DOMAIN-AWARE CONVERSATIONAL OPEN QUESTION ANSWERING FOR RESOURCE-CONSTRAINED LANGUAGES

by

Emrah Budur

B.S., Computer Engineering, Yıldız Technical University, 2007

M.S., Computer Engineering, Boğaziçi University, 2011

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

Graduate Program in Computer Engineering

Boğaziçi University

2024

To my parents, Ahmet Budur and Miyese Budur...

ACKNOWLEDGEMENTS

PhD is a long journey, full of growth, not only in terms of academical knowledge but also in many other way including but not limited to patience, self-sacrifice, delay of gratification, becoming a life-long learner etc. At the end of this journey, I see I had two fallacies that the life profoundly proved me wrong down the road: 1) I thought this journey will help me achieve self-completion but actually I ended up with believing that there may be no such thing in life. Even if you tend to complete part of yourself, you may end up with loosing another part. 2) Due to many unexpected interruptions down the road, I could never believe this journey would come to an end at some point.

As I reflect on this journey, I realize that what truly shaped me was not just the challenges or milestones, but the presence of those who believed in me along the way. They say, "It's about the journey, not the destination", often attributed to Ralph Waldo Emerson, a 19th-century philosopher. But actually, sometimes it's neither the journey nor the destination that changes you, but the "company" along the way. In this sense, I am most grateful to Prof. Tunga Güngör, who accompanied throughout this journey with constant support. As the Turkish proverb goes: "No road is long with good company", Prof. Güngör's belief in me and his steadfast support were like the Sun—the ever-present companion shining unconditionally on all living beings despite the challenges they face throughout the year.[1] His guidance lifted me up gracefully whenever I had setbacks, ultimately helping me reach the end of this journey.

This journey offered a great privilege for me to meet also other exceptional researchers and world renowned professors who had foundational role in their fields. Among them, I am most indebted to Prof. Christopher Potts, who not only welcomed me into the Stanford NLP Lab, but also guided me in this journey by generously sharing his time, perspective, experience and vision, even though he had numerous other students who might have honored his time in the best way possible. His vision has been, and will always be, a guide for me in many ways just like the the North-Star.[2]

[1] https://bit.ly/sun-and-frozen-tree
[2] https://bit.ly/step-in-the-right-direction

In every challenge in this journey, one of the most effective sources of motivation was honoring the time of our very own thesis progress committee who shared their time and feedback generously in every thesis progress presentation. I am immensely thankful to Prof. Arzucan Özgür and Assist. Prof. Ebru Arısoy-Saraçlar who candidly[3] shared their feedbacks to improve our thesis—much like ground truth labels guiding a neural model in its quest to find the right direction. Likewise, I am grateful also to all members of our thesis committee, Assist. Prof. Suzan Üsküdarlı, Prof. Deniz Yüret, Prof. Zehra Çataltepe, Prof. Ethem Alpaydın, who all had seminal contributions in different phases of our thesis.

Publication is an essential requirement of this journey. In this part, I was fortunate to collaborate with outstanding researchers and friends. They served not just as dilligent co-authors, but also as unique companions. In this sense, I would like to share my gratitudes to my beloved friends and co-authors Rıza Özçelik, Dilara Soylu and Omar Khattab, who readily dedicated their time and efforts whenever, and however needed. I wish them all a great career, however they choose to pursue it throughout their lives.

This journey requires time and effort involving not just the students themselves but also their families and close ones. I can undoubtly express that this journey would never have come to an end for me, if I had lacked family support. Thus, I extend my gratitude to all my family, especially my son, Ahmet Emre Budur (aged 10). Passing on an inspirational narrative that can steer him through his life has significantly fueled my motivation during this journey. I hope everyone enjoys a safe and healthy life, all with their loved ones.

This thesis benefited greatly from the support of several institutions. I am deeply grateful to The Scientific and Technological Research Council of Turkey (TÜBİTAK) and the Council of Higher Education (YÖK) for funding my work through the BİDEB 2214/A program and the 100/2000 Graduate Scholarship Program, respectively. Computational experiments were carried out primarily using the Boğaziçi TETAM DGX-1 GPU Cluster and partly with resources provided by TÜBİTAK ULAKBİM High Performance and Grid Computing Center (TRUBA), the Stanford Research Computing Center (FarmShare), and the

[3] https://bit.ly/adam-grant-tweet-nice-vs-kind

Stanford NLP clusters. Dataset generation was made possible through cloud credits awarded in 2019 by the AWS Cloud Credits for Research Program (formerly AWS Research Grants).[4] These resources were instrumental in facilitating various phases of our thesis, and I am deeply grateful to everyone who made them accessible to early-career researchers like me.

PhD is often seen as an individual journey, but in reality, it is a team effort that extends far beyond the boundaries of the university. The "team" in this context includes not only our amazing university professors, advisors, co-authors and friends, but also the countless individuals who support and uplift us in everyday life. With this perspective, I am deeply grateful to my friends, colleagues, teammates, managers, mentors, neighbors, relatives, and everyone who directly or indirectly contributed to my university life. Their encouragement, understanding, and kindness made even the most challenging moments manageable. Whether it was through a thoughtful conversation, shared laughter, a simple gesture of support, or even a digital interaction, their presence added immeasurable value to this journey, reminding me that no accomplishment in life is truly solitary.

As Steve Jobs ends his commencement speech succinctly as "Stay hungry, stay foolish",[5] I want to end the forewords of my thesis by extending his words as "Stay hungry, stay foolish—*stay alive*". This extension aims to emphasize the idea that life itself is a never-ending journey filled with hidden challenges and moments of uncertainty, and no setback should ever take precedence over the value of life itself. [6] In this sense, Anton Chekhov's reflection resonates deeply on this point: "I may not have amazing victories, but I can amaze you with the defeats that I came out of alive". With this perspective, I extend my heartfelt gratitudes to each and everyone who, directly or indirectly, contributed to helping me navigate and survive in this part of my life journey. And, I hope that everyone embarking on a similar path to always end up with a rewarding and memorable experience, which may be as simple as: "*...being alive in the messiest, richest and most whole sense*".[7]

---

[4]We thank Alara Dirik, Almira Bağlar, Berfu Büyüköz, Berna Erden, Gökçe Uludoğan, Havva Yüksel, Melih Barsbey, Murat Karademir, Selen Parlar, Tuğçe Ulutuğ, Utku Yavuz for their support on our grant application.

[5]`https://stanford.io/4183NBC`

[6]`https://bit.ly/dogan-cuceloglu-phd-experience`

[7]`https://bit.ly/ted-talk-what-almost-dying-taught-me-about-living`

ABSTRACT

DOMAIN-AWARE CONVERSATIONAL OPEN QUESTION ANSWERING FOR RESOURCE-CONSTRAINED LANGUAGES

The introduction of many modern conversational applications into the market not only increased the users' demand on information seeking through conversational interactions but also boosted the research-intensive innovations in the field of Conversational AI. Despite significant advancements in the field, the progress is mainly limited to the English language. This limitation is primarily attributed to the advanced models requiring large amounts of labeled datasets, hence, leaving languages with limited resources behind the scene. In this thesis, we aim to address challenges for building a robust Conversational OpenQA systems in resource-constrained languages like Turkish, and bridge the gap between a Conversational Open QA System in English and resource-constrained languages, by following a three-step systematic approach.

As the first step, we propose a cost-effective method to generate synthetic datasets for resource-constrained languages by using neural machine translation (NMT) tools to translate English Natural Language Inference (NLI) datasets into Turkish. Extending this approach to Question Answering (QA) domain in the second step, we show that OpenQA is feasible for low-resource languages, even with only a few hundred labeled examples. Finally, by incorporating conversational signals as well as an Out-of-Domain detector enhanced by an NLI model, we develop BIRI, a Conversational OpenQA system, improving performance and aligning better with real-world use cases.

ÖZET

KAYNAK KISITI OLAN DİLLER İÇİN ALAN FARKINDALIĞINA SAHİP SOHBETSEL SERBEST SORU-CEVAPLAMA

Çok sayıda modern konuşma uygulamasının pazara girişi sadece kullanıcıların konuşmalı etkileşimler aracılığıyla bilgi arama talebini artırmakla kalmamış aynı zamanda Yapay Zeka destekli Sohbet Uygulamaları alanındaki araştırmalardaki yeniliklere de ivme kazandırmıştır. Alanında önemli ilerlemelere rağmen, bu ilerlemeler genellikle İngilizce dilinde sınırlı kalmıştır. Bu kısıtlama, gelişmiş modellerin büyük miktarda etiketlenmiş veri kümesine ihtiyaç duymasından kaynaklanmaktadır ve bu da sınırlı kaynaklara sahip dilleri göz ardı etmektedir. Bu tezde üç adımlı sistemli bir yaklaşımı takip ederek, Türkçe gibi kaynak kısıtlı dillerde güçlü Yapay Zeka destekli SSC sistemleri oluşturmanın zorluklarına değinmeyi, Yapay Zeka destekli Serbest Soru Cevap Sistemler (SSC) konusunda İngilizce ve kaynak kısıtlı diller arasındaki mesafeyi kapatmayı amaçlıyoruz.

İlk adım olarak, İngilizce Doğal Dil Çıkarımı (DDÇ) veri kümelerini Türkçe'ye çevirmek için sinirsel makine çeviri (SMÇ) araçlarını kullanarak kaynak kısıtlı diller için sentetik veri kümeleri oluşturmak amacıyla uygun maliyetli bir yöntem öneriyoruz. İkinci adımda bu yaklaşımı Soru Cevaplama (QA) alanına genişleterek, sadece birkaç yüz etiketli örnekle bile kaynak kısıtlı diller için Serbest Soru Cevaplamanın (SSC) mümkün olduğunu gösteriyoruz. Son olarak, karşılıklı konuşma sinyallerini ve DDÇ modeli ile güçlendirilmiş bir Alan Dışı (OOD) algılayıcıyı da dahil ederek, gerçek dünya kullanım senaryolarıyla daha iyi uyum sağlayan bir Yapay Zeka destekli SSC sistemi olan BİRİ'yi geliştiriyoruz.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv
ABSTRACT vii
ÖZET viii
LIST OF FIGURES xiii
LIST OF TABLES xv
LIST OF ACRONYMS/ABBREVIATIONS xxi
1. INTRODUCTION 1
1.1. Contributions 3
1.2. Publications 4
1.3. Thesis Outline 4
2. RELATED WORK 6
2.1. Natural Language Inference 6
2.2. Natural Language Processing in Turkish 7
2.3. English Question Answering 7
2.4. Multilingual Question Answering 9
2.5. Open Domain Question Answering 11
2.5.1. Conversational OpenQA 15
2.5.2. Conversational OpenQA Datasets 17
3. NLI-TR: DATA AND REPRESENTATION FOR TURKISH NATURAL LANGUAGE INFERENCE 19
3.1. Introduction 19
3.2. Creating and Validating NLI-TR 21
3.2.1. English NLI Datasets 21
3.2.2. Automatic Translation Effort 21
3.2.3. Translation Quality Assurance 23
3.3. Experiments 26
3.3.1. Case Study I: Comparing BERT models on Turkish NLI Datasets . . 26
3.3.1.1. Experimental Settings 27

