TURKISH SPOKEN QUESTION ANSWERING: DATA GENERATION AND PERFORMANCE EVALUATION

by

Alican Acar

B.S., Mechanical Engineering, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering
Boğaziçi University

2024

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to those who have made this thesis possible and those who have provided their invaluable support throughout my academic journey.

First and foremost, I extend my sincere thanks to my supervisor, Murat Saraçlar, and co-supervisor Ebru Arısoy Saraçlar, whose expertise, understanding, and patience, added considerably to my graduate experience. Your guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master's study.

I am deeply grateful to all the faculty members in the Sytems and Control Engineering Department for their advice and assistance. Their knowledge and expertise in their fields have been invaluable to my academic and personal development.

I am heartily thankful to my family, whose love and guidance are with me in whatever I pursue. They are my ultimate source of inspiration and motivation. A special word of thanks to my mother, Hülya Acar, and my father, Ali Acar, for their understanding and endless love, through the duration of my studies.

I must acknowledge my fellow graduate student, Muhammet Hatipoğlu, who has helped me in the development of the Sound Collection Web App and also for recording a part of XQuAD which is a very important component of this study. For the collection of the sound recordings of XQuAD, I want to thank Alara Dirik, Çağrı Ertürk, Ece Bulaner, Emre Alpay, Hülya Acar, Murteza Mert Uyanık, Müge Bakır, Naz Karadağ, Nazlıcan Topçuoğulları and Yasin Yiğit Günaydın for their great patience.

Lastly, I offer my regards and blessings to all those who supported me in any respect during the completion of the project.

This work was supported in part by the Turkish Directorate of Strategy and Budget through the ROYAL Project under Grant 2019K12-149250.

This thesis not only stands as a testament to my academic endeavors but also embodies the collective efforts and encouragement of the many individuals who have been a part of this journey. Thank you.

ABSTRACT

TURKISH SPOKEN QUESTION ANSWERING: DATA GENERATION AND PERFORMANCE EVALUATION

This research is a comprehensive study on the development and evaluation of spoken question answering (SQA) in Turkish. The study systematically examines the challenges and methods involved in adapting question answering (QA) systems to process spoken data in Turkish. It begins with the generation of spoken datasets derived from the English to Turkish machine-translated version of the Stanford Question Answering Dataset (SQuAD) 2.0, called SQuAD-TR, and the Turkish split of Cross-lingual Question Answering Dataset (XQuAD), suitable for both natural and synthetic speech contexts. The basis for the creation of spoken datasets lies the integration of detailed text normalization and text-to-speech (TTS) systems. It continues with the iterative process of fine-tuning and evaluating various state-of-the-art automatic speech recognition (ASR) models in conjunction with the development of a robust QA model capable of accurately interpreting and processing Turkish spoken QA data. The research culminates in a comprehensive evaluation of the overall performance of the SQA system on various spoken and text-based QA datasets. This work not only improves the understanding of spoken QA systems in a multilingual context, but also provides important insights and advances specific to Turkish language processing, thus contributing to the broader field of multilingual natural language processing.

ÖZET

TÜRKÇE SESLİ SORU CEVAPLAMA: VERİ ÜRETİMİ VE PERFORMANS DEĞERLENDİRMESİ

Türkçe veri üzerinde eğitilmiş sözlü soru cevaplama makine öğrenmesi modelleri üzerine yapılan çalışmalar literatürde çok fazla yer kaplamamaktadır. Bu araştırma, Türkçe sözlü soru cevaplama sistemlerinin geliştirilmesi ve değerlendirilmesi üzerine yapılmış kapsamlı bir çalışmadır. Araştırmanın ana hedefi, otomatik soru cevap sistemlerinin sesli Türkçe verisini kavraması adına yapılan uyarlamaların zorluklarını ve yöntemlerini sistematik olarak incelemektir. Türkçe sesli soru cevap açık kaynak veri setlerinin eksikliğinden ötürü, bu çalışmada ilk olarak SQuAD 2.0'ın makine çevirisi versiyonu olan SQuAD-TR ve XQuAD'ın Türkçe bölümünden türetilen, hem gerçek konuşmacılar hem de yazıdan konuşma üretebilen yapay zeka modelleri kullanılarak veri setleri oluşturulmuştur. Bu veri seti oluşturma süreci de kapsamlı bir Türkçe metin standardizasyonu ve metinden konuşma üreten modellerin entegrasyonunu da içermektedir. Araştırma, en güncel konuşma tanıma modellerinin ince ayarının yapılıp değerlendirilmesi süreci ile devam eder. Bu süreç, Türkçe konuşulan soru cevaplama verilerini doğru bir şekilde yorumlayıp işleyebilen sağlam bir otomatik soru cevaplama modelinin geliştirilmesi ile birleştirilmiştir. Araştırma, çeşitli konuşma ve metin tabanlı soru cevaplama veri setlerinde konuşulan soru cevaplama sisteminin genel performansının kapsamlı bir değerlendirilmesi ile sonuçlanır. Bu çalışma, çok dilli bir bağlamda sesli soru cevaplama sistemlerinin anlaşılmasını geliştirmekle kalmaz, aynı zamanda Türkçe dil işlemeyle ilgili önemli içgörüler ve ilerlemeler sunar, böylece çok dilli doğal dil işleme alanına katkıda bulunur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . v
ÖZET . . . vi
LIST OF FIGURES . . . ix
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. RELATED WORK . . . 2
2.1. Text Based Question Answering . . . 2
2.2. Spoken Question Answering . . . 6
2.3. Existing Question Answering Datasets . . . 7
2.3.1. Text Based Datasets . . . 8
2.3.1.1. English Question Answering Datasets . . . 8
2.3.1.2. Turkish Question Answering Datasets . . . 9
2.3.1.3. Multilingual Question Answering Datasets . . . 10
2.3.2. Spoken Datasets . . . 12
3. DATASETS USED IN THIS STUDY . . . 15
3.1. SQuAD-TR . . . 15
3.1.1. Exploratory Data Analysis on SQuAD-TR . . . 16
3.2. Spoken SQuAD-TR . . . 16
3.2.1. Preprocessing of SQuAD-TR for Text-to-Speech . . . 17
3.2.2. Question Answering Dataset Partitions of SQuAD-TR . . . 20
3.2.3. Synthesis of Spoken Datasets . . . 22
3.3. Spoken XQuAD-TR . . . 25
4. QUESTION ANSWERING SYSTEM . . . 27
4.1. Automatic Speech Recognition . . . 27
4.1.1. XLS-R . . . 27

4.1.2. Whisper 28
4.1.3. WhisperX 29
4.2. Question Answering 30
4.3. The Overall System 30
5. EXPERIMENTS AND RESULTS 33
5.1. Experimental Setup 33
5.1.1. ASR Models 33
5.1.2. QA Model 34
5.2. Results and Discussion 35
5.2.1. Results of the ASR Models 35
5.2.2. Results of the QA Model 39
5.2.3. Results of the Spoken QA System 41
6. CONCLUSION 52
REFERENCES 53

LIST OF FIGURES

Figure 3.1. Text normalization pipeline. 18

Figure 3.2. QA examples regarding the overlapping score. 21

Figure 3.3. Sound collection app workflow. 26

Figure 4.1. A cascade SQA system. 27

Figure 4.2. ASR pipeline. 31

Figure 4.3. Text-based QA pipeline. 31

Figure 4.4. SQA pipeline. 32

Figure 5.1. Validation WERs of XLS-R fine-tuning. 36

Figure 5.2. BERT results on QA-Dev. 39

Figure 5.3. BERT results on QA-Test. 40

LIST OF TABLES

Table 3.1. Exploratory data analysis of SQuAD-TR. 16

Table 3.2. Normalized abbreviations. 19

Table 3.3. Exploratory data analysis of SQuAD-TR QA partitions. 21

Table 3.4. Naming convention of ASR training datasets. 23

Table 3.5. Naming convention of spoken QA training datasets. 23

Table 3.6. Naming convention of XQuAD datasets. 24

Table 5.1. Hyperparameter configuration of XLS-R model. 33

Table 5.2. Hyperparameter configuration of QA models. 35

Table 5.3. XLS-R test results. 37

Table 5.4. Test results of the Whisper models. 37

Table 5.5. Test results of the WhisperX models. 38

Table 5.6. Test resuls of QA models on XQuAD 41

Table 5.7. QA pairs before and after decoding across datasets 42

Table 5.8. WER metrics before elimination 43

Table 5.9. WER metrics after elimination 43

Table 5.10. WER metrics at intersection 44

Table 5.11. EM performance of BERT models on QA-Test datasets. 45

Table 5.12. F1 scores of BERT models on QA-Test datasets. 45

Table 5.13. EM performance of BERT models on XQuAD datasets. 46

Table 5.14. F1 scores of BERT models on XQuAD datasets. 47

Table 5.15. EM performance of BERT models on XQuAD-1hr datasets. 47

Table 5.16. F1 scores of BERT models on XQuAD-1hr datasets. 48

Table 5.17. EM perf. of BERT multilingual models on QA-Test datasets. . . . 49

Table 5.18. F1 scores of BERT multilingual models on QA-Test datasets. . . . 49

Table 5.19. EM perf. of BERT multilingual models on XQuAD datasets. . . . 50

Table 5.20. F1 scores of BERT multilingual models on XQuAD datasets. . . . 50

Table 5.21. EM perf. of BERT multilingual models on XQuAD-1hr datasets. . 51

Table 5.22. F1 scores of BERT multilingual models on XQuAD-1hr datasets. . 51

LIST OF SYMBOLS

S_1	The first text sequence in the overlap comparison.
S_2	The second text sequence in the overlap comparison.
$S_1[i]$	The i-th element of the first sequence.
$S_2[i]$	The i-th element of the second sequence.
δ	An indicator function that returns 1 if $S_1[i] = S_2[i]$ (the characters match) and 0 otherwise.
ϵ	Hyperparameter of ADAM optimizer to prevent division by zero
Σ	Summation symbol, used to denote the addition of a sequence of numbers.

LIST OF ACRONYMS/ABBREVIATIONS

ADAM	Adaptive Moment Estimation
API	Application Programming Interface
ARCD	Arabic Reading Comprehension Dataset
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
CMRC	Chinese Machine Reading Comprehension
CMU	Carnegie Mellon University
CNN	Convolutional Neural Network
DRCD	Delta Reading Comprehension Datase
EM	Exact Match
FQuAD	French Question Answering Dataset
GermanQuAD	German Question Answering Dataset
GPU	Graphical Processing Unit
IDF	Inverse Document Frequency
IR	Information Retrieval
KorQuAD	Korean Question Answering Dataset
LSTM	Long Short-term Memory
MC	Machine Comprehension
MCTest	Machine Comprehension Test
NLP	Natural Language Processing
NMSQA	Natural Multi-speaker Spoken Question Answering
ODSQA	Open Domain Spoken Question Answering
ParSQuAD	Persian Stanford Question Answering Dataset
QG	Question Generation
SberQuAD	Sber Question Answering Dataset
Spoken-CoQA	Spoken Conversational Question Answering
SQA	Spoken Question Answering
SQuAD	Stanford Question Answering Dataset

SQuAD-TR	Turkish Stanford Question Answering Dataset
TF	Term Frequency
THQuAD	Turkish Historic Question Answering Dataset
TOEFL	Test of English as a Foreign Language
TQuAD	Turkish Question Answering Dataset
TTS	Text to Speech
VAD	Voice Activity Detection
Wav2Vec2	Second Version of Wav-to-Vector
WER	Word Error Rate
XLS-R	Cross-lingual Self-Supervised Speech Representations for Improved Dysarthric Speech Recognition
XQuAD	Cross-lingual Question Answering Dataset

1. INTRODUCTION

The rapid development of SQA systems has shown that extending this technology to low-resource languages such as Turkish poses a particular challenge. This study specifically aims to address the lack of research and resources in the development of effective SQA systems for Turkish. Turkish, characterized by its complex linguistic structure and increasing digital influence, offers a great opportunity for technological innovation in language processing.

At the core of this research is the creation of specialized spoken datasets that are crucial for training and evaluating SQA systems, as well as the creation of benchmarks for these datasets. These spoken QA datasets, taken from the English to Turkish machine-translated version of SQuAD 2.0, SQuAD-TR, and the Turkish split of XQuAD, are designed to include both natural and synthetic speech variations. The research methodology follows a structured approach involving an iterative process of refining state-of-the-art ASR models, supported by the development of a Turkish text normalization pipeline for fair ASR evaluations in conjunction with the development of tailored spoken QA models that can accurately process and respond to Turkish spoken queries.

This work provides progress for the gaps in the research of SQA for Turkish, a language with limited resources. The development of open-source datasets for spoken QA based on SQuAD-TR and the Turkish split of XQuAD is a step towards overcoming the current limitations due to the lack of such resources. Furthermore, the inclusion of real speakers in the process of curating datasets aims to improve the realism and effectiveness of SQA systems for Turkish. Through these contributions, this study hopes to advance the field of multilingual NLP.

2. RELATED WORK

Automatic QA is a field that mainly overlaps with natural language processing (NLP) and information retrieval (IR) and aims to develop systems that are able to provide precise answers to questions posed in natural language. In recent decades, this field has evolved considerably thanks to new model architectures, increasing computing power and the availability of large datasets. This section provides an overview of the most important developments and methods in text-based and spoken QA as well as existing datasets in these areas.

2.1. Text Based Question Answering

The first QA systems relied heavily on hard-coded rules and pattern-matching techniques to parse questions and retrieve answers from structured databases. While these systems, such as BASEBALL [1] showed the potential of automated QA, they were limited by their reliance on structured data and struggled with the complexity and variability of natural language.

