, leading to improved recognition performance.

For the training of the AM, we used TDNN architecture (Peddinti et al., 2015) with maximum mutual information (MMI) sequence-level objective function (Povey et al., 2016). TDNN is a type of artificial neural network where the neurons have time delay connections. It is an extension of the feed-forward network, designed to recognize patterns in sequences of inputs spread out over time. TDNNs are specifically designed to address the variability of speech signals by applying a time-delay structure to the DNN's hidden layers. This allows the network to learn context-dependent features while reducing the effects of distortions and variations in the temporal alignment of the speech signal. The integration of TDNN within this hybrid framework leverages the time-delay structure to handle temporal variability in the speech signals and captures complex patterns within the acoustic features (Figure 2).

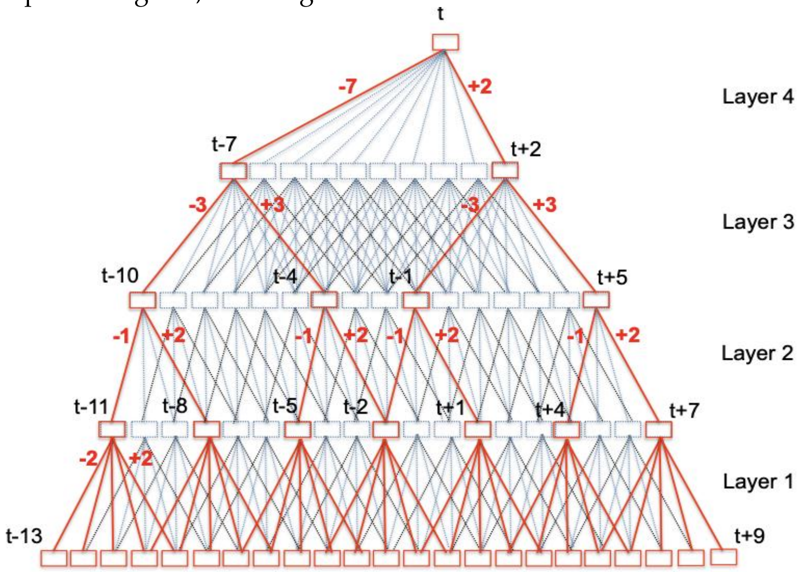

Fig. 2. Structure of Time Delay Neural Network (TDNN)

Hence, the utilization of a TDNN in a hybrid DNN/HMM ASR architecture can provide a powerful, context-sensitive, and highly accurate system for real-time speech recognition in Azerbaijani. This is particularly significant given the rich morphological structure and agglutinative nature of the Azerbaijani language, which present unique challenges for ASR systems. The TDNN can capture the context-dependent morphological variations in the language, resulting in better recognition performance.

Language Modeling. The n-gram language model computes the probability of a sequence of words, $W = w_1, w_2, \dots, w_n$, based on the joint probability of the words (Jurafsky et al., 2009). Using the chain rule of probability, we can write: The n-gram language model computes the probability of a sequence of words, $W = w_1, w_2, \dots, w_n$, based on the joint probability of the words. Using the chain rule of probability, we can write:

$$P(W) = P(w_1, w_2, \dots, w_n) = \prod_{i=1}^{n} P\,(w_i|w_{i-1}, w_{i-2}, \dots, w_1), \quad (3)$$

for an n-gram model, the probability of a word depends only on the previous n-1 words:

$$P(w_i|w_{i-1}, w_{i-2}, \dots, w_1) \approx P(w_i|w_{i-1}, w_{i-2}, \dots, w_{i-n+1}), \quad (4)$$

When dealing with larger values of n in n-gram, it is common to encounter n-grams that have not been observed in the training data, resulting in zero probabilities. Kneser-Ney smoothing, and its modified version, aim to solve this problem by assigning non-zero probabilities to these unseen n-grams. The basic idea of Kneser-Ney smoothing is to consider not just the frequency of a word (or an n-gram), but also the number of different contexts in which that word (or n-gram) has appeared. So a word that appears frequently but always in the same context will get a lower probability than a word that appears less frequently but in many different contexts (Chen et al., 1999).

Here's how modified Kneser-Ney smoothing works:

1. **Absolute Discounting:** The first step is absolute discounting, where a fixed amount (usually determined by cross-validation) is subtracted from the observed counts of n-grams. This helps to reserve some probability mass for unseen n-grams.
2. **Lower-Order Models:** The probability mass reserved from higher order n-grams (for example, trigrams or bigrams) is distributed to lower order n-grams (like unigrams or bigrams).
3. **Continuation Probabilities:** The unique feature of Kneser-Ney smoothing is that it uses continuation probabilities for distributing the reserved probability mass. Continuation probability of a word is the number of different contexts in which the word has appeared as a continuation. This means, for example, that a word that often appears as the continuation of different bigrams will have a high unigram continuation probability.
4. **Interpolation:** Finally, an interpolation of different order n-gram models is used to calculate the probabilities. This means that the probability assigned to an n-gram is a weighted sum of the discounted absolute frequency of the n-gram and the lower order n-gram continuation probabilities.

The modified Kneser-Ney smoothing thus overcomes data sparsity by not only considering the frequency of n-grams but also taking into account the number of different contexts in which a word appears. This makes it one of the most effective smoothing techniques for n-gram language modeling.