3.3.1.2. Results . . . 27
3.3.2. Case Study II: Comparing Morphological Parsers on Turkish NLI Datasets . . . 28
3.3.2.1. Experimental Settings . . . 28
3.3.2.2. Results . . . 30
3.3.3. Case Study III: Evaluating NLI-TR on Human-Translated Sentences 31
3.3.3.1. Datasets . . . 32
3.3.3.2. Models . . . 32
3.3.3.3. Results . . . 32
3.4. Summary . . . 33
4. SQuAD-TR: BUILDING EFFICIENT AND EFFECTIVE OPENQA SYSTEMS FOR LOW-RESOURCE LANGUAGES . . . 35
4.1. Introduction . . . 35
4.2. Datasets . . . 38
4.2.1. SQuAD-TR . . . 38
4.2.2. Knowledge Source . . . 40
4.2.3. Models . . . 43
4.2.3.1. Retriever Models . . . 43
4.2.3.2. Reader Models . . . 46
4.2.4. Standard QA and OpenQA Formulations . . . 48
4.2.4.1. Standard QA Formulation . . . 48
4.2.4.2. OpenQA Formulation . . . 49
4.3. Experimental Results . . . 55
4.3.1. Standard QA Results . . . 55
4.3.2. Open QA Results . . . 56
4.3.2.1. Retriever Results . . . 57
4.3.2.2. Reader Results . . . 72
4.3.2.3. OpenQA Results with Subsampled Test Sets . . . 74
4.3.3. Resource Requirements of the Models . . . 75
4.3.3.1. Resource Usage of Retriever Models . . . 75
4.3.3.2. Resource Usage of Reader Models . . . 77

4.4. Discussion . . . 78
4.5. Adaptation and Generalization: Extending Our Method to Diverse Language Contexts . . . 80
4.6. Summary . . . 84
5. BİRİ: CONVERSATIONAL OPEN QUESTION ANSWERING FOR RESOURCE-CONSTRAINED LANGUAGES . . . 86
5.1. Introduction . . . 86
5.2. Methodology . . . 89
5.2.1. Datasets . . . 89
5.2.2. Conversational OpenQA System . . . 91
5.2.2.1. Task Formulation . . . 92
5.2.2.2. Models . . . 92
5.2.3. Evaluation Metrics . . . 95
5.3. Experiment Results . . . 96
5.3.1. Retriever Results and Analysis . . . 96
5.3.2. Reader Results and Analysis . . . 99
5.4. Generalization to Other Non-English Languages . . . 102
5.5. Summary . . . 102
6. Q: WHAT IS THE ULTIMATE WISDOM OF A QA SYSTEM? - A: I DO NOT KNOW. . . . 104
6.1. Methods . . . 104
6.2. Experiment Results . . . 107
6.3. Summary . . . 107
7. DISCUSSION . . . 109
7.1. Key Findings . . . 109
7.2. Limitations . . . 110
7.3. Machine Translation as a Data Generation Tool . . . 111
7.4. Recent Data Generation Approaches . . . 112
7.5. Implications of the Work . . . 114
7.6. Future Work . . . 115
8. CONCLUSION . . . 118

REFERENCES 119
APPENDIX A: ADDITIONAL EXPERIMENTS WITH NLI-TR 148
APPENDIX B: ALGORITHM AND FLOWCHART FOR BIRI 153
APPENDIX C: DETAILED EXAMPLES FOR BIRI 155

LIST OF FIGURES

Figure 1.1. Different types of question answering systems. (a) Standard QA vs Conversational QA. (b) OpenQA vs Conversational OpenQA. 2

Figure 3.1. Development set accuracy for the three morphological parsers and a model without morphological parsing. The x-axis tracks the size of the training set. We find that morphological parsing is generally helpful in early rounds, when the training set is very small, but that its importance diminishes as the training set increases. These effects are especially clear for the two MultiNLI-TR dev sets. 30

Figure 4.1. System overview diagram for the OpenQA retriever component. We assume the knowledge source is independent of the system for the sake of clarity in the figure. 51

Figure 4.2. Overview diagram for the OpenQA reader component. For the sake of clarity in the figure, we assume the knowledge source that retrievers are based on is the same across all systems. 54

Figure 4.3. Retriever results for the OpenQA formulation for different sample portions of `XQuAD-TR` based on `Wiki-TR-2023`. 74

Figure 4.4. Reader results for the OpenQA formulation for different sample portions of `XQuAD-TR` based on `Wiki-TR-2023`. 74

Figure 5.1. Showing an XQuAD entry in a conversational context. (a) The example in English. (b) The example in Turkish. 88

Figure 5.2. Overview of the Conversational OpenQA system with the numbers showing the information flow 91

Figure 5.3. Success metrics for the retrievers evaluated on XQuAD-TR, highlighting the contribution of contextual signals to the overall performance of the retrievers. The left segments of each figure, separated by a dotted line, show the retrievers without using conversational signals. The right segments of each figure show the retrievers where conversational signals are used. a) Success@k metrics. b) Count@k metrics. 98

Figure 5.4. Success metrics for OpenQA systems evaluated on XQuAD-TR, highlighting the contribution of contextual signals to the overall performance. The left segments of each figure show systems without conversational signals, while the right segments show systems using these signals in retrievers or both retrievers and readers. (a) Success metrics for BM25-based OpenQA systems. (b) Success metrics for ColBERT-QA-based OpenQA systems. 101

Figure 6.1. Out-of-Domain (OOD) Detector System Overview. (a) High-Level Overview. (b) Low-Level Overview. 105

Figure A.1. Confusion matrix for the majority-level label consistency results of all annotations in Table 3.3. The label “broken” corresponds to the pairs which have either major translation error or no majority-level label. . . 150

Figure A.2. XNLI-Dev and XNLI-Test accuracy of three transformer models trained on MultiNLI-TR and MultiNLI-TR$^{\text{XNLI}}$. 151

Figure B.1. The flowchart of the Conversational OpenQA system. 154

LIST OF TABLES

Table 2.1. QA datasets for non-English languages. 10

Table 3.1. Sample translations from SNLI into NLI-TR. Each premise is associated with a hypothesis from each of the three NLI categories. Table A.1 in our supplementary materials provides MultiNLI examples. 22

Table 3.2. Comparative statistics for the English and Turkish NLI datasets. The Turkish translations have larger vocabularies and lower token counts due to the highly agglutinating morphology of Turkish as compared to English. 23

Table 3.3. Translation quality and label consistency of the translations in SNLI-TR and MultiNLI-TR based on expert judgements. For the quality ratings (1–5), we report mean and standard deviation (in parentheses). For label consistency, we report the percentage of labels in SNLI-TR and MultiNLI-TR judged consistent with the original label, both in annotation- and sentence-level. 24

Table 3.4. Accuracy results for the publicly available cased BERT models on NLI-TR. BERTurk performed the best in all three evaluations, highlighting the value of language-specific resources for NLI. 28

Table 3.5. Accuracy results for different morphology approaches on NLI-TR. To facilitate running many experiments, these results are for pretraining on just one-tenth of the Turkish corpus used by BERTurk and fine-tuning on NLI-TR for just one epoch. 31

Table 3.6. Accuracy results comparing NLI-TR with another machine translated dataset. NLI-TR performed better, but the gap is modest, suggesting that both datasets have value for Turkish NLI. Figure A.2 in our supplementary materials provides full learning curves. The results are very similar to those of Table 3.4 for MultiNLI, in overall quality and in the ranking of models. 33

Table 4.1. Examples for the answer spans that are recovered in SQuAD-TR-TRAIN. 39

Table 4.2. The basic statistics for the training and dev splits of SQuAD-EN and SQuAD-TR datasets along with the XQuAD-TR dataset. 40

Table 4.3. Basic statistics for the knowledge sources used in our study and the one used in the DPR model of Karpukhin et al. [1]. 42

Table 4.4. Reader results for the standard formulation of QA task tested on XQuAD-TR. 56

Table 4.5. Retriever results for the OpenQA formulation of QA task evaluated on "XQuAD-TR". All tokenizers are uncased. The highest values in each column are shown in **bold**. For equal pairs, the larger ones on more significant digits are underlined. 57

Table 4.6. Retriever results for the OpenQA formulation of QA task evaluated on "SQuAD-TR-TRAIN". All tokenizers are uncased. The highest values in each column are shown in **bold**. For equal pairs, the larger ones on more significant digits are underlined. 58

Table 4.7. The MRR scores of the retriever models evaluated on the ranks of the same relevant passages unanimously returned from all retrievers as the highest-ranked positive passages. 60

Table 4.8. The "negative" effect of the TF-IDF approach used in BM25 are exemplified in Question 165. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results. 62

Table 4.9. The "positive" effect of the TF-IDF approach used in BM25 are exemplified in Question 484, respectively. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results. 63

Table 4.10. An example showcasing how ColBERT-QA can better capture the information needs implicit in questions. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. 64

Table 4.11. An example showcasing how ColBERT-QA can better capture the information needs implicit in questions. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. 65

Table 4.12. An example that illustrates how WordPiece tokenization can produce a "favorable" outcome, depending on its ability to resist the influence of lexical bias. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results. 66

Table 4.13. An example that illustrates how WordPiece tokenization can produce an "unfavorable" outcome, depending on its ability to resist the influence of lexical bias. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results. 67

Table 4.14. Example passages chosen by each retriever for a given question from SQuAD-TR-TRAIN, and put into the training set of the corresponding readers. 69

Table 4.15. Reader results for the OpenQA formulation of QA task with respect to each of the retriever. 72

Table 4.16. Error analysis on the reader outputs on XQuAD-TR with respect to the source of the passages (SQuAD-TR or Wiki-TR) and the retriever module the reader is based on. 73

Table 4.17. Resource consumption of retriever models during training, indexing, and retrieval steps. 76

Table 4.18. Resource consumption of reader models during training and test steps categorized by the architecture of the model. 77

Table 4.19. Our general method for creating OpenQA systems in low-resource languages efficiently. 81

Table 5.1. A sample entry in the XQuAD dataset where A_k denotes the answer span for the Question k in the corresponding language. In this example, the first question disambiguates the context of the term "packet" by adding content words like "connectionless mode". However, subsequent questions lack these contextual words, posing a challenge in answering those questions without the paragraph given as input. 87

Table 5.2. Statistics for the SQuAD-TR-TRAIN [2] and XQuAD-TR [3] datasets. . . 90

Table 5.3. Retriever results with/without historical questions used for training. The bolded figures indicate the highest values within their respective columns. The results are evaluated on XQuAD-TR. 97

Table 5.4. Reader results with/without conversation history used for inference. The results are evaluated on the respective XQuAD-TR versions of the corresponding retriever model. 99

Table 6.1. Standard QA reader results with/without OOD detector, trained on SQuAD-TR-TRAIN evaluated on SQuAD-TR-DEV. 107

Table A.1. Sample translations from SNLI and MultiNLI into NLI-TR. Each premise is associated with a hypothesis from each of the three NLI categories. 148

Table A.2. Accuracy of the cased models in Table 3.4 trained on the original SNLI and MultiNLI datasets in English. We used the same fine-tuning and evaluation procedures. BERT-En ranked the first and BERT-Multi ranked the second, emphasizing the importance of in-language training one-more time as in Section 3.3.1. 149