The development of automatic QA systems underwent a decisive change with the integration of statistical methods and IR techniques, especially to meet the challenge of sifting through the ever-increasing amounts of unstructured text on the Internet. This has formed the fields of open and closed domain QA. In closed domain QA, source documents come from a constrained topic whereas in open domain QA the topics can vary. From the late 1990s to the early 21st century, data-driven approaches became essential to improve the accuracy and relevance of answers through advanced search algorithms and models. The open domain QA systems essentially consists of two sequential parts, first is the retriever module and the second is the reader. When a query is received, the retriever has the task of finding possible locations where the answer could be found. The reader then takes over the task of filtering out the exact location of text containing the answer from these identified passages. Initially, statistical QA

relied on keyword matching to find answers within large datasets, utilizing sophisticated ranking algorithms to elevate the most pertinent passages. BM25 [2] is one of the earliest examples of ranking functions which is the core of the retriever module. It is specifically designed to rank a set of documents based on the query terms appearing in each document, taking into account the term frequency (TF) within the document and the inverse document frequency (IDF) of the query terms across the entire document collection. After BM25, the application of bootstrapping surface patterns [3] , uses predefined or dynamically generated patterns to identify possible answer candidates in the text and thus improve the efficiency of the retriever module in finding relevant information.

There is another sub-field of automatic QA which is reading comprehension. Unlike OpenQA, reading comprehension systems work within the confines of specific, provided texts to understand and extract information. The earliest example is the Deep Read [4], using a corpus of 120 stories each accompanied by short answer questions and an answer key, Deep Read utilizes a basic system based on bag of words for pattern matching and followed by sophisticated automated linguistic processing tools such as stemming, name identification, semantic class identification and pronoun resolution. With this approach, Deep Read is able to successfully find sentences containing the answer in 30-40% [4] of cases. Bag of words approach has been further improved with hard-coded rules in Quarc [5] which has successfully found answers 40% of the cases. Unlike this hard-coded rule approach, logistic regression is used in Aquareas [6] which is the first machine learning approach for pattern matching in reading comprehension. However, the datasets used in these studies are very small. MCTest [7] addresses this problem with its 2,640 questions and multiple choice answers. In MCTest, the first baseline system was designed to apply a sliding window technique that uses a bag of words model. This model summarizes elements from the question and the proposed answer and attempts to find matches in the text. However, this method does not take into account long texts. Thus, a second algorithm was developed which integrates a word distance-based algorithm that evaluates the proximity between the words in the question and those in the text, thus improving the recognition of long range depenedent

relevant information. Combining these two approaches, the final system successfully locates 57.75% [7] of the answers in the test set.

The SQuAD dataset [8] represents a notable departure from the existing machine reading comprehension datasets in that it avoids offering a fixed list of possible answers for each question. Instead, the task becomes finding the correct answer from a given passage, leading to a more challenging reading comprehension task. Recognizing the diversity of question and answer types in SQuAD, methods that rely on dependency tree distances were investigated to sift through this diversity and stratify questions by their level of difficulty. The use of span-based answers allows for a more straightforward evaluation process than that of free-form answers. To assess the difficulty of SQuAD, a logistic regression model was used that incorporates several features, in particular lexical items and dependency tree paths, and shows their importance for the accuracy of the model. It was found that the performance of the model decreases with increasing complexity of the answers and the syntactic difference between the question and the corresponding text. The best performing model achieved an F1 score of 51.0% [8], significantly outperforming the sliding window method baseline of 20% [8]. Despite the progress made by sophisticated neural network based models [9], such as LSTM, which achieved an F1 score of 70.3% [9] on SQuAD v1.1, these figures still fall short of the human benchmark of 86.8% [9] F1 score based on cross-checked annotators. This discrepancy highlights the scope for improvement in the models and learning approaches applied to the SQuAD dataset and shows that there is still much scope for further research in this area which has been followed by transformer-based neural network models such as Bidirectional Encoder Representations from Transformers (BERT) [10]. The SQuAD 1.1 ranking shows that transformer-based architecture models such as ANNA [11], LUKE [12] and SpanBERT [13] have excellent performance.

The introduction of BERT has had a profound impact on machine reading comprehension, as evidenced in particular by its performance on the SQuAD dataset. The architecture of BERT, which pre-trains deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers, represents

a significant advance in the ability of machines to understand the nuances of human language. This approach stands in stark contrast to previous models, which generally processed text unidirectionally. BERT set new standards in the evaluation of the SQuAD dataset and clearly outperformed the existing models. Its ability to comprehend and answer complex questions from the dataset demonstrated the effectiveness of its bidirectional training and the potential of transformer-based models for natural language processing tasks. For example, BERT achieved a remarkable improvement on SQuAD v1.1 over previous models, achieving 93.2% F1 score on the test set while human performance is 91.2% for F1 score [10]. The impact of BERT goes beyond achieving high scores on benchmarks; it has fundamentally changed the approach to developing machine reading comprehension systems. By utilising transfer learning, where a model trained on a large corpus of text can be fine-tuned for specific tasks such as QA, BERT has enabled more efficient and effective model development. This has led to widespread use in a variety of applications, from search engines to virtual assistants, where understanding the context of user queries is critical.

As already mentioned, the transformer-based models perform excellently in SQuAD 1.1. The ANNA introduces a neighbourhood-aware mechanism in its extended pretraining task and achieves top score on the SQuAD 1.1 benchmarks which is F1 score of 95.7% [11] by focusing on richer context capture. LUKE improves entity representation by integrating a unique entity-aware self-attention mechanism and achieves 95.4% [12]. SpanBERT innovates by masking contiguous spans and focuses on span boundary training, it significantly improves its performance on span-based tasks which is a perfect fit for the QA task. SpanBERT achieved an F1 score of 94.6% [13]. SQuAD 2.0 [14], expanded from SQuAD 1.1, contains an additional 50,000 questions that cannot be answered using the context texts provided. While ANNA has not been adapted for SQuAD 2.0, both LUKE and SpanBERT have been adapted, although their performance has decreased in this version but is still good. More specifically, LUKE and SpanBERT have achieved F1 scores of 90.2% and 88.7% respectively [15]. Variants of BERT such as ALBERT [16] and RoBERTa [17] showed great performance on SQuAD 2.0 benchmarks. ALBERT introduces parameter-reduction techniques to en-

hance scalability and training speed, establishing new benchmarks in model efficiency while maintaining competitiveness on tasks requiring multi-sentence understanding. It achieved F1 score of 92.2% on SQuAD 2.0 [16]. RoBERTa, through its comprehensive replication and hyperparameter optimization study, highlights the critical impact of training regimen and dataset size, achieving F1 score of 89.8% [17].

2.2. Spoken Question Answering

SQA systems generally consist of an ASR model, which translates the spoken language into written form, and a QA model, which locates the answer of a question given either in text or spoken form. In the study [18], the CMU Sphinx [19] is used as an ASR module to obtain the transcriptions of the QA pairs of the TOEFL dataset. After obtaining glove vector embeddings for the input texts, a gated recurrent unit network is trained for the QA task. In manual transcriptions, the best performing model achieved an accuracy of 51.56% [18] which is the percentage of questions answered correctly. Even in ASR transcriptions with a word error rate (WER) of 34.32% [18], the best-performing model continued to show its effectiveness, achieving an accuracy of 48.33% [18]. Due to the size of the dataset and the fact that the answers were multiple-choice questions, Spoken SQuAD [20] was developed.

Spoken SQuAD contains synthesized speech obtained using Google's TTS system. It is important to note that whenever the answer was not found correctly in the ASR transcriptions of the corresponding item, the question-answer pair was removed from the dataset, as these examples were too difficult for the current listening comprehension engines. In this way, 37,111 question-answer pairs were compiled for the training set and 5,351 for the test set. Spoken SQuAD was decoded using CMU Sphinx. The WERs for the training and test data sets were 22.77% and 22.73% respectively. To overcome the effects of ASR errors on the QA task, subword units were used. Although ASR makes errors during inference, it is still able to infer subwords accurately. Therefore, the vector embeddings were enriched with the subword embeddings. Phoneme sequences of words are used in this context. A convolutional neural network (CNN), Phoneme-

CNN, is used to generate embeddings from the phoneme sequence of a word. Several QA models were tried for the reading comprehension and the average results of the models result in exact match (EM) 41.96% and F1 score of 55.40% [20].

The mitigation of ASR errors in the area of SQA is an important task, and various approaches have already been pursued. In [21], mitigating the errors of ASR is done by introducing contextualized word representations, which allow a single word to be represented by its neighboring words or the context in which it is located. New state-of-the-art results were achieved with contextualized word representations on both the artificially synthesized and real SQA benchmark datasets, marking a 63.4% EM and 74.4% F1 score on Spoken SQuAD dataset [21]. Another approach to ease the effects of ASR errors is the use of confusion networks. In the study [22], enriching the vector representations of words is done by introducing confusion networks with word confidence score which results in uncertainty-aware vector embeddings. Another approach to use knowledge distillation from spoken data with its transcriptions to prevent the ASR errors. In the paper [23], the models have the same architecture in contrast to the usual knowledge distillation, where the teacher model is a larger model than the student model. It is attempted to use the distilled knowledge of the reference texts, which is the output of the teacher model, for the student model, which is responsible for processing the ASR transcriptions of the reference texts.

2.3. Existing Question Answering Datasets

In this chapter, we dive into the diverse world of QA datasets, which form the backbone of research and development in the field of NLP. These datasets are crucial for the training and evaluation of algorithms that aim to understand and answer human language in a meaningful way. The scope of these datasets varies widely, encompassing both text-based and spoken formats and covering a range of languages and approaches.

2.3.1. Text Based Datasets

Text-based QA datasets are central to the development of models that process and interpret written language. They present NLP systems with unique challenges and opportunities as they often require a deep understanding of context, semantics and syntax.

2.3.1.1. English Question Answering Datasets. This section focuses on datasets specifically designed for English language QA. These datasets represent a wide range of genres and complexity and test different aspects of language comprehension and processing in English.

In the field of English QA datasets, several notable collections have significantly advanced research in natural language processing and machine comprehension. Among them, MCTest [7] stands out as one of the early contributions. MCTest is a dataset developed specifically for research into machine comprehension. It requires machines to answer multiple-choice questions on the reading comprehension of fictional stories. There are 2640 human-generated question answer pairs regarding 660 short stories. The size of the dataset is relatively small compared to current datasets.

This is where SQuAD 1.1 [8] comes into play, which consists of more than 100,000 question-answer pairs derived by humans from Wikipedia articles. In this dataset, the answers are not multiple-choice, but span of words from the corresponding article. As already mentioned, SQuAD 2.0 [14] extends SQuAD 1.1 with more than 50,000 unanswerable questions, which in turn were derived by humans. This enables the training of models that can determine whether a question in a particular article can be answered or not. The main dataset in our work, SQuAD-TR [24] is the English to Turkish machine translated version of SQuAD 2.0.

There is another paradigm in the field of QA called Multi Hop QA. The datasets mentioned so far consist of question-answer pairs for a single article. In multi hop

QA, the answer to a question can contain information from multiple paragraphs, which enables the training of models that can draw very complex conclusions. The HotpotQA dataset [25] is one of the most comprehensive datasets in this area. It consists of more than 113,000 question-answer pairs taken from Wikipedia articles. It is important to note that the questions in this dataset do not require any prior external knowledge in order to be answered. TriviaQA [26] adds external knowledge sources to the question-answer pairs in addition to the articles. It contains more than 95,000 question-answer pairs, about 6 pieces of evidence per pair, resulting in a total of 650,000 question-answer-evidence triplets created by trivia enthusiasts.

2.3.1.2. Turkish Question Answering Datasets. Turkish QA datasets provide insights into the particular challenges of the Turkish language, which has unique linguistic features that differ from Indo-European languages such as English. These datasets help in the development of systems that can take into account the nuances of Turkish syntax and semantics. However, there are not many Turkish QA datasets that are publicly available. TQuADv1 [27] is one of the first examples created at the Teknofest 2018 Artificial Intelligence Competition. This dataset contains 9,200 QA pairs on 2,507 paragraphs from the field of Turkish-Islamic history of science. THQuAD [28] is an extended version of TQuADv1. To provide context, this dataset contains an additional 6,234 QA pairs on 194 paragraphs from the field of Ottoman history. LectureQA [22,29] is a dataset that is available in both Turkish and English and consists of QA pairs in which the questions are in written form and the answers correspond to the sections of the transcripts of the lecture videos. Google Deepmind's XQuAD [30] also contains a Turkish split with 1,190 QA pairs from 240 paragraphs. This split was obtained by manually translating a subset of SQuAD 1.1 into Turkish.

In contrast to the other approaches, TurQuAse [31] offers a different method than the creation of a dataset from scratch by humans. It is a framework for the automatic creation of spoken QA datasets that includes 3 main modules: Question Generation (QG), TTS and ASR. For the QG task, a pre-trained mT5 model is trained on the combination of the THQuAD and TR-SQuAD datasets. Aprroximately 83,600

QA pairs from the well processed about 20,400 Wikipedia paragraphs were generated, which are the first paragraphs of the pages and do not contain non-Turkish characters, to improve the quality of the dataset. The paragraphs were then converted into spoken form using the TTS module. More details are given in Section 2.3.2.

2.3.1.3. Multilingual Question Answering Datasets. Multilingual datasets present a more complex challenge as they span multiple languages. They are crucial for the development of systems that can work across language boundaries. There are 2 main approaches for creating the multilingual datasets which are creating datasets from scratch and translating existing dataset from one language to another.