3. Experimental Setup and Results

Data Preparation

1. Acoustic Model Data

For the acoustic model, audio data were collected from two main sources. The first source was a Telegram bot, which allowed users to contribute speech samples through a crowdsourcing approach. Users were prompted to record and submit phrases from a predefined list, ensuring the diversity and coverage of the collected data. The second source of audio data was from YouTube videos. Videos containing Azerbaijani speech were selected, and the audio tracks were extracted for further processing. To create a balanced dataset, both sources of data were combined, segmented, and annotated with corresponding transcriptions. The resulting dataset was then split into training, validation, and testing sets to enable the training and evaluation of the HMM/TDNN acoustic model.

2. Language Model Data

The language model was trained using text data collected from two main sources: Azerbaijani Wikipedia (Wikimedia, 2023) and the AzerTag news portal (The Azerbaijan State News Agency, 2022). These sources were chosen due to their extensive coverage of various topics, which ensured the language model would be exposed to a wide range of vocabulary and linguistic structures. The collected text data were cleaned, tokenized, and processed to generate an n-gram language model using the SRILM toolkit (Stolcke, 2002).

A significant modification was made to the language modeling approach by adopting a syllable-based subword model instead of a traditional word-based model. This approach was chosen due to the agglutinative nature of the Azerbaijani language, which often results in a large number of unique word forms. By using a syllable-based subword model, the system can better handle out-of-vocabulary words and achieve more accurate recognition results (Valizada, 2021).

Acoustic and Language Model Training. The experiments described in this paper were carried out using both the Kaldi and SRILM toolkits. The configuration of the Kaldi speech recognition system is based on the Kaldi chain recipe (s5_r2) of the TED-LIUM corpus, which integrates a hybrid DNN/HMM system. The "nnet3 chain" implementation was utilized for training the acoustic model (AM), employing a TDNN architecture with MMI sequence-level objective function. The context-dependent states, acquired through the forced alignments of a GMM/DNN baseline system, served as targets for DNN training. Additionally, online i-vectors were used as input to the TDNN for improved speaker adaptation.

For the language model training, the SRILM toolkit was employed to train various language models (LMs). Based on our prior experience (Valizada et al., 2021), modified Kneser-Ney smoothing has proven to be the most effective method for handling out-of-vocabulary (OOV) words and large text corpora.

Evaluation Metrics. The system was evaluated using the Word Error Rate (WER) and real-time factor (RTF) metrics. The WER measures the recognition accuracy, while the RTF indicates the efficiency and speed of a ASR system.

Results. The proposed system achieved a WER of 5.59%, which is competitive with state-of-the-art ASR systems for Azerbaijani. The system's performance can be further improved by incorporating additional data sources and refining the acoustic and language models.

The real-time factor (RTF) of the proposed system was found to be 0.165, indicating that the system processes the input speech efficiently, making it suitable for real-time applications. The use of WebSockets for streaming speech data and the optimized hybrid HMM/DNN model contribute to the system's efficiency.

4. Conclusion and Future Work

This paper presents a real-time speech recognition system for the Azerbaijani language, leveraging WebSockets for data streaming and a hybrid HMM/DNN model for ASR. The proposed system demonstrates promising results in terms of both accuracy and efficiency, making it a valuable contribution to the field of ASR for Azerbaijani. The Kaldi toolkit and SRILM toolkit were used for training the system, providing a solid foundation for further research and development.

Future work can focus on improving the system's performance by incorporating additional data sources, refining the acoustic and language models, and exploring alternative ASR architectures such as end-to-end neural models. Additionally, the system can be extended to support multilingual and code-switching scenarios, broadening its applicability and usefulness in real-world applications.

In conclusion, the proposed real-time speech recognition system for Azerbaijani offers a robust and efficient solution for ASR tasks, paving the way for a range of applications such as transcription services, voice assistants, and real-time translation.

References

Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. Computer Speech & Language, 13(4), 359–393. https://doi.org/10.1006/csla.1999.0128

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A. R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. IEEE Signal Processing Magazine, 29(6), 82–97. https://doi.org/10.1109/msp.2012.2205597

Jurafsky, D., & Martin, J. H. (2009). Speech and Language Processing (2nd ed., pp. 83-122). Prentice Hall.

MDN contributors. *The WebSocket API*. Retrieved May 13, 2023, https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

Peddinti, V., Povey, D., & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. Interspeech 2015. https://doi.org/10.21437/interspeech.2015-647

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N.K., Hannemann, M., Motlícek, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., & Veselý, K. (2011). The Kaldi Speech Recognition Toolkit.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., & Khudanpur, S. (2016). Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. Interspeech 2016. https://doi.org/10.21437/interspeech.2016-595

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257–286. https://doi.org/10.1109/5.18626

Rustamov, S., Akhundova, N., & Valizada, A. (2019). Automatic Speech Recognition in Taxi Call Service Systems. Automatic Speech Recognition in Taxi Call Service Systems | SpringerLink. https://doi.org/10.1007/978-3-030-23943-5_18

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. 7th International Conference on Spoken Language Processing (ICSLP 2002). https://doi.org/10.21437/icslp.2002-303

The Azerbaijan State News Agency (2022). [Dataset]. https://azertag.az/

Valizada, A. (2021). Subword Speech Recognition for Agglutinative Languages. 2021 IEEE 15th International Conference on Application of Information and Communication Technologies (AICT). https://doi.org/10.1109/aict52784.2021.96204

Valizada, A., Akhundova, N., & Rustamov, S. (2021). Development of Speech Recognition Systems in Emergency Call Centers. MDPI. https://doi.org/10.3390/sym13040634

Wikimedia, (2023). azwiki dump (20230320) [Dataset]. https://dumps.wikimedia.org/azwiki/20230320/