Table A.3. Accuracy results of the models in Table 3.6 for machine translated XNLI. 150

Table C.1. An example emphasizing the positive effect of using conversational history in the retrieval process for locating the relevant paragraph to respond to a query. 156

Table C.2. An example demonstrating the limitations of using conversational history in the retrieval process for locating a relevant passage for a question. . 158

Table C.3. An example highlighting the positive impact of incorporating conversational history (previous questions) in the reader to find the answer based on a given question and the retrieved passage. 159

Table C.4. An example where incorporating conversational history causes the reader to select an incorrect answer span, while the reader without conversational signals accurately identifies the correct answer. 160

LIST OF ACRONYMS/ABBREVIATIONS

BM25	Best Match 25
ColBERT	Contextual Late Interaction over BERT
ConvBERT	Conversational BERT
ConvBM25	Conversational BM25
ConvColBERT	Conversational ColBERT
ConvOpenQA	Conversational Open Question Answering
DPR	Dense Passage Retriever
MultiNLI	Multi-Genre Natural Language Inference
MT	Machine Translation
NLI	Natural Language Understanding
NLP	Natural Language Processing
NLU	Natural Language Understanding
SNLI	Stanford Natural Language Inference
SQuAD	Stanford Question Answering Dataset
OpenQA	Open Question Answering
QA	Question Answering

1. INTRODUCTION

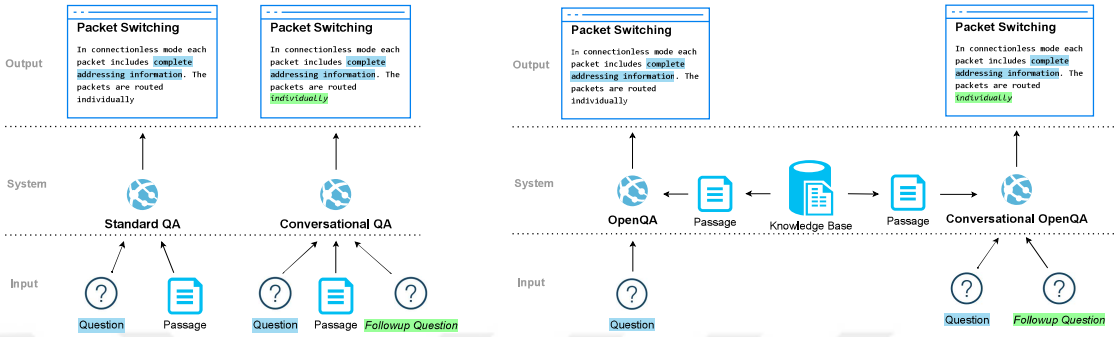
Over the past few years, there has been an increased interest on Conversational AI, driven by demanding user expectations, and the recent breakthroughs of neural models, notably propelled by the Transformer architecture [4]. Conversational AI is akin to a task-oriented dialogue system [5], with its primary goal being to satisfy the user's information need through conversational interfaces. Among many types of conversational systems [6], Question Answering (QA) has gained extensive usage in fulfilling the information-seeking needs of the users [7].

Question answering (QA) is the task of answering questions posed in natural language with free-form natural language answers. In its standard formulation, QA is posed in a highly constrained way. The system is given a passage and a question with a guarantee that the answer can be found in the passage [8, 9, 10]. The main component of standard QA systems is a "reader", which takes a passage and a question as input and returns an answer. Present day systems are extremely successful at such tasks, often surpassing human performance [11]. However, they are of limited use, since real-world question answering scenarios mostly do not involve gold passages or provide answerability guarantees.

To address this limitation, Open Question Answering (OpenQA) systems have been introduced, where only the question text is given as input without any passage. Related passages are retrieved from a large corpus by a "retriever" and then used by the reader to predict an answer. In this setting, there is no guarantee that the retrieved passages will contain the answer, and the success of the system thus depends on having a successful retriever module to provide appropriate passages to the reader.

Nevertheless, these systems operate on single-turn interactions, while users commonly engage with QA systems through follow-up interactions to fulfill their information needs. To address this, Conversational QA and Conversational OpenQA systems have been emerged leveraging conversation history to enhance the QA system's performance and cater to users'

information needs more effectively. Figure 1.1 illustrates the distinctions among different types of QA systems.



(a) (b)

Figure 1.1. Different types of question answering systems. (a) Standard QA vs Conversational QA. (b) OpenQA vs Conversational OpenQA.

Recent years have seen rapid improvements in such QA systems stemming from the use of neural retriever modules that can provide semantically rich representations of documents. We are approaching the point where OpenQA systems will be as effective as standard QA systems [12]. However, this rapid progress in all types of QA systems is largely confined to English. Progress in other languages is constrained by a scarcity of gold data. While there are some high-quality multilingual resources in this domain [3, 13], the amount and diversity of such data remain low. The cost of creating new datasets is the main obstacle to progress in this area.

In this thesis, we introduce a step-by-step approach to remove the frictions on the progress of Conversational Open QA system for the languages with limited resources. As a specific case study, we implement our system for Turkish, a language that exhibits significant typological and grammatical differences from English, thereby reflecting various distinctions present in other languages as well. To address data limitations in a cost-effective manner, we use machine translation to translate widely available English datasets into Turkish. As a result, we create new research opportunities for Conversational Open QA systems in Turkish, and hope to inspire fellow researchers to adopt a similar approach to make Conversational AI accessible also for other languages.

1.1. Contributions

In this thesis, our contributions can be summarized as follows.

- We offer a cost-effective methodology to remedy the data scarcity issue in resource-contrained languages by leveraging neural machine translation approach.
  - We created and published the first Turkish Natural Language Inference dataset, NLI-TR, through the machine translation of SNLI and MultiNLI.
  - We verified the quality of the translated dataset by means of human evaluation.
  - We utilized NLI-TR to explore key aspects of Turkish Natural Language Inference (NLI). Initially, we employed NLI-TR to examine the impacts of pretraining exclusively in Turkish resources. Our findings indicate that a pretraining approach focused solely on Turkish resources can notably improve Turkish models. Moreover, we observed that investing in morphological parsing is likely justified only when dealing with a limited training dataset.
- We demonstrated that obtaining a robust OpenQA system is feasible also for low-resource languages.
  - We translated the mainstream QA dataset, SQuAD2.0 into Turkish to obtain `SQuAD-TR`, a large scale Turkish QA dataset. We verified its utility using human translated dataset, `XQuAD-TR`.
  - We demonstrated that it is possible to obtain OpenQA system also for low-resource languages when we apply the power of weak supervision in OpenQA setting.
  - We also proved that only a few hundred labeled ground-truth examples may be enough to confidently assess the performance of OpenQA systems.
- We showed that obtaining a conversational OpenQA system is possible also for the languages with limited resources.
  - We repurposed `SQuAD-TR` and `XQuAD-TR` and used them as if they were created in conversational setting.
  - We proved that incorporating conversational signals into different components of the system helps improve the performance of OpenQA systems also in resource-constrained setting.

1.2. Publications

The content of this thesis including the images is covered by the following publications in accordance with the publisher's policies and licensing terms.

- Budur, Emrah. Özçelik, Rıza. Güngör, Tunga and Potts, Christopher. "Data and Representation for Turkish Natural Language Inference", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp.8253–8267, Online, 2020.
- Budur, Emrah. Özçelik, Rıza. Soylu, Dilara. Khattab, Omar. Güngör, Tunga and Potts, Christopher. "Building Efficient and Effective OpenQA Systems for Low-Resource Languages", *Knowledge-Based Systems*, Vol. 302, p. 112243, 2024.
- Budur, Emrah and Güngör, Tunga. "Conversational Open Question Answering for Resource-Constrained Languages". *Turkish Journal of Electrical Engineering and Computer Sciences*, 2024.

The visuals prepared as part of this thesis and transferred to the respective publishers have been used in the thesis in accordance with the publishers' publication policies on "reuse of text and graphics for authors' dissertations" as stated on the publishers' official websites.

1.3. Thesis Outline

Chapter 2 provides the essential background information and relevant prior work forming the foundation of our thesis. In Chapter 3, we address the question: "Can we extend the accessibility of mainstream datasets widely used in English to low-resource languages through machine translation?" Our results affirmatively answer this question. We detail our methodology for translating mainstream Natural Language Inference (NLI) datasets into Turkish, illustrating how machine translation helps overcome data limitations in low-resource languages. We selected the NLI task as a representative task for our thesis, given its focus on sentence-level labeling, enabling us to concentrate on the quality of machine translation without requiring sequence-level labeling. Chapter 4 expands our framework to

derive a large-scale Question Answering dataset for Turkish based on SQuAD2.0, a widely used dataset for question answering in English. We demonstrate the feasibility of Open Question Answering for low-resource languages through the weak-supervision approach that is resilient against the inherent data quality issues in machine translation. In Chapter 5, we present our core solution providing a Conversational Open Question Answering system for Turkish, leveraging the datasets and resources acquired in the preceeding chapters within a conversational context. Chapter 6 delves into the philosophical question of whether a QA system can truly say "I don't know" exploring the limitations and inherent uncertainties in such systems, especially when faced with unanswerable questions. Chapter 7 provides a detailed discussion of the key findings, limitations, implications of our work, insights into recent data generation techniques along with potential future directions to further improve Conversational OpenQA systems in low-resource languages. Finally, Chapter 8 concludes the thesis by summarizing the contributions.

2. RELATED WORK

2.1. Natural Language Inference

Early in the development of textual entailment tasks, Mehdad et al. [14] argued for multilingual versions of them. This led to subsequent explorations of a variety of techniques, including crowdsourcing translations [15, 16], relying on parallel corpora to support reasoning across languages [17], and automatically translating datasets using MT systems [14, 18, 19]. This research informed SemEval tasks in 2012 [20] and 2013 [21] followed by ASSIN 1 [22] and 2 [23] shared tasks exploring the viability of multilingual NLI. From the perspective of present-day NLI models, these datasets are very small, but they could be used productively as challenge problems.

More recently, Conneau et al. [24] reinvigorated work on multilingual NLI with their XNLI dataset. XNLI provides expert-translated evaluation sets from English into 14 other languages, including Turkish. Though they are valuable resources to push NLI research beyond English, test sets alone are insufficient for in-language training on target languages, which is likely to lower the performance of the resulting systems.

Although it was not the main focus of the XNLI effort, Conneau et al. [24] distributed machine translations of MultiNLI into other languages, including Turkish, which we call MultiNLI-TR$^{\text{XNLI}}$ in this thesis. The translations helped them form a strong baseline for their cross-lingual models, which proved superior in their assessments. However, the quality of the translations remains crucial, as the authors note.

XNLI's primary focus on test sets rather than training is justified by a wide body of recent results on cross-lingual transfer learning. Multilingual embeddings (embeddings trained on multilingual corpora) have played an important role in these developments. The BERT team [25] released multilingual embeddings and demonstrated their value using XNLI. At the same time, BERT models have been released for a variety of individual languages [26]

and specialized domains [27, 28]. While we might expect the language- and domain-specific embeddings to be superior for the kind of data they were trained on, the multilingual versions might be more efficient in large-scale deployments in diverse environments. Balancing these trade-offs is challenging. Here, we offer some insight into these trade-offs for Turkish in our thesis.