Most of the datasets from scratch follow the same approach and structure as one of the pioneers: SQuAD 1.1. KorQuAd [32] consists of 70,079 QA pairs from Korean Wikipedia articles. FQuAD 1.0 [33] is a dataset that focuses on French reading comprehension and comprises over 25,000 questions. These questions were developed by higher education students based on a collection of Wikipedia articles. SberQuAd [34] contains 50364 paragraph-question-answer triplets extracted from Russian Wikipedia articles which is constructed with the same approach as SQuAD; Wikipedia pages split into paragraphs and those paragraphs are presented to crowdworkers to derive QA pairs. GermanQuAD [35] which consists of 13,722 QA pairs from German Wikipedia articles, counterpart of the articles from SQuAD 1.1, followed a more sophisticated approach where the QA annotators are selected from different groups instead of sole crowdworkers. Annotators familiar with supervised learning were selected and offered a BERT workshop on QA training and labeling. Students of computational linguistics and computer science as well as an experienced annotator contributed to the expansion of the dataset. Crowdworkers were also used to ensure quality through careful monitoring and feedback. The CMRC 2018 [36] is enriched with 19,071 QA pairs carefully annotated by human experts on Chinese Wikipedia paragraphs. Its design not only emphasizes a wide range of informative topics, but also ensures a high level of accuracy and reliability of the content. In addition, CMRC 2018 includes a special task set with questions that require in-depth understanding and the ability to synthesize informa-

tion across multiple sentences, making an important contribution to the diversity and advancement of Chinese language comprehension tools. The Arabic Reading Comprehension Dataset (ARCD) [37] takes a hybrid approach, creating a dataset from scratch and using the machine translation of SQuAD 1.1 from English into Arabic. The ARCD includes 1,395 questions formulated by crowdworkers based on Arabic Wikipedia articles. In addition, it contains an Arabic machine-translated version of the SQuAD (Arabic-SQuAD), which contains 48,344 questions.

Creating translated datasets are done by using two primary methods: human translation and machine translation. Human translation, while accurate, requires significantly more resources compared to its machine counterpart, resulting in many machine-translated datasets being available for public use. A common practice is to translate versions of the SQuAD, specifically SQuAD 1.1 or SQuAD 2.0, using different machine translation models. As mentioned earlier, a large part of the ARCD consists of the machine-translated version of SQuAD 1.1. The French version of SQuAD, referred to as French-SQuAD [11], is another machine-translated adaptation of SQuAD 1.1, which is run via the Google Translate API and comprises approximately 107,700 question-answer (QA) pairs. The Spanish SQuAD [38] translates SQuAD 1.1 into Spanish and comprises over 100,000 QA pairs. Another variant, ParSQuAD [39], is the Persian translation of SQuAD 2.0 and also uses the Google Translate API. Due to the inherent inaccuracies in machine translation, it is decided to publish two different versions of each dataset. These versions differ in how they identify the correct answer spans. The first version, in which the correct answer spans are identified manually, is smaller in size and contains a combined total of 24,632 QA pairs across the training and development sets. The second version uses a string matching algorithm to identify the correct answer spans and is larger with 70,560 QA pairs in both sets.

This study mainly uses SQuAD-TR, a machine-translated version of SQuAD 2.0. This dataset, which forms the core of our research, is described in more detail in Section 3.1.

As far as human-translated datasets are concerned, there are not many publicly available datasets. XQuAD contains 240 paragraphs and 1,190 QA pairs taken from the SQuAD 1.1 development set. These paragraphs and QA pairs were translated into ten languages: Spanish, German, Greek, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese and Hindi. The translation of both the paragraphs and the QA pairs was carried out by professional human translators. To simplify the process of annotating answer spans, the most frequently occurring answer of each question is selected. The beginning and the end of these answers were marked in the context sections with special symbols. The translators were also instructed to leave these symbols in their respective places in the translated versions in order to ensure the consistency of the answers in the different languages. The Turkish partition of XQuAD is used as one of the test sets in this study.

2.3.2. Spoken Datasets

This section focuses on datasets specifically designed for spoken language understanding and highlights their unique features and applications. We examine several prominent spoken language datasets, each offering different insights into the field of NLP and ASR.

As with translated QA datasets, there are 2 main approaches to creating a spoken dataset: text data is read by humans or audio data is synthesized through the use of a TTS system. For the Open Domain Spoken Question Answering (ODSQA) Dataset [40], 20 native Chinese speakers are recruited to create the dataset. The reference texts come from the Delta Reading Comprehension Dataset (DRCD) [41], which contains 30,421 QA pairs in the training and development set combined. A reader is recruited to record all the sentences of a document, and each sentence was recorded at 16 kHz. In the end, 3,654 spoken QA pairs were collected.

Since the creation of a spoken dataset requires intense human resources, advances in TTS have made the synthesis of spoken datasets possible. The Spoken SQuAD is

one of the first examples of this approach. To develop a spoken variant of the SQuAD dataset, the several number of techniques were used. First, Google's TTS technology was used to convert the text articles from SQuAD into spoken form. The CMU Sphinx was used to create transcriptions of these audio documents. The source of the Spoken SQuAD training dataset was the original SQuAD training dataset, and similarly the SQuAD development dataset was adapted to form the test set for Spoken SQuAD. The QA pairs from the dataset where the answer did not appear in the ASR transcriptions of the corresponding article were removed from the spoken dataset, as these cases were considered too complex for the existing capabilities of the listening comprehension technologies. Using this method, 37,111 QA pairs were generated for the training dataset and 5,351 pairs for the test dataset.

Another dataset, Natural Multi-speaker Spoken Question Answering (NMSQA) Dataset [42], was created with a similar approach using the Amazon Polly TTS service that synthesizes SQuAD 1.1. In developing the dataset, the 12 TTS voices are selected to ensure that a wide range of speakers were available for each spoken document and QA pairs. This approach resulted in the compilation of a comprehensive dataset with 297 hours of audio recordings for training purposes and 37 hours for development. To accurately identify the segments containing the responses in these audio files, the Montreal Force Aligner is used for precise annotation.

The Spoken-CoQA [43] dataset, whose approach is similar to that of Spoken SQuAD, differs from the latter by including a test set read by humans. This dataset was derived from the CoQA dataset for spoken questions and answers, which includes about 8,000 stories and over 120,000 question-answer pairs. The CoQA content was converted into an audible format using Google's TTS service. The spoken material was then transcribed into scripts using CMU Sphinx. This generated approximately 300 hours of audio data, with the ASR transcriptions having a WER of 15.9%. For the test set of the dataset, five native English speakers were assigned to read the documents and QA pairs. The sentences of each document were read by the same speaker throughout to ensure consistency, even though the questions within a sample could be presented

by different speakers.

As previously mentioned, LectureQA [22,29] comprises a dataset with text-based questions and answers derived from lecture video transcripts. Similarly, TurQuAse is an automatic framework to create Turkish spoken dataset. It contains 83,600 QA pairs from 20,400 Wikipedia paragraphs. Google's TTS service is used and before applying TTS, several normalization techniques have been applied for a fair evaluation of the performance of ASR. A pre-trained multilingual XLS-R is used as the ASR module. Using TTS, 223 hours of speech data is obtained.

3. DATASETS USED IN THIS STUDY

3.1. SQuAD-TR

As mentioned in Section 2.3.1.3, it is common to apply machine translation to a large English dataset. SQuAD-TR [24] was created in this way. SQuAD 2.0 was translated into Turkish using Amazon Translate. All of the titles, context paragraphs, questions and answers in the original dataset were translated. Since the answer spans were not retained in the translated paragraphs, it was necessary to reassign the starting positions of the answer spans. This necessity arose not only from the linguistic differences between the source and target languages, but also from the context-dependent nature of the translation task.

The translation of a span of text can vary considerably depending on the context. This poses a challenge when it comes to obtaining consistent translations, especially for Turkish, due to the context-dependent morphological variations of Turkish words. The problem with finding correct correspondence of answers is that it requires a great amount of time and manual work. Thus, in SQuAD-TR common hard-coded and rule-based post processing methods were used to find the right answer spans. An attempt was first made to find the exact matches. If the translated answer occurs completely in the translated context paragraph, the start of the answer was changed accordingly, which is crucial information for training a model for the QA task. However, it was not mentioned how many answer spans were found using this approach. After the answers were found by EM, an attempt was made to find them by approximate match with the character edit distance. The strategy was to use different edit values for different answer lengths. If the answer length was below 4 characters, an edit distance of 1 was used. If the answer length was more than 4 characters, then a search was made for an answer within a range of maximum 3 edit distances to select the longest possible span of texts.

3.1.1. Exploratory Data Analysis on SQuAD-TR

SQuAD consists of titles, context paragraphs and QA pairs. Each title contains context paragraphs and each paragraph contains relevant QA pairs. A detailed data analysis on these units can be found in Table 3.1. The version of SQuAD-TR mentioned in the paper is the first version, but we used the third version. The differences in terms of the number of context paragraphs and QA pairs are mainly due to the approach used to determine the correct answer span after translation. In the first version, the correct answer spans could not be found for 25,528 QA pairs, which also led to the removal of 259 paragraphs that did not contain any QA pairs. However, in the third version of SQuAD-TR, exactly the same number of paragraphs and QA pairs were included as in the original SQuAD 2.0. This improvement is not perfect, the details of this imperfection are discussed in Section 3.2.2.

Table 3.1. Exploratory data analysis of SQuAD-TR.

Feature	**SQuAD 2.0**	**SQuAD-TR**
Titles	442	442
Paragraphs	19,035	19,035
QA Pairs	130,319	130,319
Answerable Questions	86,821	86,821
Unanswerable Questions	43,498	43,498

3.2. Spoken SQuAD-TR

As mentioned in Section 2.3.2, creating a spoken dataset with real speech is a very laborious process. Therefore, we have followed a common approach to create the Spoken SQuAD-TR using machine synthesis. Due to advances in large deep models, there are now many well-performing TTS models, even for low-resource languages. Performance of spoken SQuAD-TR was evaluated using a cascade QA sytem where the dataset is first decoded with ASR and then QA is performed on ASR transcriptions.

For a fair evaluation of an ASR model, the reference texts must undergo several preprocessing steps for normalization or standardization.

3.2.1. Preprocessing of SQuAD-TR for Text-to-Speech

ASR systems were developed to convert spoken language into a written format. However, spoken language often contains colloquialisms, non-standardized word forms and verbal nuances that cannot be directly translated into written text. Text normalization standardizes these deviations by converting numbers into words, expanding contractions and correcting grammatical inconsistencies. This process creates a standardized format that better adapts to the output of ASR systems and allows for a more accurate assessment of their performance. Without normalization, reference texts could retain irregularities and colloquial speech patterns, unfairly penalizing the ASR system for discrepancies that are not errors but stylistic or contextual differences. In this study, we have applied several text normalization steps to the reference texts before input them into the TTS model and the decodings of the ASR model.

The text normalization pipeline is demonstrated in Figure 3.1. Since the actual data is the machine translation of SQuAD 2.0 which contains thousands of Wikipedia articles, it has lots of characters outside Latin alphabet, therefore these articles have been removed by using regular expressions. Second step is to normalize percent sign which is done hard-coded in Python to transform it to its Turkish correspondance (which is "yüzde"). Third step is standardizing abbreviations, in SQuAD-TR there have been lots of abbreviations and some of them may correspond to different things. For instance, 'c.e.' may correspond to "Common Era" and product certification in European Economic Area or "ms." may correspond to title for unmarried woman or "Anno Domini" in Turkish. Thus, the most frequent and apparent abbreviations have been normalized. Table 3.2 shows the list of abbreviations that are normalized at preprocessing. Also, it is important to note that both lowercase and uppercase versions of the abbreviations have been normalized. After that, Roman numerals and ordinal Roman numerals have been normalized by hard-coded rules to regular numbers which

are denoted by digits. After this step, dash normalization has been applied with a hard-coded rule of transforming dash between 2 numbers to “between” in Turkish (i.e. “18-22” is converted into “18 ila 22”) which is done right before the all number normalizations.

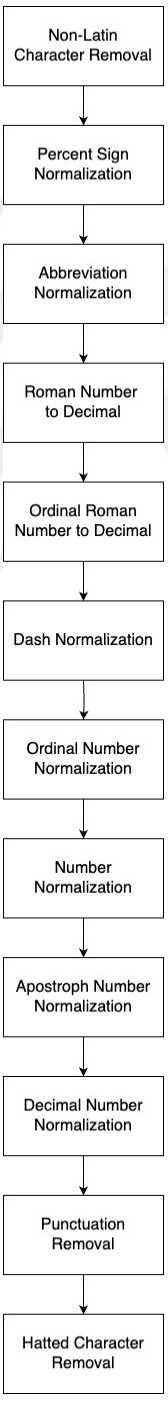

Figure 3.1. Text normalization pipeline.

Table 3.2. Normalized abbreviations.

Abbreviation	**Correspondance**
A.B.D.	Amerika Birleşik Devletleri
A.Ş.	Anonim Şirketi
Dr.	Doktor
Vd.	Ve diğerleri
Vs.	Vesaire
Örn.	Örneğin
Sr.	Senior
Jr.	Junior
Rev.	Muhterem
Bkz.	Bakınız
No.	Numara
Vb.	Ve benzeri
M.Ö. - MÖ. - MÖ	Milattan Önce
M.S.	Milattan Sonra
Mr.	Bay
Mrs.	Bayan

Number standardization covers for 4 cases which are ordinal numbers, numbers, numbers containing apostrophes and decimal numbers. Normalizing ordinal numbers contain both machine learning and hard-coded rules. First, the strings containing the ordinal number is detected with a part-of-speech tagger [44]. After that, the normalization requires to handle vowel harmony ("Büyük Ünlü Uyumu") which is a phonological phenomenon in Turkic languages by applying hard-coded rules. Regular number normalization is done by fixed rules which covers the cases of negative numbers or the inconsistency using comma or point while writing large numbers. Strings that contain numbers and apostrophes are very common in Turkish because the suffix after the apostrophe can be a locative case marker or possessive suffix (i.e. "1920'de" means "In 1920"). This normalization is done by fixed rules again. The final case regarding

numbers is decimal number normalization which covers both use cases of comma or point. It is important to note that, units require text normalization. In this study, we have tried Quantulum3 [45] which is a stochastic gradient descent classifier to detect the units and then translate them with an open source version of Google translate. The detection part was accurate but translation part was working poorly, so we didn't apply normalization to units. Finally, removal of punctuation and hatted characters were done with hard-coded rules.