2.2. Natural Language Processing in Turkish

Turkish is a morphologically-rich language in which new word forms are freely created using suffixation. Several morphological parsers [29, 30, 31] and morphological disambiguation systems [29, 32] have been developed for Turkish. The state-of-the-art morphological analyzers can parse with success rates around 95%. We use three of these parsers in our thesis to evaluate the role of morphology in NLI systems (Section 3.3.2).

Turkish, a member of the Turkic language family [33], exhibits distinct linguistic features that set it apart from other languages like English. Its agglutinative structure forms words by adding suffixes to a base root, allowing complex expressions in single words [34, 35, 36, 37]. Additionally, Turkish employs vowel harmony, a phonological process where vowels within a word harmonize to share certain features [38], a characteristic absent in English. Turkish primarily follows a Subject-Object-Verb (SOV) order, contrasting with English's Subject-Verb-Object (SVO) order [39]. These aspects highlight Turkish's distinct morphology, phonology and syntax, posing special challenges for various NLP tasks [36, 40, 41, 42, 43, 44, 45], thus emphasizing the potential applicability of our system to languages that differ from English in various constraints and linguistic characteristics.

2.3. English Question Answering

There are many QA datasets for English used to address different challenges; see Cambazoglu et al. [46] for a thorough review. One class of QA datasets consists of multiple-choice questions. MCTest [47] is an early dataset built in this style (see also CBT [48]; Booktest [49]). MCTest contains 2640 human-generated questions associated with a correct

answer from a set of candidate answers. The questions and answers are based on 660 short fictional stories at a grade-school level. The fictional nature of the stories limits the use of world knowledge to answer the questions, which is one of the special challenges of this dataset. The main drawback of MCTest is its small size.

SQuAD1.0 [8] was the first major extractive reading comprehension dataset. SQuAD1.0 contains over 100K examples, and each example is a question–passage–answer triple, where annotators selected a span of text from the passage as the answer to the question. SQuAD2.0 [50] is a follow-up that includes over 50K additional examples representing unanswerable questions. The goal here is to encourage the development of systems that detect whether a question is answerable based on the passage given and abstain from answering if necessary [51]. We built SQuAD-TR from SQuAD2.0 to facilitate future research not only on answerable questions but also on unanswerable questions in Turkish.

HotPotQA [52] extends the extractive reading comprehension paradigm to multi-hop questions, i.e., questions whose answers need to be pieced together from information in multiple passages. A closely related task is multi-hop claim verification, as in HoVer [53].

Another class of datasets leverages an existing set of human-generated question–answer pairs, and augments these with supporting passages from external knowledge sources. A prominent example of this type of dataset is TriviaQA [10], which contains 95K question–answer pairs that were prepared by trivia enthusiasts. The question–answer pairs are accompanied by documents retrieved from the Web and Wikipedia. In a similar manner, Dunn et al. [54] built SearchQA by using the Google search engine to retrieve context snippets relevant for question–answer pairs obtained from the Jeopardy! game show archive.[8]

Search engine query logs are also used as a source of examples. WikiQA [55] and Natural Questions [9] are the most commonly used datasets in this class. WikiQA is derived from the 3K most frequent user queries in the query logs of the Bing search engine. Each query is paired with a Wikipedia page clicked by at least five unique users. If a sentence

[8] http://j-archive.com

in the summary part of the associated Wikipedia page includes the answer to the query, the sentence is labeled as “correct”, otherwise “incorrect”. This version of the QA task is referred as “answer-sentence selection”, as it only selects the target sentence answering the question without requiring extraction of the correct answer span from that sentence. WikiQA includes question–page pairs with no “correct” sentences, so the dataset can also be used to build “answer triggering” models, which predict whether the sentences include the answer or not and then select a sentence only if it answers the question.

Like WikiQA, Natural Questions (NQ) relies on queries to a real search engine. NQ contains a total of 320K examples with queries obtained from Google query logs. Each query is associated with a Wikipedia page, which may or may not contain the answer for the query. If the Wikipedia page has the answer, “a long answer” is included in the example to show the passage answering the question. The example may also contain “a short answer” denoting the short form of the target answer. If the example contains neither long nor short answer, then no answer span exists on the page. NQ is a challenging dataset with realistic queries supported by high-quality annotations for the long and short answers. Likewise WikiQA, NQ also provides an opportunity to build answer triggering models with its examples having no long and short answers.

All of these datasets can be re-cast in the OpenQA mould, assuming we can find a large collection of relevant unlabeled documents to be used as a knowledge source. SQuAD1.0, NQ, TriviaQA, HotPotQA have been extensively explored in these terms [12, 56].

2.4. Multilingual Question Answering

Various methods have been used to address the dataset bottleneck for QA in non-English languages [57]. One approach is to curate in-language datasets from scratch. A number of datasets for different languages have been created in this way. We provide a summary in Table 2.1. Datasets created in this manner are likely to be of high quality, but they are expensive and time-consuming to create.

Table 2.1. QA datasets for non-English languages.

Dataset	**Language**	**Number of examples**
KorQuAD [58]	Korean	70,079
FQuAD [59]	French	62,003
SberQuAD [60]	Russian	50,364
CMRC 2018 [61]	Chinese	19,071
GermanQuAD [62]	German	13,722
ARCD [63]	Arabic	1,395

One way to reduce the cost of creating in-language QA systems is to try to rely on the zero-shot transfer capabilities of cross-lingual models. In this approach, multilingual language models (e.g., mBERT [25]; XLM [64]; XLM-RoBERTa [65]) are finetuned on English QA datasets and then used to answer questions in target non-English languages. Multilingual models are efficient and cost effective, especially in large-scale applications requiring multiple language support. However, their performance on the target languages is relatively lower than models that use in-language embeddings [13, 43, 58, 59, 62].

Another cost-effective approach is to rely on machine translation services. In this approach, in-language training datasets are automatically obtained by translating an existing English dataset using machine translation (MT). Previously, SQuAD1.0 [8] was translated into Arabic [63], French [59], and Spanish [66], and SQuAD2.0 [50] was translated into Persian [67]. Similar techniques have also been used in other areas of NLP [43, 68]. For example, Senel et al. [33] recently introduced KardeşNLU[9] using MT systems and Turkish resources in their process to obtain a cost-effective evaluation benchmark dataset for various Natural Language Understanding (NLU) tasks in other Turkic languages, which are often relatively less-resourced than Turkish in several NLP tasks. MT systems can also be effectively applied to extremely low-resource languages, including endangered Indigenous Languages [69, 70, 71, 72, 73].

[9]The term “KardeşNLU” translates to “SiblingNLU” in English.

Using MT systems is undoubtedly productive, but relying on automatic translations for system assessment raises concerns about the validity of those assessments. To the extent that there are systematic errors in the MT output, assessment numbers are likely to be untrustworthy. To address this, Lewis et al. [13] proposed MLQA, which is a multi-way aligned QA dataset to be used for evaluation purposes in 7 languages, with over 5K examples for each language. Similarly, Artetxe et al. [3] developed XQuAD, which consists of a subset of the SQuAD1.0 development dataset with human translations into 10 languages, including Turkish. In our thesis, we rely on the Turkish portion of XQuAD, namely XQuAD-TR, for evaluation as it is the only standard QA evaluation dataset that supports Turkish, currently.

2.5. Open Domain Question Answering

Various methods are employed by researchers to develop OpenQA systems. An investigation of these methods can be found in the comprehensive survey presented by Zhu et al. [74]. Additionally, Zhang et al. [75] provide a thorough analysis of different OpenQA systems, examining them in terms of complexity, efficiency, speed, resource demands, and other relevant factors. Traditionally, OpenQA systems involve two pipelined components: a "retriever" and a "reader". Given a question, the retriever is expected to retrieve candidate passages, and the reader is supposed to extract the target answer span from those retrieved passages.

BM25 was a common choice for the retriever component in the earliest OpenQA systems [76] and it remains in wide use today [77, 78, 79, 80]. BM25 and other retrievers in its class rely on lexical matching. The guiding idea behind more recent neural retrievers is that lexical matching alone is not sufficiently semantic in nature to capture the nuanced ways in which passages can be relevant to user queries. Prominent recent examples of these neural retrievers include ORQA [56], REALM [81], DPR [1], RAG [82], and ColBERT [83]. The leaderboards for OpenQA systems are currently dominated by systems that employ neural retrievers, though BM25 remains a very strong baseline, especially where latency and cost are important additional considerations beyond accuracy-style metrics.

Neural retrievers, despite their advantages, suffer from the drawback of requiring a significant amount of storage space, especially for their indexes. To mitigate this limitation, Yang and Seo [84] proposed a solution that involves several techniques. These techniques include filtering out unnecessary passages prior to the retrieval step, consolidating retriever-reader models with a single encoder, and employing post-training compression.

The English-centric nature of research in this area is arguably holding back retriever development as well. The largest and most widely used dataset in this space is the MS MARCO Passage Ranking dataset [85, 86], and it contains only English texts and queries. However, Bonifacio et al. [87] translated MS MARCO into 13 different languages using automatic translation. The result is mMARCO [87], the first multilingual MS MARCO variant. mMARCO has enabled much new research on multilingual passage retrieval. However, mMARCO does not have any labels within the text to denote answer spans, and so it cannot by itself support the development of multilingual QA systems.

Neural information retrieval (IR) systems can begin from pretrained multilingual embeddings, and this can facilitate multilingual retrieval work. For example, Asai et al. [88] used DPR in the retriever step and proposed a cross-lingual transfer method (XOR QA) to obtain answers for unanswerable questions in the non-English languages from English Wikipedia. In order to do that, (1) they translate the questions in non-English languages into English, (2) find relevant passages and answer spans from English Wikipedia, and (3) translate the English answer spans back to the original language. The leaderboard for their paper, XOR-TyDi [88], includes cross-lingual retrieval and OpenQA tasks.[10] XOR-TyDi has similar motivations to our work, in that it tackles issues around building OpenQA systems in non-English languages effectively, but it differs from our work in substantive ways. To achieve its goals, XOR-TyDi makes the English knowledge source available to non-English languages with the help of a cross-lingual retriever. In contrast, we propose a method to build an in-language retriever that benefits from an existing in-language knowledge source in a non-English language. Both methods are based on translation, but our method benefits from translated data at training time (`TRANSLATE-TRAIN`) even if the translation is noisy, whereas XOR-TyDi requires translations

[10] `https://nlp.cs.washington.edu/xorqa/`

at test time (TRANSLATE-TEST), making the overall system highly vulnerable to translation errors.

For the most part, the reader component in OpenQA systems is an extractive reader: given a retrieved passage and a question, it is trained to extract a substring of the passage corresponding to the answer. Readers of this sort are clearly best aligned with standard QA datasets where the answer is guaranteed to be a substring of the passage provided. In datasets where the answer can be expressed more indirectly, extractive strategies will fail. Extractive readers are also potentially sub-optimal for OpenQA systems, for two reasons: we might be able to retrieve multiple relevant passages, and the passages themselves might indirectly express the answer. In response to these shortcomings, Lewis et al. [82] explore readers that can consume multiple passages and generate original texts in response. Yu et al. [89] introduces KG-FiD, which incorporates knowledge graphs to rerank passages by utilizing Graph Neural Networks (GNN) before the reader generates the response. Nie et al. [90] present a multi-modal approach where the model is guided by heterogeneous knowledge sources and visual cues when generating responses within a conversational context. Lozano et al. [91] adopt the Retrieval-Augmented Generation (RAG) [82] approach to generate answers using a large language model (LLM) for clinical questions based on medical literature in PubMed. Mao et al. [92] compare the performance of both extractive and generative readers in an OpenQA system based on the passages obtained by a Generation-Augmented Retrieval (GAR) step where additional context is generated for the queries to form generation-augmented queries.

As the adoption of LLMs and generative models continues to grow, researchers are increasingly exploring advanced approaches to optimize the use of generative models within OpenQA systems. In this respect, Wei et al. [93] introduces “Chain-of-Thought (CoT) Prompting”, which breaks the question into intermediate reasoning steps to generate a final answer. Yao et al. [94] introduce ReAct using CoT to create prompts that blend reasoning and suitable actions, such as seeking additional information from knowledge sources, for accurate and interpretable answers. Khattab et al. [95] presents Demonstrate-Search-Predict (DSP), a framework that orchestrates the retriever model and a language model generating a series

of intermediary questions helping find multiple relevant passages that answer the question when combined. New toolkits like LangChain [96] and LlamaIndex [97] have emerged to simplify the integration and orchestration of LLMs into multi-stage pipelines and external tools, where LLMs are guided by hand-crafted prompts for specific tasks including RAG, relying on in-context learning [98, 99, 100]. Khattab et al. [101] introduce DSPy, a novel programming model and compiler that can eliminate reliance on hand-crafted prompts in LLM pipelines by automatically generating prompts and LLM invocation strategies based on a declarative program. We leave exploration of generative readers for Turkish in the OpenQA formulation for future work.

Several end-to-end neural models have recently emerged in OpenQA (e.g., SOQAL [63]; DPR [1]; ColBERT-QA [12]; YONO [102]). Early examples predominantly relied on sparse vector representations in the retrieval component. For instance, Mozannar et al. [63] proposed SOQAL as an OpenQA system for Arabic using a hierarchical TF-IDF retriever pipelined with a BERT-based reader [25]. This was followed by an answer ranking component that assigns a score for each answer candidate obtained as a linear combination of the retriever and reader outputs. However, these retrievers, based on sparse representations, struggle to recognize similarity between synonyms and paraphrases that use different lexical terms.

To address the sparse vector representation problem, Karpukhin et al. [1] introduced DPR which is one of the early examples of dense retrievers in the OpenQA domain. DPR utilized dual-encoder architecture to encode dense and latent semantic representations of the questions and the contexts. Given a question, DPR was trained to distinguish the positive passages from the negative passages in the batch. One limitation of DPR was its use of a single vector for the question and the context, resulting in limited interactions between the terms in the two texts. Khattab et al. [12] recently developed ColBERT-QA as a novel end-to-end neural OpenQA model for English, offering more extensive and effective interaction between the question and context terms through a late-interaction mechanism. Alternatively, Lee et al. [102] proposed YONO, a single end-to-end architecture that jointly optimizes the retriever, reranking, and reader components. The fully end-to-end architecture of YONO contributes to its efficiency in terms of model size. However, there is a drawback to combining multiple

components in a single architecture, as each component demonstrates different overfitting characteristics. This vulnerability becomes apparent especially when the training data is limited, which is often the case for low-resource languages.

In our thesis, we focus on two advanced end-to-end neural models used in OpenQA, the DPR and ColBERT-QA models, for their ability to provide dense representations of queries and passages. Each model is explained further in Section 4.2.3 along with the other models we use in our thesis.

2.5.1. Conversational OpenQA

Conversational OpenQA systems must track multi-turn conversational history to effectively address users' information needs in conversational context, alongside the usual challenges of OpenQA systems. Qu et al. [103] introduced ORConvQA as an early instance of a Conversational OpenQA model demonstrating the use of the OR-QuAC dataset. They showcase an end-to-end Conversational OpenQA system that pipelines the retriever, reranker, and reader components, all with conversation history enabled across the board. Unlike other studies [104, 105, 106, 107, 108], they do not use the previous answers to model the conversation history as it is largely impractical to provide answers, especially ground-truth answers, as input in the Conversational OpenQA setting. Coman et al. [109] further advanced this approach by addressing the latency limitations of these pipelined models using a fast reranker in between Dense Passage Retrieval (DPR) [1] and Fusion-in-Decoder (FiD) [110], which are used as the retriever and generative reader models in the Conversational OpenQA setting, respectively. Their reranker helped the retriever to return fewer candidate passages with higher relevancy, allowing the reader to provide better results faster. However, all these studies depend on a fully supervised setting. Yu et al. [111] alleviated the need for a large amount of supervision signals partially by employing a knowledge distillation approach through a teacher-student model. Nevertheless, their model still depends on query rewriting signals. To eliminate reliance on query rewriting signals, Krasakis et al. [112] modified ColBERT [83], known as one of the most effective dense retrievers currently, to contextualize query embeddings with conversational history without the need for query resolution resource.

However, their findings are constrained to the performance of the retriever, as they did not assess whether the modified retriever's output contributes to enhancing the performance of the reader. Moreover, their system relies on gold passages as the supervision signals linked to the training data, posing a significant challenge in scenarios with limited data availability, particularly in low-resource languages. These supervision signals are not always practical, especially in cases involving low-resource languages, given the long-tail characteristics of the required Conversational OpenQA datasets [111, 112]. In our work, we address the limitation of needing supervision signals for resource-contrained languages by utilizing a machine-translated dataset for training, which may contain potentially noisy labels. Our approach remediates the scarcity of high-quality, annotated data in many languages by leveraging existing resources through machine translation. By using machine-translated datasets, we accept the trade-off of dealing with noisier labels in exchange for greater data availability. The use of potentially noisy labels introduces a degree of uncertainty and error into the training process, which can impact the model's performance and accuracy. However, we show that the models can mitigate the potential noise in the dataset, hence, addressing the limitation of the availability of supervision signals cost-effectively. Unlike Krasakis et al. [112], we also incorporate conversational history not only in the retriever but also in the extractive reader, offering a holistic and more reliable perspective on its contribution to OpenQA systems.

Large Language Models (LLMs) have greatly improved the Conversational OpenQA task by providing closed-book zero-shot Conversational QA experience emulating human-like dialogue [113, 114]. This improvement has been boosted by pioneering models like GPTs [99, 115], followed by other models such as PaLM [116] and LLaMA [117] families. More recently, there has been a shift towards in-language LLMs such as TURNA [118] and cosmosGPT [119], which serve as current examples of Turkish LLMs. A major hallmark of LLMs is their support for in-context learning, where a pre-trained language model (LM) is adapted to perform specific tasks by conditioning it on a textual prompt that includes task instructions and a few-shot examples [98, 99, 115].

The outstanding performance of LLMs comes along with intrinsic deficiencies including but not limited to hallucination [120], providing inaccurate [121] and inconsistent [122] infor-

mation, and their limited adaptability and capacity for changes in the knowledge source [123]. For instance, unlike the modular OpenQA system our method is based on, LLMs necessitate comprehensive retraining from scratch for each incremental update or addition in the knowledge source. This process is not only time-consuming but also computationally intensive and ineffective [124]. Consequently, integrating new information or rectifying errors poses considerable challenges, particularly in domains characterized by rapid changes. On the other hand, the approach proposed in this thesis is based on a modular OpenQA system that can adapt itself cost-effectively as the knowledge source gets incremental updates.

Khattab et al. [12] have recently proposed ColBERT-QA incoporating ColBERT [83] as the dense retriever of an end-to-end OpenQA system to have a wholistic view of its task-specific performance. Using weak supervision with only question-answer pairs as gold data, their approach leverages passages retrieved by less powerful retrievers, such as BM25, eliminating the need for gold passages in training. This characteristic paved the way for the development of OpenQA systems for low-resource languages without high-quality gold data. Budur et al. [2] adapted ColBERT-QA for Turkish using machine translation for the training question-answer pairs in the training dataset. While promising for constructing OpenQA systems in low-resource languages, the Turkish adaptation of ColBERT-QA focuses on ad-hoc information seeking questions, overlooking the need for multi-turn Conversational QA. Unlike previous studies, we propose an efficient Conversational OpenQA system tailored to resource-constrained languages like Turkish.

2.5.2. Conversational OpenQA Datasets

To facilitate research in the Conversational QA domain, Reddy et al. [125] introduced the CoQA dataset, comprising 127K question and answer pairs derived in conversational context. Similarly, Choi et al. [104] released QuAC, containing 100K pairs from 14K conversations. However, these datasets assume that the evidence passages are provided as input. Addressing the need for passage extraction, Qu et al. [103] introduced the OR-QuAC dataset, transforming QuAC into an open retrieval setting supported by over 11 million passages from Wikipedia. Utilizing also the context-independent rewrites from CANARD [126], OR-QuAC emerges as

a crucial resource for developing Conversational OpenQA models.

Datasets like CoQA [125], QuAC [104], and OR-QuAC [103] have driven advancements in Conversational QA and OpenQA systems, but their impact is limited to English. Progress in other languages requires innovative approaches to address the challenge of limited data availability in the Conversational QA and OpenQA. While previous efforts created multilingual QA datasets [57, 58, 59, 60, 62, 63], none of them are specifically tailored to the Conversational OpenQA domain for languages with limited resources. Addressing this gap, Budur et al. [2] released SQuAD-TR for training QA systems in Turkish and used XQuAD-TR [3] for evaluation. Although SQuAD-TR is a valuable resource for training QA systems in a low-resource language, it is based on SQuAD2.0 [50] which lacks a conversational setting. In this thesis, we address this constraint cost-effectively by demonstrating the utility of SQuAD-TR in the Conversational OpenQA setting, even though it is derived from machine translation and was not originally designed to support conversational scenarios.

Researchers have recently gathered currently available resources for Turkish NLP [127, 128]. Additionally, Uskudarli et al. [129] introduced TULAP, a platform for Turkish NLP resources. However, none of these resources include a dataset proven effective for Conversational OpenQA in Turkish. In this thesis, we repurpose SQuAD-TR and XQuAD-TR, extending their utility to the conversational setting. This cost-effective approach aims to provide datasets for constructing Conversational OpenQA systems, particularly in low-resource languages. Our approach is distinctive in that it allows for building and improving Conversational OpenQA models using an existing QA dataset obtained using machine translation. Consequently, it enables the development of Conversational OpenQA systems without requiring labor-intensive human-annotated datasets in this specific context, while maintaining the quality and relevance of the data. This method not only mitigates the data scarcity issue in a cost-effective way but it also demonstrates that leveraging machine-translated datasets is a viable solution for low-resource languages, enabling efficient and scalable model development. Furthermore, this approach can be applied to other resource-constrained languages, opening up new opportunities for advancing Conversational OpenQA across diverse linguistic settings.