3.2.2. Question Answering Dataset Partitions of SQuAD-TR

SQuAD-TR consists of context paragraphs and related questions and answers. Since it is a machine translated version of SQuAD 2.0, several methods are applied to find the correct answer spans. Despite those post-processing steps, all correct answer spans could not be found and if the answer span is not correct, the performance of the QA models trained on this data decreases drastically due to noise. Thus, we first checked how many exact matches of the answers ending with a whitespace with the given "answer start" data. The number of exact matches of answers that are followed by whitespace in the data is 30,958. Thus, we have decided to complete the deficient answers until the end of the alphanumeric characters in the possible answer span. We used a simple overlapping score in character level for validation as

$$\text{overlap}(S_1, S_2) = \frac{\sum_{i=1}^{|S_1|} \delta(S_1[i], S_2[i])}{|S_2|}. \tag{3.1}$$

For two strings, S_1 and S_2, where S_1 is either shorter or the same length as S_2, the function $\text{overlap}(S_1, S_2)$ calculates the proportion of overlapping characters. It is defined as $\text{overlap}(S_1, S_2)$, where $|S_1|$ and $|S_2|$ are the lengths of S_1 and S_2 respectively, $S_1[i]$ and $S_2[i]$ denote the ith character of each string, and $\delta(x, y)$ is an indicator function that returns 1 if $x = y$ (the characters match) and 0 otherwise. The numerator sums the instances of matching characters at each position i in S_1 and S_2, and the denominator is the length of S_2, which gives the proportion of matching characters in S_1 with respect to the total number of characters in S_2. Here S_1 is the original answer and S_2 is the

completed candidate answer. In this way, we can obtain answers that have problems with the case, or the answers with suffixes responsible for dative or locative information, as shown in Figure 3.2.

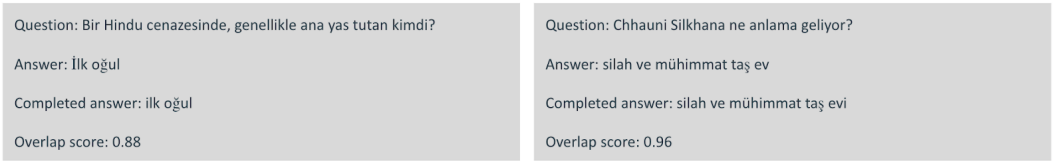

Figure 3.2. QA examples regarding the overlapping score.

We applied several threshold values by using this overlapping score to get different datasets which are subsets of SQuAD-TR. For the clarification of the threshold values, overlapping score of "0" means that there is no need to be any correspondance in character level between the original answer and the updated answer. In the case of overlapping score of "1", new answers mean the addition of exact matches ending without the whitespace. Table 3.3 shows the total number of paragraphs, additional QA pairs and resulting number of total QA pairs.

Table 3.3. Exploratory data analysis of SQuAD-TR QA partitions.

Threshold	**New Answers**	**Total Answers**
0	55,862	86,820
0.2	35,214	66,172
0.4	34,279	65,327
0.5	33,791	64,749
0.6	32,806	63,764
0.8	29,009	59,967
1.0	18,076	49,034

3.2.3. Synthesis of Spoken Datasets

In developing our spoken datasets, we used Google's TTS application programming interface (API) which uses WaveNet [46]. WaveNet, developed by DeepMind, differs significantly from traditional TTS systems that rely on the concatenation of short segments of speech. Instead, it generates speech by producing audio waveforms sample by sample, resulting in more natural-sounding speech. The basis of WaveNet is a CNN that can process sequential data, especially audio signals. By training with an extensive corpus of high-quality speech recordings, WaveNet learns to reproduce the subtleties of human speech, including tempo and emotional inflections. This approach enables the creation of a variety of voices that differ in accent, inflection and style without the need for extensive manual adjustments. For Turkish synthesis, there are 5 available speakers which are not very different in terms of the accent. There are 3 male and 2 female speakers. In this study, we used 2 male (speakers B and E) and 2 female speakers (speakers A and D) for the creation of the datasets. The speed parameter is 1.00 and pitch parameter is 0.0 that are the default values of the API. All of the reference texts of the synthesized spoken datasets has been normalized with the procedure, depicted in Figure 3.1. After the text normalization pipeline, all context paragraphs were subjected to sentence splitting in order to synthesize the texts at sentence level.

The first group of spoken datasets is used for fine-tuning of the ASR models. The main goal is to synthesize training datasets of 2, 4 and 8 hours duration, where each larger dataset contains the data of the smaller dataset. The total of 8 hours form the ASR-Train dataset, which is shown in Table 3.4. The development and test datasets have a duration of 2 hours and there is no overlap between the samples, the datasets do not share paragraphs, which is also shown in Table 3.4. Female speaker (Speaker A) and male speaker (Speaker B) participated equally in the creation of the ASR-Train dataset. All speakers were used for the development and test sets. The suffix "01" stands for the use of the A and B speakers and the suffix "02" for the D and E speakers. The reference texts are selected from the paragraphs in SQuAD-TR that have no answer to their questions. There are 1947 paragraphs that have no answers. 704

paragraphs were used for the training dataset and 176 paragraphs for the development and test datasets. These were subjected to the sentence splitting pre-processing step. This results in a total number of 3,072 sentences, which are used to create a training dataset lasting around 8 hours. 768 sentences are used separately for the development and test sets.

Table 3.4. Naming convention of ASR training datasets.

Name	**Task**	**Description**	**Duration**
ASR-Train	ASR Train	Synthesized with A, B Speakers	8 hours
ASR-Dev-01	ASR Dev.	Synthesized with A, B Speakers	2 hours
ASR-Dev-02	ASR Dev.	Synthesized with D, E Speakers	2 hours
ASR-Test-01	ASR Test	Synthesized with A, B Speakers	2 hours
ASR-Test-02	ASR Test	Synthesized with D, E Speakers	2 hours

The second subset of the synthesized spoken datasets is used for training the spoken QA model after decoding by the best performing ASR model. After training and evaluating the QA model on different QA dataset partitions, the best performing dataset is selected as the reference texts. The detailed explanation of these reference texts can be found in Section 3.2.2. This dataset is of course a subset of SQuAD-TR, which contains 15,868 paragraphs and 63,764 QA pairs. All paragraphs, questions and answers are used for synthesis, resulting in approximately 280 hours. Female and male speakers are used in equal proportions, again using A and B speakers for one set and D and E speakers for the other set, amounting to approximately 560 hours.

Table 3.5. Naming convention of spoken QA training datasets.

Name	**Task**	**Description**	**Duration**
QA-Train-01	QA Train	Synthesized with A, B Speakers	280 hours
QA-Train-02	QA Train	Synthesized with D, E Speakers	280 hours
QA-Dev-01	QA Dev.	Synthesized with A, B Speakers	1 hour
QA-Dev-02	QA Dev.	Synthesized with D, E Speakers	1 hour

Table 3.5. Naming convention of spoken QA training datasets (cont.)

Name	**Task**	**Description**	**Duration**
QA-Test-01	QA Test	Synthesized with A, B Speakers	1 hour
QA-Test-02	QA Test	Synthesized with D, E Speakers	1 hour

As part of our research, further test sets were created for the final subset of synthesized spoken datasets. The source material for these sentences comes from the Turkish split of the XQuAD dataset. It should be recalled that the Turkish section of XQuAD comprises 240 paragraphs and 1,190 QA pairs. After the initial editing, these paragraphs underwent a sentence splitting process that culminated in a total of 1101 sentences. The use of male and female speakers for audio synthesis was carefully considered to ensure diversity and representativeness. As can be seen in Table 3.6, there are two groups of XQuAD variations, one group with the suffix "1hr" and the other group without. This is due to the constraints of "XQuAD-03-1hr" recorded by the human speakers. This is explained in more detail in Section 3.3. For a fair evaluation, we created the datasets "XQuAD-01-1hr" and "XQuAD-02-1hr", which contain exactly the same paragraphs and QA pairs compared to "XQuAD-03-1hr". The partitions that do not have the suffix "1hr" are the complete Turkish split of XQuAD. The same naming convention applies here too. The suffix "01" stands for the use of A and B speakers and the suffix "02" for the use of D and E speakers.

Table 3.6. Naming convention of XQuAD datasets.

Name	**Description**	**Duration**
XQuAD	Only text documents	-
XQuAD-01	Synthesized with A, B Speakers	5 hours
XQuAD-02	Synthesized with D, E Speakers	5 hours
XQuAD-1hr	Only text documents	-
XQuAD-01-1hr	Synthesized with A, B Speakers	1 hour

Table 3.6. Naming convention of XQuAD datasets. (cont.)

Name	**Description**	**Duration**
XQuAD-02-1hr	Synthesized with D, E Speakers	1 hour
XQuAD-03-1hr	Real speakers	1 hour

3.3. Spoken XQuAD-TR

The creation of spoken datasets with human participants represents a major financial and logistical challenge in the field of linguistic and computational research. In particular, the lack of an accessible open-source platform for the compilation of sound data makes it harder. To address this gap, in our study we present a novel open-source web application developed for the efficient collection of sound data. This application streamlines the process by allowing participants to read and record paragraphs and QA pairs. They can then upload their recordings to a central database, as long as they agree with the content of their recordings.

The functionality of the web application is described by a user-centered workflow. At the beginning, participants are presented with a text excerpt, either a paragraph or the corresponding QA pairs. Following the recording phase, participants have the opportunity to review their audio recording before it is officially uploaded to ensure the integrity and quality of the dataset. It is important that each record entry, which includes both a paragraph and the corresponding QA pairs, comes from a single participant only, ensuring a consistent source of speech data.

The technical architecture of the sound collection application includes a front-end interface developed with the React framework and a back-end system developed with the FAST API framework. Data management and user registration are facilitated by the integration of Google's Firebase. Firebase serves as a repository for user information, submission progress tracking and storage of the resulting audio files. The comprehensive workflow and structural design of the application is detailed in Figure 3.3,

which provides a visual representation of the user interaction process and data management framework. Creating a spoken version of the entire Turkish split of XQuAD is a very time-consuming task. We have selected 12 speakers which are 6 females and 6 males to record about the quarter of XQuAD-TR. XQuAD-TR contains 240 paragraphs and 1,190 QA pairs but for this task we have obtained 72 paragraphs and 359 QA pairs recorded by 12 speakers. All individuals are subjected to 6 paragraphs and about 30 QA pairs evenly. This results in 73 minutes of spoken dataset.

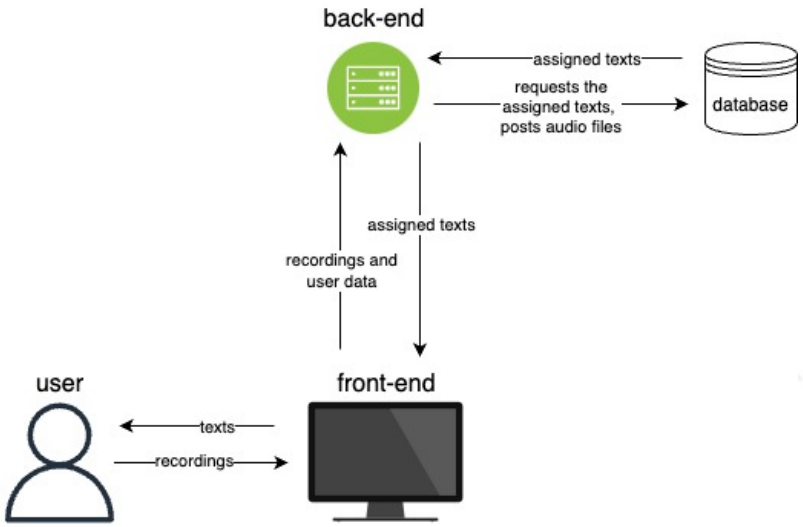

Figure 3.3. Sound collection app workflow.

4. QUESTION ANSWERING SYSTEM

Our QA system is structured like a typical cascade SQA system which is shown in Figure 4.1. SQA system is designed specifically for processing spoken queries. It consists of two main modules: the ASR module, which transcribes spoken queries, and the QA module, which performs reading comprehension on these transcriptions. This configuration corresponds to the main goal of this study, which is to curate a spoken QA dataset and create corresponding benchmarks. This section describes the methods and models used in both modules and provides an overview of the cascade SQA system.

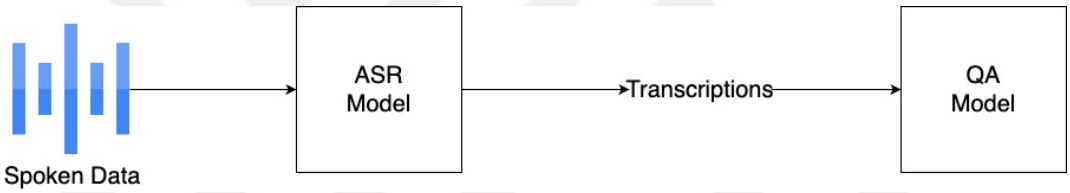

Figure 4.1. A cascade SQA system.

4.1. Automatic Speech Recognition

The XLS-R [47] and Whisper [48] ASR models were used in this study. In addition, the WhisperX [49] framework was used to improve the performance of the Whisper models.

4.1.1. XLS-R

XLS-R [47] is a multilingual ASR model. It is trained to provide accurate speech recognition for 128 languages, including both low resource and high resource languages. The training data of XLS-R comprises about 436,000 hours of publicly available speech data and is not evenly distributed among the languages. Turkish language data comprises about 136 hours.

The model is based on a self-supervised learning method, Wav2Vec2 [50], i.e. it can learn from the data without the need for manual annotation. Wav2Vec2 is used as a feature extractor for the appropriate representations of the speech. Wav2Vec2 consists of a multi-layer CNN, called the encoder network, and a context network (transformer). The encoder network creates latent speech representations from the raw audio signals and the context network uses these representations to create its own unique context representations. XLS-R is trained by taking advantage of two main concepts; contrastive learning and a language model. Contrastive learning helps the model learn to distinguish between distractor patterns and true samples and that's why less annotated data is needed. The language model helps the model to understand the context of the audio signals. XLS-R also takes advantage of transfer learning. When the latent representations obtained from Wav2Vec2 are fed into a neural network, this network can be trained on a specific task such as language translation or classification of bad words, etc. XLS-R has 3 model configurations regarding the number of parameters which are 300 million, 1000 million and 2000 million. Due to the long fine-tuning time required by one billion parameter or more models, we have selected the model with 300 million parameters for the experiments.