3. NLI-TR: DATA AND REPRESENTATION FOR TURKISH NATURAL LANGUAGE INFERENCE

3.1. Introduction

Many tasks in natural language processing have been transformed by the introduction of very large annotated datasets. Prominent examples include paraphrase [130], parsing [131], question answering (8), machine translation (MT; 132), and natural language inference (NLI; 133, 134).

Unfortunately, outside of parsing and MT, these datasets tend to be in English. This is not only an obstacle to progress on other languages, but it also limits the field of NLP itself: English is generally not a representative example of the world's languages when it comes to morphology, syntax, or spelling conventions and other kinds of standardization [135], so it's risky to assume that models and results for English will generalize to other languages.

A natural response to these gaps in our dataset coverage might be to launch new annotation efforts for multiple languages. However, this would likely be prohibitively expensive. For example, based on the costs of SNLI [133] and MultiNLI [134], we estimate that each large dataset for NLI would cost upwards of US $50,000 if created completely from scratch.

At the same time, commercial MT systems have improved dramatically in recent years [136, 137, 138, 139]. They now offer high-quality translations between hundreds of language pairs. This raises the question: can we use these MT systems to translate English-language datasets and use the translated versions to drive more genuinely multilingual development in NLP? In this thesis, we offer evidence that the answer is "yes".

Using Amazon Translate, we translated SNLI and MultiNLI from English into Turkish to create the first large Turkish NLI data sets, NLI-TR, at a tiny fraction of the cost of creating them from scratch. Turkish is an interesting challenge in this context since it is very different

from English, most notably in its very free word order and complex morphology. A word in Turkish bears morpho-syntactic properties in the sense that phrases formed of several words in languages like English can be expressed with a single word form.

In our validation phase, a team of Turkish–English bilingual speakers assessed the quality of a large sample of the translations in NLI-TR. They found the quality to be very high, which suggests that translated datasets can provide a foundation for NLI research on a resource-constrained language, even if it has significantly different characteristics from English.

We then use these datasets to study the roles of pre-trained language models and morphological parsing in successful NLI systems for Turkish (See Section 3.3). For these experiments, we fit classifiers on top of pre-trained BERT parameters [25] and compare the original BERT-base release, the multilingual BERT embeddings released by the BERT team, and the Turkish BERT (BERTurk) embeddings of Schweter [140]. We find BERTurk to be superior to the others for NLI-TR.

Morphological parsing is a natural preprocessing step for Turkish due to its complex morphology. Thus, we assess the use of three morphological parsers as the second case study: Zemberek [29], BOUN parser [32], and Turkish Morphology [30]. We find that the parsers help where training data is sparse, but the need for a parser disappears as the training data increases. This is a striking finding: one might expect that Turkish would require morphological parsing given its complex word-formation processes. It might be regarded as welcome news, though, since the parsers are expensive to run. In Section 3.3.2, we report on some new optimizations of existing tools to make the relevant parsing jobs feasible, but we would still like to avoid these steps if possible, and it seems that we can for NLI.

Finally, we investigate how models trained on the machine translated datasets perform on the human translations from XNLI [24]. We find that machine translated and human translated sentences yield similar results, suggesting that it is safe to apply models trained on machine-translated datasets to human-written sentences.

3.2. Creating and Validating NLI-TR

3.2.1. English NLI Datasets

We translated the Stanford Natural Language Inference Corpus (SNLI; [133]) and the Multi-Genre Natural Language Inference Corpus (MultiNLI; [141]) to create labeled NLI datasets for Turkish, NLI-TR.

SNLI contains ≈570K semantically related English sentence pairs. The semantic relations are entailment, contradiction, and neutral. The premise sentences for SNLI are image captions from the Flickr30K corpus [142], and the hypothesis sentences were written by crowdworkers. SNLI texts are mostly short and structurally simple. We translated SNLI while respecting the train, development (dev), and test splits.

MultiNLI comprises ≈433K sentence pairs in English, and the pairs have the same semantic relations as SNLI. However, MultiNLI spans a broader range of genres, including travel guides, fiction, dialogue, and journalism. As a result, the texts are generally more complex than SNLI. In addition, MultiNLI contains "matched" and "mismatched" dev and test sets, where the sentences in the former set are from the same sources as the training set, whereas the latter consists of texts from different genres than those found in the training set. We translated the training set and both dev sets for NLI-TR.

3.2.2. Automatic Translation Effort

As we noted in Section 5.1, Turkish is a resource-constrained language with few labeled data sets compared to English. Furthermore, Turkish has a fundamentally different grammar from English that could hinder transfer-learning approaches. These facts motivate our effort to translate SNLI and MultiNLI from English to Turkish. We employ an automatic MT system and hope that it will deliver high-quality translations that we can use for NLI research and system development in Turkish.

Table 3.1. Sample translations from SNLI into NLI-TR. Each premise is associated with a hypothesis from each of the three NLI categories. Table A.1 in our supplementary materials provides MultiNLI examples.

		English	Turkish
SNLI	**Premise**	Three men are sitting near an orange building with blue trim.	Üç adam mavi süslemeli turuncu bir binanın yanında oturuyor.
	Entailment	Three males are seated near an orange building with blue trim.	Üç erkek mavi süslü turuncu bir binanın yakınında oturuyor.
	Contradiction	Three women are standing near a yellow building with red trim.	Üç kadın kırmızı süslemeli sarı bir binanın yanında duruyor.
	Neutral	Three males are seated near an orange house with blue trim and a blue roof.	Üç erkek mavi süslü ve mavi çatılı turuncu bir evin yakınında oturuyor.

We used Amazon Translate, a commercial neural machine translation service. Translation of all folds of SNLI and MultiNLI cost just US $2K (vs. the ≈US $100K we would expect for replicating these two datasets from scratch) and five days with no parallelization. We refer to the translated datasets as SNLI-TR and MultiNLI-TR, and collectively as NLI-TR. Translation examples are provided in Table 3.1. We publicly share NLI-TR.[11]

SNLI-TR and MultiNLI-TR are different from SNLI and MultiNLI in terms of token counts and vocabulary sizes. Table 3.2 illustrates these features before and after translation. For each fold in each dataset, translation decreased the number of tokens in the corpus, but it increased the vocabulary sizes drastically, in both the cased and uncased versions. Both of these differences are expected: many multiword expressions in English are translated into individual words due to the agglutinating nature of Turkish. For instance, the four-word English expression “when in your home” can be translated to the single word “evinizdeyken”.

[11] https://github.com/boun-tabi/NLI-TR

Table 3.2. Comparative statistics for the English and Turkish NLI datasets. The Turkish translations have larger vocabularies and lower token counts due to the highly agglutinating morphology of Turkish as compared to English.

Dataset		**English**			**Turkish**		
Name	**Fold**	**Token Count**	**Vocab Size (Cased)**	**Vocab Size (Uncased)**	**Token Count**	**Vocab Size (Cased)**	**Vocab Size (Uncased)**
SNLI	**Train**	5900366	38565	32696	4298183	78786	66599
	Dev	120900	6664	6224	88668	11455	10176
	Test	120776	6811	6340	88533	11547	10259
MultiNLI	**Train**	6356136	81937	66082	4397213	216590	187053
	Matched Dev	161152	14493	12659	112192	27554	24872
	Mismatched Dev	170692	12847	11264	119691	26326	23941

Table 3.2 also reflects the complexity difference between SNLI and MultiNLI that we noted in Section 3.2.1. Although SNLI contains more sentence pairs than MultiNLI, it has fewer tokens and a smaller vocabulary.

3.2.3. Translation Quality Assurance

Two major risks arise when using MT systems to translate NLI datasets. First, the translation quality might be low. Second, even if the individual sentences are translated correctly, the nature of the mapping from the source to the target language might affect the semantic relations between sentences. For example, English has the words “boy” and “girl” to refer to male and female children, and both those words can be translated to a gender-neutral Turkish word “çocuk”. Now, consider a premise sentence “A boy is running” and its contradiction pair “A girl is running”. Both sentences can be translated fluently into the same Turkish sentence, “Çocuk koşuyor”, which changes the semantic relation from contradiction to entailment.

Thus, to determine the viability of NLI-TR as a tool for NLI research, we must assess both translation quality and the consistency of the NLI labels. To do this, we assembled a

team of ten Turkish–English bilingual speakers (♡)[12] who were familiar with the NLI task and were either MSc. candidates or graduates in a relevant field.

For expert evaluation, we grouped the translations into example sets of four sentences as in Table 3.1, where the first sentence (premise) is semantically related to the rest (hypotheses). We distributed the sets to the experts so that each set (and sentence) was examined by five randomly chosen experts and each expert co-examined approximately the same number of sets with each other expert. Each expert evaluated the translation by (i) grading the translation quality between 1 and 5 (inclusive; 5 the best) and (ii) checking if the translation altered the semantic relation. We distributed an annotation guide[13] to the team to standardize the criteria. In total, 500 example sets (2,000 translated sentences) were examined by five experts, yielding 10,000 annotations.

Table 3.3. Translation quality and label consistency of the translations in SNLI-TR and MultiNLI-TR based on expert judgements. For the quality ratings (1–5), we report mean and standard deviation (in parentheses). For label consistency, we report the percentage of labels in SNLI-TR and MultiNLI-TR judged consistent with the original label, both in annotation- and sentence-level.

Dataset	**Fold**	**Translation Quality**	**Annotation-level Label Consistency**	**Majority-level Label Consistency**
SNLI-TR	**Train**	4.55 (0.78)	92.62%	98.67%
	Dev	4.46 (0.90)	90.53%	95.33%
	Test	4.45 (0.86)	87.87%	94.00%
MultiNLI-TR	**Train**	4.56 (0.80)	89.96%	96.22%
	Matched Dev	4.42 (0.86)	88.53%	95.33%
	Mismatched Dev	4.49 (0.82)	92.53%	98.00%
	All	4.51 (0.82)	90.72%	96.73%

[12]We greatly appreciate all our beloved friends for diligently providing annotation support: Alara Dirik, Berfu Büyüköz, Berna Erden, Fatih Mehmet Güler, Gökçe Uludoğan, Gözde Aslantaş, Melike Esma İlter (and Elif Beyza Gülaç), Ramazan Pala, Rıza Özçelik and Selen Parlar.

[13]`https://github.com/boun-tabi/NLI-TR`

We use the average translation score of the annotations to estimate translation quality. For label consistency, there are two comparisons we can make, since we have five new annotations per example. The "annotation-level" analysis compares each new annotation with the gold label on the original English example. The "majority-level" analysis compares only the majority label (if any) of the five new annotations with the English gold label. The annotation-level analysis is more stringent, whereas the majority-level analysis directly connects with how we expect NLI-TR to be most commonly used. Table 3.3 reports these analyses for SNLI and MultiNLI. The results are extremely reassuring. First, average translation quality is near 5 (ceiling) for all the splits. Second, annotation-level label consistency is over 90% and majority-level label consistency is over 95%, indicating that the linguistic differences between English and Turkish are not a major issue for preserving NLI labels.

To assess the reliability of the translation quality scores, we calculated the Intra-Class Correlation (ICC [143]). ICC is frequently adopted in medical studies to assess ordinal annotations provided by experts randomly drawn from a team. Its assumptions align well with our evaluation scheme. We obtained an ICC of 0.8426, which suggests excellent agreement [144, 145].