4.1.2. Whisper

Whisper is an ASR model which is trained on about 680,000 hours of multilingual speech data which is considered as a massive dataset. The training dataset of Whisper contains high-quality audio recordings as well as noisy samples which are collected from web. This clearly explains the model's robustness to noise which is one of the main reasons that we selected the model for this study. The training dataset of Whisper contains mainly English audio data which corresponds to 438,000 hours and multilingual audio covers about 117,000 hours of the training data. In the training data, there is also about 126,000 hours of training data which contains translations from several languages to English as captions. This enables Whisper to make translations from speech data which is out of the cope of this study. The training dataset has 4,333 hours of Turkish speech data.

Whisper is based on an encoder-decoder transformer architecture which is especially good at processing sequential data. The samples in the training data are chunked at length of 30 seconds and log-mel spectrograms are obtained from those chunks. The encode is trained on those spectrograms. After those spectrograms are encoded, the decoder is trained to infer their corresponding text captions. Whisper comes with various model sizes in terms of number or parameters; tiny (39 million), base (74 million), small (244 million), medium (769 million) and large (1550 million). By the time of this study, large-v2 and large-v3 models are released. These models are the same size of the large model. The large-v2 model, undergoes training for 2.5 times more epochs than large model and incorporates additional regularization. Large-v3 model changed the input size of mel frequency bins from 80 to 128 and it's trained on 1 million hours of weakly labeled audio and 4 million hours of labeled audio by large-v2.

4.1.3. WhisperX

WhisperX [49] is a state-of-the-art framework designed to overcome the challenges of transcribing long audio content with high accuracy, especially when generating precise word-level timestamps. WhisperX builds on the foundations of Whisper, and introduces new methods to improve transcription performance of Whisper models. First, the system segments the audio with the help of voice activity detection (VAD). These segments are then divided into smaller chunks at points of low voice activity, each lasting a maximum of 30 seconds. After this segmentation, the segments undergo two important processes: parallel transcription with Whisper and precise synchronization with a phoneme recognition model to obtain accurate word-level timestamps with improved efficiency. In this study, we have used WhisperX due to its ability to deal with the audio samples greater than 30 seconds because Whisper has a hallucination problem while inferring audio clips longer than 30 seconds. Also, its ability to perform parallel inference enables much faster speech recognition.

4.2. Question Answering

BERT is one of the earliest models based on the famous transformer architecture. It is pretrained on multilingual text data coming from Wikipedia articles of most frequent 104 languages in a self-supervised manner. Only the header bodies of the articles were included. Again, it doesn't require data annotation therefore pretraining on a large corpus becomes feasible. In order to balance the imbalance of the representation of low-resource languages in the pre-training corpus, exponentially smoothed weighting is applied to penalize the likelihood of the inclusion of high resource languages in the pre-training dataset.

It has been trained with 2 main objectives. First, masked language modeling, where the model randomly masks the 15% of tokens in a sentence, allowing the model to learn bidirectional representations, unlike recurrent neural networks. Second, the next sentence prediction which again involves masking. This time, the model masks a sentence from a random pair of sentences and tries to predict whether the second sentence will follow the first. With these training configurations, BERT has learned the inner representation of a language. This has skyrocketed its use in many classification tasks like part of speech tagging, named entity recognition etc.

In this thesis, we utilize the 110 million parameter BERT base multilingual cased model and BERTurk [51], which is the same model but fine-tuned on a Turkish corpus.

4.3. The Overall System

As mentioned earlier, this study comprises a typical cascade spoken QA system based on an ASR and a QA module. The ASR module is shown in Figure 4.2. The datasets referenced in Section 3.2.3 were formulated using the pipeline shown in the orange section of the figure. In particular, paragraphs from the SQuAD-TR dataset that did not contain answerable questions were used to fine-tune the models for ASR. After this fine-tuning, the models were subjected to performance evaluation, both with

their intended test sets and with the paragraphs from the XQuAD-TR dataset recorded by real speakers. This evaluation process facilitated the selection of the best performing ASR model, which was then used to decode the largest speech dataset of the study.

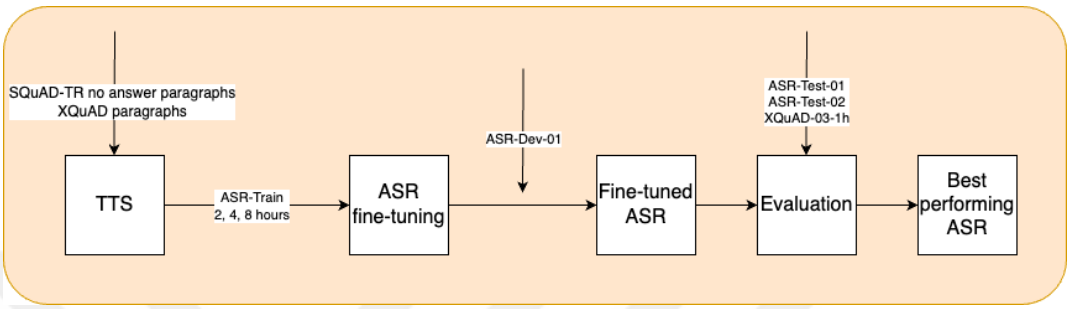

Figure 4.2. ASR pipeline.

In Figure 4.3, the QA model was fine-tuned with different partitions of the SQuAD-TR dataset mentioned in Section 3.2.2. These partitions were crucial for the experiments aimed at determining the optimal answer spans to maximize data extraction from SQuAD-TR. The selection of the best performing QA model was based on its performance against a fixed test set consisting of standard answerable questions. Once the most appropriate partition of the QA model was determined, the performance of the models was evaluated on the Turkish split of the XQuAD dataset.

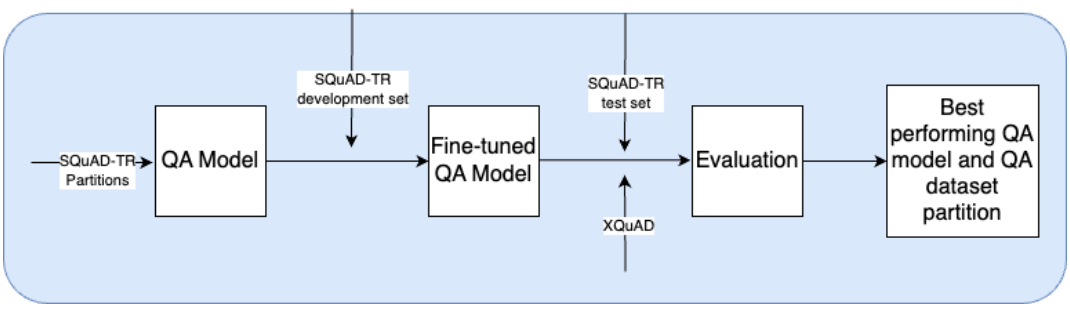

Figure 4.3. Text-based QA pipeline.

Once the optimal QA partition was determined, it was processed by text normalization procedures prior to TTS synthesis. The final set of experiments, shown in the green section of Figure 4.4, involved decoding the synthesized QA partition using the

best performing ASR model. In addition, the multilingual BERT base cased model and BERTurk were trained using the training partition of this synthesized dataset. The culmination of this methodology was to evaluate the performance of this training on different datasets, including variations of XQuAD, spoken XQuAD, and synthesized spoken XQuAD, thus ensuring a comprehensive evaluation of the model's effectiveness across different data representations.

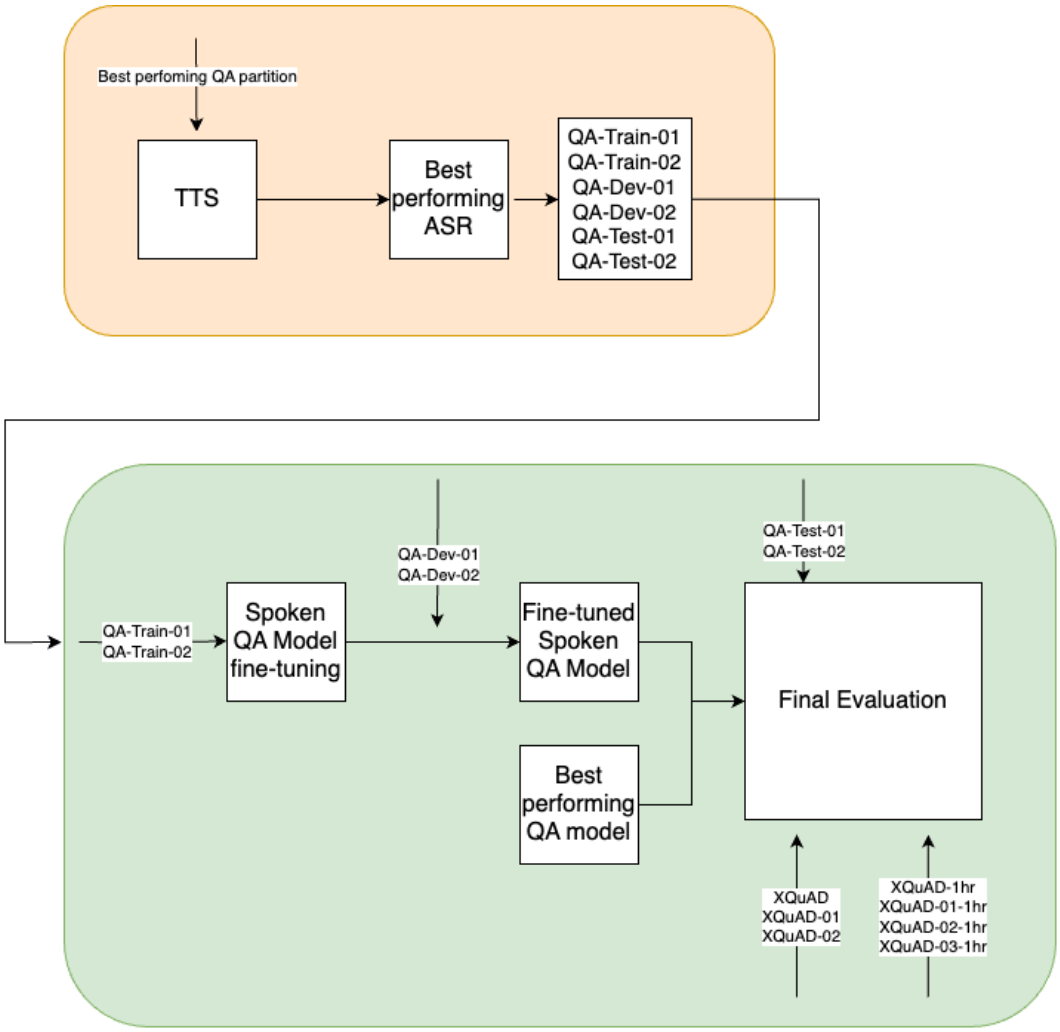

Figure 4.4. SQA pipeline.

5. EXPERIMENTS AND RESULTS

In this section, all experiments and their results are presented to achieve the main goal of this study: the curation of datasets and the creation of benchmarks in the field of spoken QA.

5.1. Experimental Setup

In this section, the experimental frameworks for ASR and QA models and the overall system are outlined.

5.1.1. ASR Models

This study comprises the use of XLS-R and Whisper models and WhisperX framework for the ASR module. Only XLS-R was used in the fine-tuning experiments. Given the significant computational resources required to fine-tune Whisper Large models and the limitation that the Whisper X framework does not support the integration of custom, fine-tuned Whisper models as a backend, fine-tuning processes for Whisper models were omitted in this study. XLS-R fine-tuning experiments were done with the ASR-Train set (8 hours) and its two subsets containing 2 and 4 hours of data which is synthesized with A and B speakers. In conducting these experiments, three different learning rates were systematically evaluated, with the other hyperparameters held constant. Comprehensive details on the hyperparameters for both models are carefully documented in Table 5.1.

Table 5.1. Hyperparameter configuration of XLS-R model

Hyperparameter	**XLS-R**
Learning Rate	5×10^{-4}
Train Batch Size	2

Table 5.1. Hyperparameter configuration of XLS-R model. (cont.)

Hyperparameter	**XLS-R**
Evaluation Batch Size	8
Gradient Accumulation Steps	4
Total Train Batch Size	8
Number of Epochs	20
Training Steps	-
Attention Dropout	0.0
Hidden Dropout	0.0
Feature Projector Dropout	0.0
Mask Time Probability	0.0
Layer Dropout	0.0
CTC Loss Reduction	0.0
Optimizer	ADAM (ϵ =1 $\times 10^{-8}$)

5.1.2. QA Model

The BERT base multilingual cased and BERTurk models were used for the QA task. It is important to divide the set of QA experiments into 2 groups. The first is the extraction of the largest possible subset of SQuAD-TR. For these experiments only BERT base multilingual cased was used. The second group of experiments deals with the main task of machine comprehension, for which both models were used. The hyperparameters of these models in these experiments are shown in Table 5.2.

The experiments for training of BERT base multilingual cased model were conducted with 7 different datasets with thresholds 0, 0.2, 0.4, 0.6, 0.8 and 1. These dataset configurations were clearly explained in Section 3.2.2. The hyperparameters were kept constant during all these experiments and are as follows:

Table 5.2. Hyperparameter configuration of QA models.

Hyperparameter	**BERT Base Multilingual**	**BERTurk**
Learning Rate	5×10^{-4}	5×10^{-4}
Train Batch Size	16	16
Evaluation Batch Size	16	16
Number of Epochs	3.0	3.0
Maximum Sequence Length	384	384
Weight Decay	0.01	0.01

5.2. Results and Discussion

In this section, we discuss the experiments and their results which constitute the general approach of the ASR and QA modules, to explore the advantages and limitations.