We also computed Krippendorff's alpha [146], which is an inter-annotator agreement metric used more commonly in NLP. This metric is suitable for both nominal and ordinal annotations involving multiple annotators. We calculated intercoder reliability of the ordinally-scaled translation quality score as 0.47. Our annotation-level label consistency yielded a score of 0.78 whereas our majority-level label consistency resulted in a score of 0.99. In contrary to the perfect agreement in the majority-level label consistency, the Krippendorff's alpha values of annotation-level labels and translation quality scores suggest less overall agreement than our ICC values do, but they are still acceptable, and ICC is arguably the more appropriate metric for our study. Krippendorff's alpha is generally used for large, diverse annotation teams, and its penalties for disagreements are known to be harsh.

Overall, it seems that the very high estimates of translation quality and label consistency of NLI-TR are trustworthy, and only a small percentage of premise–hypothesis have inconsistent semantic labels between their original and translated forms. Still, we would like to better understand why inconsistencies do arise. To this end, we inspected all 49 label-inconsistent pairs in our annotations. We find that low translation quality is the leading source of such errors, which further emphasizes how essential it is to work with high-quality translations.

Of the label-inconsistent pairs with good translations, we find that about 20 probably trace to differing perspectives on how to apply the NLI annotation guidelines. Relatedly, Conneau et al. [24] find that NLI labels often cannot be completely recovered by different annotators even with no sentence modifications.

Finally, we did find one example of label inconsistency that traces to a subtle difference between the English and Turkish lexicons. In this example, the premise "Your speeches are inflammatory" was translated to Turkish as "Konuşmalarınız çok kışkırtıcı", which can be back-translated as "Your speeches are provocative", while its entailment hypothesis "Your speeches upset people" was translated as "Konuşmaların insanları üzüyor", equivalent to "Your speeches make people sad". Annotators agreed that both of these translations are of maximum quality, but also stated that the Turkish pair should be labeled neutral. As bilingual speakers, we feel that this is essentially correct; the relevant English and Turkish adjectives are subtly different in ways that affect the NLI label. However, such examples seem to be rare and so pose minimal risk for conducting research using NLI-TR.

3.3. Experiments

3.3.1. Case Study I: Comparing BERT models on Turkish NLI Datasets

The arrival of pre-trained model-sharing hubs (e.g., Tensorflow Hub,[14] PyTorch Hub,[15] and Hugging Face Hub[16]) has democratized access to Transformer-based models [4], which

[14] https://github.com/tensorflow/hub
[15] https://pytorch.org/hub
[16] https://huggingface.co/models

are mostly in English. Combined with the abundance of labeled English datasets for fine-tuning, this has increased the performance gap between English and resource-constrained languages.

Here, we use NLI-TR to analyze the effects of pretraining Transformer-based models. We compare three BERT models trained on different corpora by fine-tuning them on NLI-TR. The results quantify the importance of having high-quality, language-specific resources.

3.3.1.1. Experimental Settings. We compared cased BERT-English (BERT-En), BERT-Multi, and BERTurk [140]. BERT-En is the original BERT-base model released by Devlin et al. [25], which used an English-only corpus for training. BERT-Multi was released by the BERT team as well, and was trained on a corpus containing texts from 104 languages, including Turkish. Schweter's BERTurk [140] also uses the same model architecture and is trained on a Turkish corpus (≈30GB).

We fine-tuned each model on train folds of NLI-TR separately and fixed the maximum sequence length to 128 for all experiments. Similarly, we used a common learning rate of 2×10^{-5} and batch size of 8 with no gradient accumulation. We fine-tuned each model for 3 epochs using HuggingFace's Transformers Library [26]. We evaluated the models on the test set of SNLI-TR and the "matched" and "mismatched" dev splits of MultiNLI-TR. Table 3.4 reports the accuracy of each model on the evaluation sets.

3.3.1.2. Results. Table 3.4 demonstrates that NLI-TR can be used to train high quality Turkish NLI models. We observe that every model performed better on the dev and test folds of SNLI-TR than the dev folds of MultiNLI-TR, which is an expected outcome given the greater complexity of MultiNLI compared to SNLI. The translation effort seems to have preserved this fundamental difference between the two datasets.

Table 3.4. Accuracy results for the publicly available cased BERT models on NLI-TR. BERTurk performed the best in all three evaluations, highlighting the value of language-specific resources for NLI.

	SNLI-TR		MultiNLI-TR	
Model Name	**Dev**	**Test**	**Matched Dev**	**Mismatched Dev**
BERT-En	81.83%	82.09%	69.98%	70.56%
BERT-Multi	85.37%	85.12%	75.97%	76.34%
BERTurk	**87.28%**	**87.04%**	**79.58%**	**80.87%**

In addition, BERTurk, which was trained on a Turkish corpus, achieved the highest accuracy, and BERT-Multi, which used a smaller Turkish corpus, was ranked the second, consistently on every evaluation fold. The ranking emphasizes the importance of having a Turkish corpus for pre-training.

3.3.2. Case Study II: Comparing Morphological Parsers on Turkish NLI Datasets

In this case study, we use NLI-TR to compare three morphological parsers with regular tokenization. We train a BERT model from scratch utilizing each approach for pretraining and use NLI-TR for fine-tuning. This leads to the striking result that morphology adds additional information where training data is sparse, but its importance shrinks as the dataset grows larger.

3.3.2.1. Experimental Settings. As the morphological parsers, we use Zemberek [29], BOUN [32], and Turkish Morphology [30] as parsers and compare them with an approach that does not do morphological parsing.

Zemberek is a mainstream Turkish NLP library used in research [147, 148, 149, 150, 151, 152] and applications such as iOS 12.2 and Open Office. It has 67,755 entries in its lexicon and uses a rule-based parser. BOUN implements the Turkish morphology rules described by Oflazer [153] with a Finite State Transducer, and its lexicon has 55,278 entries.

Finally, Turkish Morphology is an OpenFST-based [154] morphological parser that was recently released by Google Research and uses a lexicon with 47,202 entries.

Out of the box, Zemberek and BOUN can parse 398K and 51K tokens per minute respectively, whereas Turkish Morphology can process only 1K tokens. We sped up Turkish Morphology to parse 11 times more tokens per minute by implementing a dynamic programming wrapper [155] that increased the cache hit ratio to 89.9%. This technique is already used by Zemberek.

For the pretraining, we opted to use one-tenth (≈4GB, 500M tokens) of the Turkish corpus used by BERTurk [140] to pretrain BERT models in order to conduct a wide range of experiments on a limited budget. We analyzed each token morphologically using Zemberek, BOUN, and Turkish Morphology and trained a BERT model using the stems of the tokens only. For the model that does not utilize morphological information, we used tokens as they are. We used the `BertWordPieceTokenizer` class of HuggingFace Tokenizers[17] with the same set of parameters for each model.

We trained each model on a single Tesla V100 GPU of an NVIDIA DGX-1 system, allocating 128GB memory for 1 day. We split the dataset into 30 equal shards for parallel processing, where each shard comprises 1M sentences, and shuffled the shards prior to training to reduce the adverse effects of variance across the sentence styles in the different shards [156]. We used an effective batch size of 128 with gradient accumulation to address memory limitations.

We fine-tuned each model on NLI-TR with the same setting as in Section 3.3.1, with the exception that we trained for only 1 epoch. We measured the accuracy on the evaluation sets with an interval of 1,000 training steps to observe the effect of morphological parsing as the dataset grew. Figure 3.1 reports the accuracy of all models with respect to fine-tuning steps on NLI-TR development sets, and Table 3.5 shows the final accuracies.

[17] `https://github.com/huggingface/tokenizers`

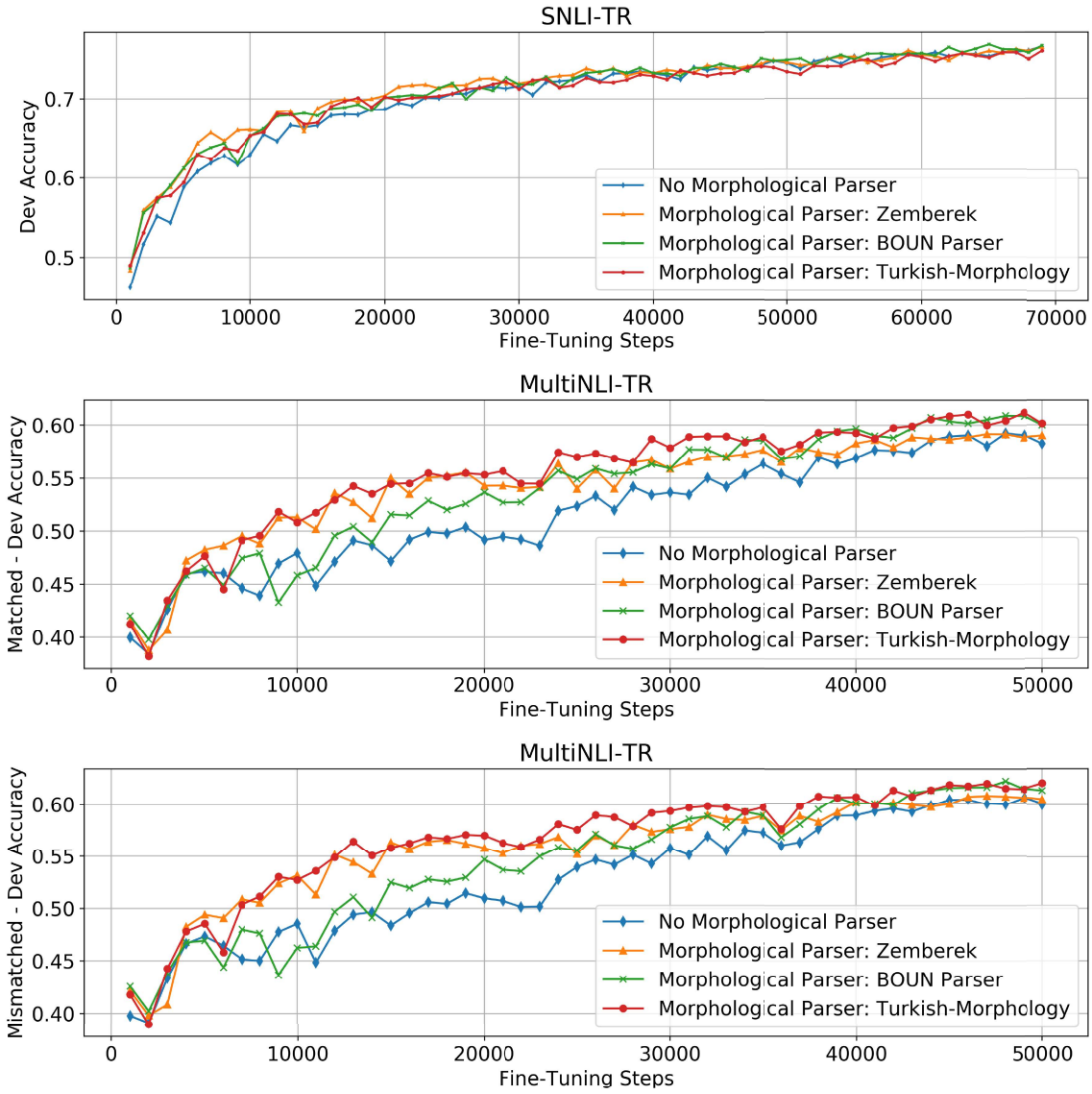


Figure 3.1. Development set accuracy for the three morphological parsers and a model without morphological parsing. The x-axis tracks the size of the training set. We find that morphological parsing is generally helpful in early rounds, when the training set is very small, but that its importance diminishes as the training set increases. These effects are especially clear for the two MultiNLI-TR dev sets.