5.2.1. Results of the ASR Models

Performance of the ASR models are evaluated with WER. It is important to note that the output of the best-performing model undergoes several normalization steps, including case reduction, punctuation removal and numerical normalization, to ensure a standardized WER evaluation. WER compares the transcribed text output by the ASR system with the reference text and measures the proportion of inserted errors. WER is calculated by counting the number of substitutions, deletions and insertions required for the ASR output to match the reference, normalized by the number of words in the reference.

ASR experiments can be divided into 2 groups. First set of experiments are conducted for the fine-tuning of XLS-R. The second group is done for the decoding of spoken QA data.

Regarding the first group of experiments, all of the ASR models in the experiments, were subjected to hyperparameter tuning, using number of training steps, on the development set, ASR-Dev-01 which has the same speakers (A and B speakers) as in the ASR-Train set. Figure 5.1 shows validation WERs during the fine-tuning of XLS-R for 2, 4 and 8 hours of ASR training data respectively. It is important to note that the training steps in the graphs are an independent concept, but one that correlates with the number of epochs. The training steps usually refer to the number of times the training loop runs over a batch of data, while the number of epochs refers to the number of passes of the entire dataset through the network during training. The number of training steps is calculated by multiplying the number of epochs by the total number of batches, where the total number of batches is determined by dividing the total number of samples by the batch size. Since the number of epochs was kept constant at 3 in all experiments, the number of training steps varies between the graphs due to the different number of samples between the datasets. As an expected behaviour, validation WER drops when the size of the training data increases.

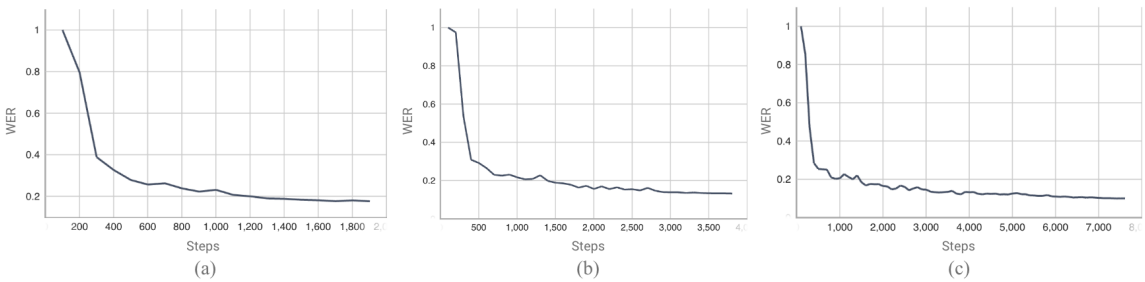

Figure 5.1. Validation WERs of XLS-R fine-tuning.

Among the fine-tuned ASR models, the best performance was obtained with the 8-hour data. The performance of ASR-Test-01 and ASR-Test-02 were evaluated with this model. The results are demonstrated in Table 5.3. Initial baseline tests of the XLS-R model resulted in a WER of 10.0% on ASR-Test-01 which is synthesized with the same set of speakers that the model's training data synthesized with. It is clear that the performance decreases to 22.3% when the test data is synthesized with different set of speakers, D and E speakers, which is due to the different distribution of the data. Despite the different set of speakers, the fine-tuning with ASR-Test-01 worked

for ASR-Test-02 because the adaptation was also made to the machine-synthesized speech as well as the speaker.

Table 5.3. XLS-R test results.

Test Dataset	**WER (%)**
ASR-Test-01	10.0
ASR-Test-02	22.3

The robustness of the XLS-R model has been the subject of investigation, with the prevailing opinion suggesting the need for fine-tuning to adapt to specific data distributions. When evaluating the XLS-R model fine-tuned with synthesized speech on the XQuAD-03-1hr dataset, which contains data from real speakers, the model performed markedly poorly, suggesting overfitting to the distribution of machine-synthesized speech data. This led us to investigate the readily available Whisper models as an alternative.

For the ASR experiments with off-the-shelf Whisper models, we selected the small model as the bare minimum due to its parameter size (244 million) that resembles with XLS-R. We also tried medium, large, large-v2 and large-v3 models. The results on ASR-Test-01 and ASR-Test-02 are given in Table 5.4.

Table 5.4. Test results of the Whisper models.

Whisper Model	**ASR-Test-01 WER (%)**	**ASR-Test-02 WER (%)**	**XQuAD-03-1hr WER (%)**
Small	15.6	15.3	22.3
Medium	11.8	11.6	26.7
Large	11.1	10.8	15.6
Large-v2	10.6	10.1	14.7
Large-v3	9.8	9.4	13.0

A look at Table 5.4 shows that the large-v3 model performs best, which was expected as it is the largest model in terms of parameter size and was trained with the largest amount of data compared to the other models.

We also tested the robustness of the model with the recordings coming from real speakers, XQuAD-03-1hr which is mentioned in Section 3.3. The results are given in Table 5.4. The introduction of noise from recordings with real speakers led to a drop in performance, although not as pronounced as with the XLS-R model.

Although all audio samples were synthesized after sentence splitting, a significant number of samples exceeded 30 seconds in duration, a factor that triggered hallucination effects in the Whisper models and subsequently penalized the WER of the output. Given these challenges, we explored the use of the Whisper X framework with identical models in the same datasets to evaluate potential improvements in model robustness and performance accuracy. The results are shown in Table 5.5.

Table 5.5. Test results of the WhisperX models.

WhisperX Model	**ASR-Test-01 WER (%)**	**ASR-Test-02 WER (%)**	**XQuAD-03-1hr WER (%)**
Small	14.6	13.8	20.9
Medium	11.4	10.9	16.1
Large	9.7	9.3	12.7
Large-v2	10.0	9.7	13.9
Large-v3	9.7	9.3	11.7

The results indicate a modest increase in performance, which can be attributed to the use of Whisper X. However, the amount of data in the ASR-Test sets is significantly smaller compared to the QA-Train-01 and QA-Train-02 datasets. Consequently, the probability of encountering audio samples longer than 30 seconds is much higher in these larger datasets. In addition, the implementation of batch inference within Whisper X doubles the real-time factor compared to conventional Whisper models.

For this reason, we chose the large-v3 backed Whisper X framework as the ASR model for batch decoding tasks related to the QA-Train-01 and QA-Train-02 datasets.

5.2.2. Results of the QA Model

Evaluation of QA models is generally done by using 2 important metrics, EM and F1 score. EM measures the percentage of predictions that exactly match one of the ground truth answers, providing a rigorous assessment of accuracy. The F1 score, on the other hand, considers both precision and recall of predictions and calculates the harmonic mean of these two metrics to evaluate a model's ability to find relevant information while minimizing irrelevant output.

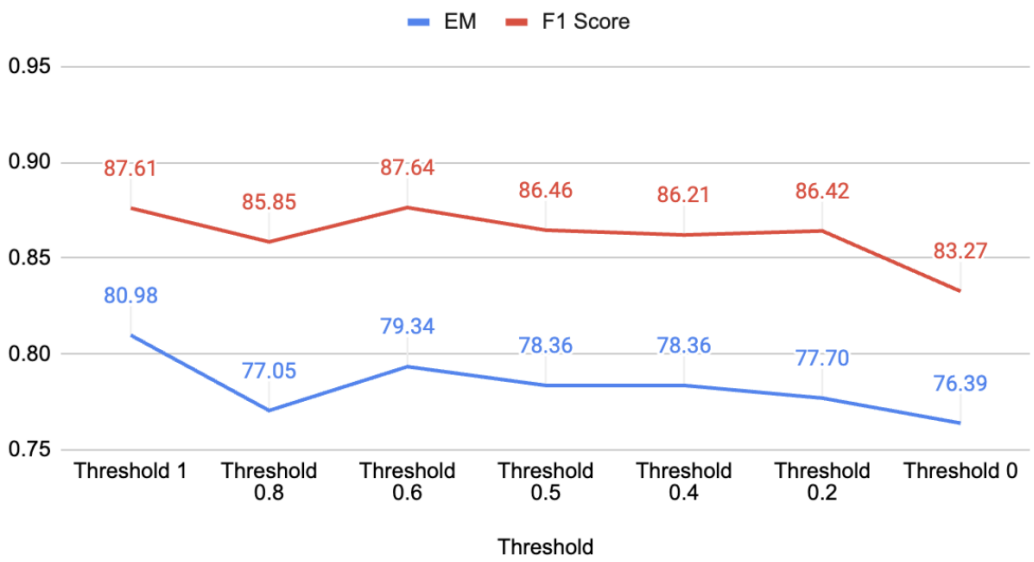

Figure 5.2. BERT results on QA-Dev.

As mentioned earlier, the first set of QA experiments is performed for extracting the largest possible subset of SQuAD-TR. This is done by applying different thresholds to the text-sequence matching algorithm to improve the quality of the answers obtained from a machine-translated SQuAD-TR dataset. In this section, we first explain how we selected the largest possible dataset. Only BERT base multilingual cased is used for these experiments. Figure 5.2 shows the results of each model with different threshold

values on the evaluation set and Figure 5.3 shows the results on the test set. Please remember that the QA-Dev partition is the reference texts, a subset of SQuAD-TR, for the synthesis of QA-Dev-01 and QA-Dev-02. QA-Test corresponds to the reference texts used for the synthesis of QA-Test-01 and QA-Test-02.

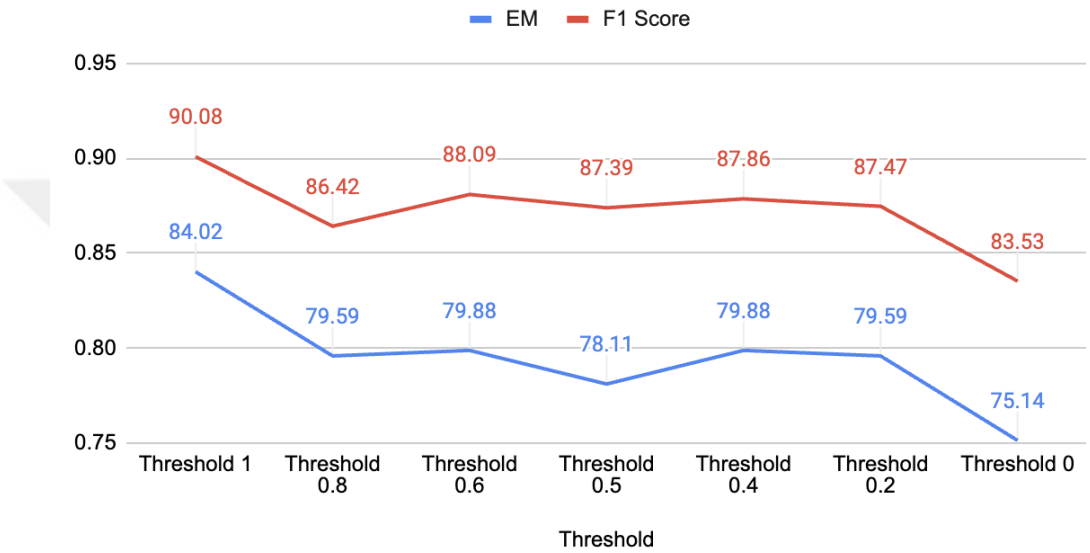

Figure 5.3. BERT results on QA-Test.

A look at the Figure 5.2 shows that models using threshold values of 0.6 and 1 deliver the most favorable results. This trend is also reflected in the results of the test set shown in Figure 5.3, where the model with a threshold of 1 performs better than its counterparts. A threshold of 1 primarily involves the addition of QA pairs whose EM answers ending without spaces, maintaining a structure that is very similar to the original EM answers. The reason for implementing such a threshold is to maximize the extraction of QA pairs and thus include a certain amount of noise to improve the generalization capabilities of the subsequent spoken QA model. Setting the threshold to 0.6 leads to the inclusion of approximately 15,000 additional QA pairs compared to the model with threshold 1, effectively doubling the number of original EM-QA pairs from the SQuAD-TR dataset. To illustrate these results, we compared both models with thresholds 1 and 0.6 to the XQuAD dataset. The results are shown in Table 5.6. This comparative analysis aims to provide a comprehensive understanding

of the impact of threshold settings on model performance and the effectiveness of data augmentation.

Table 5.6. Test results of QA models on XQuAD.

Model	**EM (%)**	**F1 (%)**
Threshold 1	50.42	65.40
Threshold 0.6	54.20	68.69

It is to be expected that the model with a threshold of 0.6 performs better than the model with a threshold of 1 on the XQuAD dataset, which comes from a different distribution. Therefore, we chose the partition of the QA dataset with a threshold of 0.6 for the synthesis of Spoken SQuAD-TR.

5.2.3. Results of the Spoken QA System

The thresholded QA partition set contains 15,868 paragraphs and 63,764 QA pairs. As mentioned earlier, this dataset was synthesized in roughly 30 hours using Google's TTS API with A, B speakers and D, E speakers, resulting in total of 560 hours spoken data. WhisperX framework backed with Whisper large-v3 model was used to decode all samples in both datasets. Using 2 Tesla V100 GPUs, each set was decoded in 16 hours. The results of the decoding of paragraph sentences and questions was formed to the original format of SQuAD-TR with their unique identifiers. The answers were selected from the original SQuAD-TR. Note that this results in the eliminating answers containing digits, special characters such as percent sign. In the process of generating the spoken QA dataset we faced the problem of locating the answers in ASR transcriptions. The answer start information in the original dataset was not true anymore. This applies also to the other decoded QA sets such as the variations of QA-Test, XQuAD and XQuAD 1 hour versions. The following procedure was applied to all of these decoded QA datasets. As we did before, we first checked the EM answers with the original answer start information. If the answer start information

doesn't correspond to the span in the paragraph, we removed the parantheses and the whitespaces at the beginning and the end of the answer. Then, we performed a basic search to find the first occurrence of the answer in the paragraph to update the answer start information and at the same time we validated the new answer start information. The number of QA pairs of all datasets before and after the elimination of incorrectly found answers is shown in Table 5.7.

Table 5.7. QA pairs before and after decoding across datasets.

Dataset	**Before Elimination**	**After Elimination**	**Intersection**
QA-Train-01	63,764	38,167	36,110
QA-Train-02	63,764	38,142	36,110
QA-Test-01	300	120	118
QA-Test-02	300	123	118
XQuAD-01	1190	806	784
XQuAD-02	1190	816	784
XQuAD-01-1hr	359	252	204
XQuAD-02-1hr	359	242	204
XQuAD-03-1hr	359	222	204

QA-Test family has an intersection of 118 QA pairs, XQuAD family has an intersection of 784 QA pairs and XQuAD-1hr family has an intersection of 204 QA pairs. For standardized testing of all of the models, each set's own intersection subset is used.

Since ASR errors have a catastrophic impact on the performance of SQA models, it is important to mention the WERs across datasets. Table 5.8 shows the WERs at both paragraph and question level for all datasets before eliminating the wrong answers found. Table 5.9 shows the WERs after elimination and Table 5.10 shows the WERs for the intersection of each set, which is crucial for the evaluation of the SQA models in upcoming paragraphs.

Table 5.8. WER metrics before elimination.

Dataset	**Paragraph WER (%)**	**Question WER (%)**	**Total WER (%)**
QA-Train-01	10.7	8.2	10.1
QA-Train-02	10.6	8.1	10.0
QA-Test-01	9.8	7.7	9.3
QA-Test-02	9.8	7.4	9.3
XQuAD-01	11.7	6.6	10.3
XQuAD-02	11.8	6.4	10.3
XQuAD-01-1hr	10.6	5.4	9.2
XQuAD-02-1hr	10.6	5.2	9.2
XQuAD-03-1hr	13.0	8.0	11.7

Significant decreases in WERs between Table 5.8 and Table 5.9 are expected due to the drastic changes in the number of QA pairs during the elimination process, as illustrated in Table 5.7. However, there is no significant drop between Table 5.9 and Table 5.10 as the number of QA pairs after elimination and at the intersection are similar.

Table 5.9. WER metrics after elimination.

Dataset	**Paragraph WER (%)**	**Question WER (%)**	**Total WER (%)**
QA-Train-01	7.6	6.0	7.3
QA-Train-02	7.4	5.8	7.2
QA-Test-01	7.8	6.3	7.6
QA-Test-02	7.6	6.2	7.5
XQuAD-01	8.4	5.1	7.7
XQuAD-02	8.4	5.2	7.7
XQuAD-01-1hr	7.2	4.3	6.6

Table 5.9. WER metrics after elimination. (cont.)

Dataset	**Paragraph WER (%)**	**Question WER (%)**	**Total WER (%)**
XQuAD-02-1hr	7.4	4.1	6.7
XQuAD-03-1hr	8.6	4.9	7.9

Table 5.10. WER metrics at intersection.

Dataset	**Paragraph WER (%)**	**Question WER (%)**	**Total WER (%)**
QA-Train-01	7.5	6.0	7.3
QA-Train-02	7.4	5.8	7.1
QA-Test-01	7.8	6.1	7.6
QA-Test-02	7.7	6.2	7.5
XQuAD-01	8.4	5.1	7.7
XQuAD-02	8.4	5.1	7.7
XQuAD-01-1hr	7.2	4.5	6.7
XQuAD-02-1hr	7.5	3.7	6.8
XQuAD-03-1hr	8.6	5.0	8.0

The training of the BERT base multilingual cased model and BERTurk model with QA-Train-01 and QA-Train-02 is done with the same hyperparameters that is shown in Table 5.2. The training of one model takes approximately 3 hours with a single Tesla V100 GPU. For the evaluation, EM performance of the models in the test set partitions is shown in Table 5.11, whereas the F1 scores are depicted in Table 5.12. In these tables, the labels of “00”, “01” corresponds to the associated partition, QA-Test-00, QA-Test-01 respectively and the model name Model-00 denotes the models that were trained using the reference texts. The actual model is specified in the associated “QA Model” column cell. These naming conventions are applicable to all tables in this section. Model-01 models were trained on the ASR transcriptions of the reference texts,

i.e. QA-Train-01 synthesized with A and B speakers. Model-02 models were trained on the ASR transcriptions of the reference texts, QA-Train-02, synthesized with D and E speakers. As outlined previously, answers were selected from the reference texts but questions were selected from the ASR transcriptions.

Table 5.11. EM performance of BERT models on QA-Test datasets.

Model Name	**QA Model**	**00**	**01**	**02**
Model-00	BERTmultilingual	79.8	75.0	73.2
Model-01	BERTmultilingual	68.8	79.2	79.7
Model-02	BERTmultilingual	68.2	75.8	78.0
Model-00	BERTurk	79.0	72.5	74.0
Model-01	BERTurk	67.9	74.2	74.8
Model-02	BERTurk	69.5	75.8	78.1

Table 5.12. F1 scores of BERT models on QA-Test datasets.

Model Name	**QA Model**	**00**	**01**	**02**
Model-00	BERTmultilingual	88.1	82.4	80.9
Model-01	BERTmultilingual	76.7	86.6	85.9
Model-02	BERTmultilingual	77.0	83.1	85.4
Model-00	BERTurk	88.6	81.7	81.5
Model-01	BERTurk	78.4	82.5	82.8
Model-02	BERTurk	78.8	83.3	86.4

Both results, EM and F1 scores show similar trend on the test partitions of the SQuAD-TR. The Model-00 (BERTmultilingual) and Model-00 (BERTurk) trained with only the text documents, thresholded version of SQuAD-TR, faces a performance drop while inferring the ASR decoded partitions because ASR errors are introduced as seen in the Table 5.10. At the same time, Model-01 and Model-02 which are trained with the ASR transcriptions, they perform better at the inference of ASR decoded partitions. This is reasonable because QA-Train-01 and QA-Train-02 has a more similar

distribution with QA-Test-01 and QA-Test-02 compared to the initial QA-Test. Same is applicaple to Model-00 (BERTurk) and Model-01 (BERTurk) models as well. Also, BERTurk models generally perform better compared to BERT base multilingual cased models due to its large Turkish pre-training data.

The results of the tests on XQuAD family are shown in Table 5.13 and Table 5.14. XQuAD is the reference text, XQuAD-01 is the spoken version of XQuAD synthesized with A and B speakers. XQuAD-02 is synthesized with D and E speakers. When we look at the results on XQuAD partitions, the patterns of the results on the test set only continues for the performance drop of Model-00 (BERTmultilingual) and Model-00 (BERTurk) at the inference of ASR decoded datasets. This can be explained with the WER shown in Table 5.10.

The reason for the better performance of the Model 00 models trained on the reference texts compared to the models trained on ASR transcripts could be the size of the training dataset of the Model-00 models. The reference texts contain 63,764, while the ASR transcriptions contain about 38,000 QA pairs. Therefore, we conducted a series of experiments in which all 3 models were trained on the intersection of QA-Train, QA-Train-01 and QA-Train-02, resulting in 36,110 QA pairs.

Table 5.13. EM performance of BERT models on XQuAD datasets.

Model Name	**QA Model**	**00**	**01**	**02**
Model-01	BERTmultilingual	54.2	52.0	51.2
Model-01	BERTmultilingual	49.7	50.3	50.7
Model-02	BERTmultilingual	50.2	50.4	50.5
Model-00	BERTurk	55.0	54.1	54.3
Model-01	BERTurk	51.2	53.1	52.9
Model-02	BERTurk	51.5	53.5	53.2

Table 5.14. F1 scores of BERT models on XQuAD datasets.

Model Name	**QA Model**	**00**	**01**	**02**
Model-00	BERTmultilingual	68.7	65.8	65.4
Model-01	BERTmultilingual	63.7	64.1	64.3
Model-02	BERTmultilingual	64.1	64.2	64.1
Model-00	BERTurk	70.2	68.1	68.4
Model-01	BERTurk	66.3	67.1	67.6
Model-02	BERTurk	67.0	67.6	67.6

Model-01 (BERTmultilingual), Model-01 (BERTurk) and Model-02 (BERTurk) tends to show the same pattern, but these changes are not significant and the performance of Model-02 (BERTmultilingual) does not follow the same pattern as the performance on the test sets. This might be due to the distributions of the datasets. The test dataset has a similar distribution to the training dataset. Both are originally machine-translated datasets that contain the same type of noise. However, XQuAD is a human-curated dataset. It therefore has a different distribution. Again, it is clear that the extensive Turkish pre-training of the multilingual BERT model, which led to BERTurk, has improved the QA performance of BERT for Turkish.

Table 5.15. EM performance of BERT models on XQuAD-1hr datasets.

Model Name	**QA Model**	**00**	**01**	**02**	**03**
Model-00	BERTmultilingual	47.4	47.6	45.8	46.4
Model-01	BERTmultilingual	42.1	45.6	45.8	44.6
Model-02	BERTmultilingual	44.3	45.6	44.6	44.6
Model-00	BERTurk	51.0	51.6	49.0	49.6
Model-01	BERTurk	45.4	45.6	46.2	45.5
Model-02	BERTurk	45.7	48.4	45.8	46.0

Table 5.16. F1 scores of BERT models on XQuAD-1hr datasets.

Model Name	**QA Model**	**00**	**01**	**02**	**03**
Model-00	BERTmultilingual	63.8	60.9	60.5	59.8
Model-01	BERTmultilingual	58.3	60.0	59.9	58.3
Model-02	BERTmultilingual	59.0	58.5	58.4	58.0
Model-00	BERTurk	66.7	63.8	63.0	61.2
Model-01	BERTurk	62.1	60.7	61.9	60.1
Model-02	BERTurk	62.8	61.7	61.4	60.0

Table 5.16 shows that Model-00 (BERTurk) performs best on XQuAD-1hr, which is derived from reference texts. However, its performance gradually declines on the synthesized and human-recorded partitions. Model-00 (BERTmultilingual) exhibits a similar trend. Model-01 (BERTmultilingual) shows a slight improvement when inferring the decoded partitions, which is to be expected since the QA-Train-01 dataset as well as XQuAD-01-1hr and XQuAD-02-1hr contain noise originating from ASR errors. In contrast, Model-02 (BERTmultilingual), Model-01 (BERTurk) and Model-02 (BERTurk) do not follow this pattern. In addition, the extensive Turkish pre-training of the multilingual BERT model, which led to the creation of BERTurk, apparently improved the QA capabilities of BERT for Turkish. Finally, the performance of all models is lowest for XQuAD-03-1hr, coming from the real speakers. This is quite natural as the WER is higher when decoding this dataset than the other datasets as seen in Table 5.10.

The reason for the better performance of the Model-00 models trained on the reference texts compared to the models trained on ASR transcripts could be the size of the training dataset of the Model-00 models. The reference texts contain 63,764, while the ASR transcripts contain about 38,000 QA pairs. Therefore, we conducted a series of experiments in which all 3 models were trained on the intersection of QA-Train, QA-Train-01 and QA-Train-02, resulting in 36,110 QA pairs. All fine-tuned models in these datasets are based on BERT base multilingual cased. Table 5.17 shows the results

regarding EM performance and Table 5.18 shows the results of the F1 scores. If we look at the F1 scores for QA-Test-00, which comes from the reference texts there are slight drops between the inital BERT multilingual and the models trained on intersection. This is understandable as QA-Test-00 comes from a similar distribution as the QA-Train dataset. However, the increase in performance of Model-00 (intersection) on QA-Test-02 is unexpected. This could be due to the noise caused by the ASR errors in QA-Test-02. There is a slight decrease in performance when comparing Model-01 and Model-01 (intersection), with Model-02 (intersection) also showing a slight decrease compared to Model-02.

Table 5.17. EM perf. of BERT multilingual models on QA-Test datasets.

Model Name	**00**	**01**	**02**
Model-00	79.8	75.0	73.2
Model-01	68.8	79.2	79.7
Model-02	68.2	75.8	78.0
Model-00 (intersection)	79.3	75.8	78.9
Model-01 (intersection)	66.9	78.3	77.2
Model-02 (intersection)	68.5	75.8	78.0

Table 5.18. F1 scores of BERT multilingual models on QA-Test datasets.

Model Name	**00**	**01**	**02**
Model-00	88.1	82.4	80.9
Model-01	76.7	86.6	85.9
Model-02	77.0	83.1	85.4
Model-00 (intersection)	87.6	82.3	84.7
Model-01 (intersection)	76.3	83.9	83.4
Model-02 (intersection)	76.9	82.4	84.5

The results of the tests of BERT multilingual models on the XQuAD family are shown in Table 5.19 and Table 5.20. In contrast to the experiments on QA-Test, Model-00 (intersection) on XQuAD shows a significant drop compared to initial Model-00. The reason for this lies in the fact that XQuAD was translated by humans and therefore comes from a different distribution than QA-Train. The performance of Model-00 (intersection) also decreases slightly for XQuAD-01 and XQuAD-02. Significant changes in the performance of Model-01 and Model-02 are not visible, which is to be expected as the training data has not changed significantly.

Table 5.19. EM perf. of BERT multilingual models on XQuAD datasets.

Model Name	**00**	**01**	**02**
Model-01	54.2	52.0	51.2
Model-01	49.7	50.3	50.7
Model-02	50.2	50.4	50.5
Model-00 (intersection)	49.8	50.5	50.4
Model-01 (intersection)	59.7	52.2	52.3
Model-02 (intersection)	49.8	50.4	50.0

Table 5.20. F1 scores of BERT multilingual models on XQuAD datasets.

Model Name	**00**	**01**	**02**
Model-00	68.7	65.8	65.4
Model-01	63.7	64.1	64.3
Model-02	64.1	64.2	64.1
Model-00 (intersection)	63.9	64.6	64.2
Model-01 (intersection)	64.1	65.1	65.2
Model-02 (intersection)	63.9	64.2	64.4

Table 5.22 shows that the same pattern persists for Model-00 (intersection). Again, there is a significant decrease on XQuAD-1hr and a slight decrease is also observed for XQuAD-01-1hr and XQuAD-02-1hr, the synthesized versions. This could

be due to the different distribution of XQuAD compared to SQuAD-TR, which makes up QA-Train. This decrease is not visible in Model-00 (intersection) on XQuAD-03-1hr, which was recorded by real speakers. Model-01 (intersect) also shows a slight decrease on XQuAD-02-1hr, which is unexpected as there are no significant changes between the training datasets of the models that are fine-tuned by ASR transcripts.

Table 5.21. EM perf. of BERT multilingual models on XQuAD-1hr datasets.

Model Name	**00**	**01**	**02**	**03**
Model-00	47.4	47.6	45.8	46.4
Model-01	42.1	45.6	45.8	44.6
Model-02	44.3	45.6	44.6	44.6
Model-00 (intersection)	42.1	44.4	43.8	46.4
Model-01 (intersection)	42.3	47.2	45.0	44.1
Model-02 (intersection)	44.6	46.0	42.6	47.3

Table 5.22. F1 scores of BERT multilingual models on XQuAD-1hr datasets.

Model Name	**00**	**01**	**02**	**03**
Model-00	63.8	60.9	60.5	59.8
Model-01	58.3	60.0	59.9	58.3
Model-02	59.0	58.5	58.4	58.0
Model-00 (intersection)	58.6	59.2	58.2	60.0
Model-01 (intersection)	58.3	60.3	57.9	56.9
Model-02 (intersection)	59.7	59.2	57.6	58.9

6. CONCLUSION

To summarize, this study represents advancements in the field of SQA, with a focus on Turkish. It takes an in-depth look at the unique challenges associated with adapting QA systems for Turkish spoken data, starting with the creation of spoken datasets derived from SQuAD-TR and the Turkish partition of XQuAD. This research focuses on the iterative refinement of state-of-the-art ASR models and frameworks, in combination with the development of robust SQA models that can process and interpret the Turkish language. Another notable aspect of this study is the integration of advanced text normalization and TTS systems, leading to a thorough evaluation of the performance of the SQA system in different speech and text datasets. Results show that, as expected, Model-00 models, which were trained solely on text-based data, exhibited a degradation in performance when applied to ASR decoded output, which is a direct consequence of the ASR errors. Models trained with ASR decoded data performed better on similar ASR decoded QA-Test partitions, confirming the hypothesis that congruence between the distributions of the training and test data increases the performance of the model. Finally, BERTurk models generally perform better compared to BERT base multilingual cased models especially on XQuAD because BERTurk is pre-trained on a larger dataset of Turkish text. This study enriches our understanding of SQA systems in the context of low-resource languages and is an important step towards optimizing their efficiency and applicability.

REFERENCES

1. Green Jr, B. F., A. K. Wolf, C. Chomsky and K. Laughery, "Baseball: An Automatic Question-answerer", *Papers Presented at the May 9-11, 1961, Western Joint Institute of Radio Engineers - American Institute of Electrical Engineers - Association for Computing Machinery Computer Conference*, pp. 219–224, 1961.

2. Robertson, S., S. Walker, S. Jones, M. M. Hancock-Beaulieu and M. Gatford, "Okapi at TREC-3", *Overview of the Third Text REtrieval Conference (TREC-3)*, pp. 109–126, 1995.

3. Ravichandran, D. and E. Hovy, "Learning Surface Text Patterns for a Question Answering System", P. Isabelle, E. Charniak and D. Lin (Editors), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 41–47, Philadelphia, Pennsylvania, USA, 2002.

4. Hirschman, L., M. Light, E. Breck and J. D. Burger, "Deep Read: A Reading Comprehension System", *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 325–332, College Park, Maryland, USA, 1999.

5. Riloff, E. and M. Thelen, "A Rule-based Question Answering System for Reading Comprehension Tests", *2000 Association for Natural Language Processing - North American Association for Computational Linguistics Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*, 2000.

6. Ng, H. T., L. H. Teo and J. L. P. Kwan, "A Machine Learning Approach to Answering Questions for Reading Comprehension Tests", *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 124–132, Hong Kong, China, 2000.

7. Richardson, M., C. J. Burges and E. Renshaw, "MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text", D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu and S. Bethard (Editors), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 193–203, Seattle, Washington, USA, 2013.

8. Rajpurkar, P., J. Zhang, K. Lopyrev and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text", J. Su, K. Duh and X. Carreras (Editors), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, 2016.

9. Wang, S. and J. Jiang, "Machine Comprehension Using Match-lstm and Answer Pointer", *arXiv preprint arXiv:1608.07905*, 2016.

10. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", J. Burstein, C. Doran and T. Solorio (Editors), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019.

11. Jun, C., H. Jang, M. Sim, H. Kim, J. Choi, K. Min and K. Bae, "ANNA: Enhanced Language Representation for Question Answering", S. Gella, H. He, B. P. Majumder, B. Can, E. Giunchiglia, S. Cahyawijaya, S. Min, M. Mozes, X. L. Li, I. Augenstein, A. Rogers, K. Cho, E. Grefenstette, L. Rimell and C. Dyer (Editors), *Proceedings of the 7th Workshop on Representation Learning for NLP*, pp. 121–132, Dublin, Ireland, 2022.

12. Yamada, I., A. Asai, H. Shindo, H. Takeda and Y. Matsumoto, "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention", B. Webber, T. Cohn, Y. He and Y. Liu (Editors), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6442–6454, Online, 2020.

13. Joshi, M., D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer and O. Levy, "SpanBERT: Improving Pre-training by Representing and Predicting Spans", *arXiv preprint arXiv:1907.10529*, 2019.

14. Rajpurkar, P., R. Jia and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD", I. Gurevych and Y. Miyao (Editors), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, 2018.

15. Rajpurkar, P., "SQuAD Explorer", `https://rajpurkar.github.io/SQuAD-explorer/`, 2024, accessed: April 8, 2024.

16. Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", *CoRR*, Vol. abs/1909.11942, 2019.

17. Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *CoRR*, Vol. abs/1907.11692, 2019.

18. Lee, C.-H., H.-y. Lee, S.-L. Wu, C.-L. Liu, W. Fang, J.-Y. Hsu and B.-H. Tseng, "Machine Comprehension of Spoken Content: TOEFL Listening Test and Spoken SQuAD", *Institute of Electrical and Electronics Engineers - Association for Computing Machinery Transactions on Audio, Speech, and Language Processing*, Vol. 27, No. 9, pp. 1469–1480, 2019.

19. Walker, W., P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf and J. Wölfel, "Sphinx-4: A Flexible Open Source Framework for Speech Recognition", *Sun Microsystems*, 2004.

20. Lee, C.-H., S.-L. Wu, C.-L. Liu and H.-y. Lee, "Spoken SQuAD: A Study of Mitigating the Impact of Speech Recognition Errors on Listening Comprehension",

Proc. Interspeech 2018, pp. 3459–3463, 2018.

21. Su, D. and P. Fung, "Improving Spoken Question Answering Using Contextualized Word Representation", *2020 Institute of Electrical and Electronics Engineers International Conference on Acoustics, Speech, and Signal Processing*, pp. 8004–8008, 2020.

22. Ünlü, M. and E. Arisoy, "Uncertainty-Aware Representations for Spoken Question Answering", *2021 Institute of Electrical and Electronics Engineers Spoken Language Technology Workshop*, pp. 943–949, 2021.

23. You, C., N. Chen and Y. Zou, "Knowledge Distillation for Improved Accuracy in Spoken Question Answering", *2021 Institute of Electrical and Electronics Engineers International Conference on Acoustics, Speech, and Signal Processing*, pp. 7793–7797, 2021.

24. Budur, E., R. Özçelik, D. Soylu, O. Khattab, T. Güngör and C. Potts, "Building Efficient and Effective OpenQA Systems for Low-Resource Languages", *arXiv preprint arXiv:2401.03590*, 2024.

25. Yang, Z., P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov and C. D. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering", E. Riloff, D. Chiang, J. Hockenmaier and J. Tsujii (Editors), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018.

26. Joshi, M., E. Choi, D. Weld and L. Zettlemoyer, "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension", R. Barzilay and M.-Y. Kan (Editors), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017.

27. TQuad, "TQUAD/Turkish-NLP-Qa-Dataset: Turkish Question Answering Dataset", `https://github.com/TQuad/turkish-nlp-qa-dataset`, 2024, accessed January 17, 2024.

28. Soygazi, F., O. Çiftçi, U. Kök and S. Cengiz, "THQuAD: Turkish Historic Question Answering Dataset for Reading Comprehension", *2021 6th International Conference on Computer Science and Engineering*, pp. 215–220, 2021.

29. Ünlü, M., E. Arisoy and M. Saraçlar, "Question Answering for Spoken Lecture Processing", *2019 Institute of Electrical and Electronics Engineers International Conference on Acoustics, Speech, and Signal Processing*, pp. 7365–7369, 2019.

30. Artetxe, M., S. Ruder and D. Yogatama, "On the Cross-lingual Transferability of Monolingual Representations", D. Jurafsky, J. Chai, N. Schluter and J. Tetreault (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4623–4637, Online, 2020.

31. Ünlü Menevşe, M., Y. Manav, E. Arisoy and A. Özgür, "A Framework for Automatic Generation of Spoken Question-Answering Data", Y. Goldberg, Z. Kozareva and Y. Zhang (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2022*, pp. 4659–4666, Abu Dhabi, United Arab Emirates, 2022.

32. Lim, S., M. Kim and J. Lee, "KorQuAD1.0: Korean QA Dataset for Machine Reading Comprehension", *CoRR*, Vol. abs/1909.07005, 2019.

33. D'Hoffschmidt, M., W. Belblidia, Q. Heinrich, T. Brendlé and M. Vidal, "FQuAD: French Question Answering Dataset", T. Cohn, Y. He and Y. Liu (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2020*, pp. 1193–1208, Online, 2020.

34. Efimov, P., L. Boytsov and P. Braslavski, "SberQuAD - Russian Reading Compre-

hension Dataset: Description and Analysis", *CoRR*, Vol. abs/1912.09723, 2019.

35. Möller, T., J. Risch and M. Pietsch, "GermanQuAD and GermanDPR: Improving Non-English Question Answering and Passage Retrieval", A. Fisch, A. Talmor, D. Chen, E. Choi, M. Seo, P. Lewis, R. Jia and S. Min (Editors), *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pp. 42–50, Punta Cana, Dominican Republic, 2021.

36. Cui, Y., T. Liu, W. Che, L. Xiao, Z. Chen, W. Ma, S. Wang and G. Hu, "A Span-Extraction Dataset for Chinese Machine Reading Comprehension", K. Inui, J. Jiang, V. Ng and X. Wan (Editors), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 5883–5889, Hong Kong, China, 2019.

37. Mozannar, H., E. Maamary, K. El Hajal and H. Hajj, "Neural Arabic Question Answering", W. El-Hajj, L. H. Belguith, F. Bougares, W. Magdy, I. Zitouni, N. Tomeh, M. El-Haj and W. Zaghouani (Editors), *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 108–118, Florence, Italy, 2019.

38. Carrino, C. P., M. R. Costa-jussà and J. A. R. Fonollosa, "Automatic Spanish Translation of SQuAD Dataset for Multi-lingual Question Answering", N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk and S. Piperidis (Editors), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 5515–5523, Marseille, France, 2020.

39. Abadani, N., J. Mozafari, A. Fatemi, M. Nematbakhsh and A. Kazemi, "ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0", *International Journal of Web Research*, Vol. 4, No. 1, pp. 34–46, 2021.

40. Lee, C.-H., S.-M. Wang, H.-C. Chang and H.-Y. Lee, "ODSQA: Open-Domain Spoken Question Answering Dataset", *2018 Institute of Electrical and Electronics Engineers Spoken Language Technology Workshop*, pp. 949–956, 2018.

41. Shao, C., T. Liu, Y. Lai, Y. Tseng and S. Tsai, "DRCD: A Chinese Machine Reading Comprehension Dataset", *CoRR*, Vol. abs/1806.00920, 2018.

42. Lin, G.-T., Y.-S. Chuang, H.-L. Chung, S. wen Yang, H.-J. Chen, S. A. Dong, S.-W. Li, A. Mohamed, H. yi Lee and L. shan Lee, "DUAL: Discrete Spoken Unit Adaptive Learning for Textless Spoken Question Answering", *Proceedings Interspeech 2022*, pp. 5165–5169, 2022.

43. You, C., N. Chen, F. Liu, S. Ge, X. Wu and Y. Zou, "End-to-end Spoken Conversational Question Answering: Task, Dataset and Model", M. Carpuat, M.-C. de Marneffe and I. V. Meza Ruiz (Editors), *Findings of the Association for Computational Linguistics: North American Association for Computational Linguistics 2022*, pp. 1219–1232, Seattle, United States, 2022.

44. Turker, M., E. Ari and A. Han, "VNLP: Turkish NLP Package", *arXiv preprint arXiv:2403.01309*, 2024.

45. Tron, N., "Quantulum3: Extract Quantities from Text.", `https://github.com/nielstron/quantulum3`, 2024, accessed: April 4, 2024.

46. van den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio", *CoRR*, Vol. abs/1609.03499, 2016.

47. Babu, A., C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale", *arXiv*, Vol. abs/2111.09296, 2021.

48. Radford, A., J. W. Kim, T. Xu, G. Brockman, C. McLeavey and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision", *arXiv preprint arXiv:2212.04356*, 2022.

49. Bain, M., J. Huh, T. Han and A. Zisserman, "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio", *INTERSPEECH 2023*, 2023.

50. Baevski, A., Y. Zhou, A. Mohamed and M. Auli, "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations", *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 33, pp. 12449–12460, 2020.

51. Schweter, S., "BERTurk-BERT Models for Turkish", `https://doi.org/10.5281/zenodo.3770924`, 2020, accessed: April 10, 2024.