3.3.2.2. Results. Figure 3.1 suggests that morphological parsing is beneficial where the training set is small, but its importance largely disappears for large training sets. This is reflected also in the final results in Table 3.5. We relate this to the fact that BERT models create contextual embeddings of both word and subword tokens [157, 158, 159]. Given a sufficiently large dataset, BERT models can approximate the effects of morphological parsing

even for Turkish, a morphologically-rich language.

Table 3.5. Accuracy results for different morphology approaches on NLI-TR. To facilitate running many experiments, these results are for pretraining on just one-tenth of the Turkish corpus used by BERTurk and fine-tuning on NLI-TR for just one epoch.

	SNLI-TR		**MultiNLI-TR**	
	Dev	**Test**	**Matched Dev**	**Mismatched Dev**
No Parser	76.50%	76.59%	58.24%	60.01%
Zemberek	76.47%	76.71%	59.01%	60.44%
BOUN Parser	**76.64%**	**76.89%**	59.99%	61.29%
Turkish Morphology	76.00%	76.36%	**60.13%**	**62.00%**

The trends are not uniform for SNLI-TR and MultiNLI-TR. For SNLI-TR, all three models display a similar learning curve, with a slight edge for Zemberek early on. For MultiNLI-TR, models with morphological parsers are more differentiated. However, all three converge to similar performance at the end of training on both datasets (Table 3.5).

In light of these findings, we suggest avoiding the use of morphological parsers for Turkish NLI where the training set is large, since the benefits of such parsers are generally not enough to offset the cost of running them.

3.3.3. Case Study III: Evaluating NLI-TR on Human-Translated Sentences

Thus far, we have used NLI-TR for both training and assessment. One might worry that machine-translated test sets are not reliable tools for measuring how models will perform on examples written by humans. In this section, we address this concern using the Turkish dev and test portions of XNLI, which were translated entirely by humans. The models we assess on XNLI are those from our first case study as well as models trained on a different machine-translated training dataset, MultiNLI-TR$^{\text{XNLI}}$. Overall, we find that performance on XNLI is consistently very similar to performance on NLI-TR.

3.3.3.1. Datasets. MultiNLI-TR^{XNLI} was created to investigate the performance of cross-lingual sentence embeddings compared to in-language ones (24). It provides machine translations of only the MultiNLI training set, so we report comparisons with just the corresponding section of NLI-TR, and we train models only on these two training sets.

3.3.3.2. Models. We used the BERT models from Case Study I (Section 3.3.1) for evaluation. We fine-tuned each model on the training sets of MultiNLI-TR and MultiNLI-TR^{XNLI} separately, following the same fine-tuning steps as in Section 3.3.1, and computed their accuracy on XNLI-Dev and XNLI-Test.

3.3.3.3. Results. Table 3.6 provides the results of the experiments. All three models consistently achieve higher accuracy on XNLI-Dev and XNLI-Test when fine-tuned with MultiNLI-TR, but the performance difference is modest. Table 3.6 also illustrates that BERTurk, backed by a Turkish-only training corpus, outperforms the other two models on all eight evaluations. Its performance is followed by BERT-Multi, which is trained on a corpus with texts in multiple languages, including Turkish. The same result was also shown in Case Study I using the evaluation splits of NLI-TR. Therefore, machine-translated MultiNLI-TR and human-translated XNLI display similar characteristics across evaluations, which lends further credence to our claim that MT can help provide a viable path to robust Turkish NLI.

To better understand how in-language pretraining (BERTurk) helps, we investigated the 57 hypotheses from XNLI-Dev and XNLI-test where BERTurk was successful and BERT-Multi was not. For these sentences, we observed that the BERT-Multi tokenizer was often unable to segment the words into meaningful Turkish subword units, most likely due to its training on a multilingual corpus. For instance, BERT-Multi often could not segment the suffix “-me/ma”, which negates a verb in Turkish, and thus bears crucial semantics for many contradiction examples [160]. This shows that in-language training is essential not only for good vector representations but also for effective tokenization.

Table 3.6. Accuracy results comparing NLI-TR with another machine translated dataset. NLI-TR performed better, but the gap is modest, suggesting that both datasets have value for Turkish NLI. Figure A.2 in our supplementary materials provides full learning curves. The results are very similar to those of Table 3.4 for MultiNLI, in overall quality and in the ranking of models.

	MultiNLI-TR		**MultiNLI-TR^{XNLI}**	
Model Name	**XNLI-Dev**	**XNLI-Test**	**XNLI-Dev**	**XNLI-Test**
BERT-En	66.99%	67.74%	65.66%	65.71%
BERT-Multi	73.82%	72.95%	71.61%	71.20%
BERTurk	**76.75%**	**78.72%**	**76.43%**	**76.43%**

We also hypothesize that subtle lexical distinctions are another factor in the performance difference between BERTurk and BERT-Multi. For example, though BERT-Multi successfully identified the semantic relations created by frequent pairs such as “hiç” (‘any’) and “hepsi” (‘all’), it missed many other distinctions like these. We propose that this is due to the more limited vocabulary of BERT-Multi for Turkish and the more robust word representations in BERTurk.

In addition to manual inspection, we computationally analyzed the pairs where BERT-Multi was unsuccessful and BERTurk was successful. We computed the frequency of each semantic class in the BERT-Multi predictions for these sentences and observed that the neutral class is the most common. This perhaps reflects the fact that neutral is the default choice where the model cannot robustly identify a semantic relation.

3.4. Summary

In this chapter, we shared the methodology to derive the first large Turkish NLI dataset, NLI-TR, by machine translating SNLI and MultiNLI. Though English and Turkish have very different grammars and thus stress-test automatic approaches, our team of experts judged the translations to be of very high quality and to preserve the original NLI labels consistently.

These results suggest that MT can help address the paucity of datasets for Turkish NLI. We release code, models, and data publicly for further research.[18]

We also used NLI-TR to investigate central issues in Turkish NLI. First, we used NLI-TR to analyze the effects of in-language pretraining. Second, we compared three morphological parsers for Turkish with simpler tokenization schemes. We found that a Turkish-only pretraining regime can enhance Turkish models significantly, and that morphological parsing is arguably worth its costs only when the training dataset is small. In our final case study, we returned to the general issue of translation quality, but now from the perspective of developing NLI systems. We showed that models trained on MultiNLI-TR perform well on the expert-translated test set from XNLI.

On the basis of these findings, we argue that MT can be more widely adopted for advancing NLP studies on resource-constrained languages. Though language-dependent tasks like dependency parsing are challenging to translate, MT can efficiently transfer large and expensive-to-create labeled datasets from English to other languages in many NLP tasks, including text classification, question answering, and text summarization. In addition, MT will presumably get cheaper, faster, and better over time, thereby further strengthening our core claims.

In the following chapters, we will apply our methodology to the Question Answering domain, automatically deriving the essential large-scale datasets required to train advanced Question Answering models in languages with limited resources.

[18] https://github.com/boun-tabi/NLI-TR

4. SQuAD-TR: BUILDING EFFICIENT AND EFFECTIVE OPENQA SYSTEMS FOR LOW-RESOURCE LANGUAGES

4.1. Introduction

In this chapter, we address the following research question: Can we build cost-effective QA systems for low-resource language contexts without having a gold training dataset? As a positive answer to this question, we propose a cost-effective approach to remedying the data scarcity problem for the QA task in non-English languages. This serves as a use case for low-resource language contexts.

We use the phrase "low-resource language context" to refer to *any language* as well as *any domain*—such as e-commerce [161], medical [162, 163], legal [164], finance [165], customer service [166], space [167]—wherein the available OpenQA data is scarce. In other words, the low resource status of a context depends both on the language and the domain: a high resource language with ample resources, like English, may also face data limitations within certain domains or scenarios.

Several effective methods have been proposed for overcoming various challenges and data limitations when building OpenQA systems for low-resource language contexts [168]. However, most current research primarily focuses on low-resource language contexts in English [168], emphasizing the need for analyzing the efficiency and effectiveness of these methods when applied to low-resource contexts in non-English languages.

Our proposal extends previous work on Standard QA in languages other than English [59, 63, 67], and we argue that adopting the OpenQA formulation of the problem is a key step to remedy data-scarcity issue for QA applications in non-English languages. OpenQA differs from standard QA by allowing the retrieval of relevant passages at runtime, rather than requiring them as part of the training dataset. For OpenQA, only gold question–answer pairs are required, and only for assessment. In particular, passages need not be a component of

the gold data, since they are retrieved by the system to use as (perhaps noisy) evidence. This eliminates the need for costly manual curation of gold passages for each question–answer pairs in training dataset, making the approach more scalable for low-resource languages. Our formulation still requires training data, but this can be created by automatic translation from English datasets. These translations may contain mistakes, but we show that they can still lead to robust QA systems. Specifically, our approach demonstrates that even noisy machine-translated data can produce accurate results for OpenQA formulation, significantly reducing the costs and time associated with manual dataset creation for low-resource languages. Whereas the cost of creating a dataset like SQuAD [8, 50] can be upwards of US$50,000, our costs are only around US$500, most of which is for machine translation services.

The cost of creating a gold assessment set could in principle be very large, but we show that one can get robust assessments of OpenQA systems with only around 200 question–answer pairs. Such gold datasets can be created by a small team very quickly. By showing that reliable OpenQA assessments can be conducted with as few as 200 gold question–answer pairs, we highlight a scalable and cost-effective path for OpenQA system development in resource-constrained languages. This extension broadens the applicability of OpenQA to low-resource languages further, offering a practical solution to the data scarcity issue in this context.

As a case study, we explore OpenQA for Turkish, beginning from English resources. We expect English/Turkish to be a challenging case for our approach, since the two languages are typologically diverse. This case can also serve as a representation of challenges found between other languages and English.

In this chapter, we make several key contributions to the field of OpenQA in low-resource language contexts:

- We demonstrate that QA is feasible in low-resource language contexts with the OpenQA formulation, even without any gold labels for training. This finding has significant implications for the accessibility and scalability of OpenQA systems in low-resource

language contexts.

- We provide in-depth qualitative and quantitative analyses on the efficiency and effectiveness of OpenQA systems in non-English contexts, particularly when using noisy labels obtained through machine translation, offering insights into potential improvements of these models.
- We show that only a few hundred gold assessment examples are needed to effectively evaluate OpenQA systems, significantly reducing the resources and time required for model evaluation in various language settings.
- Our results highlight that increasing the size of unstructured knowledge sources can have varying effects on the performance of OpenQA systems, depending on the ability of the retriever systems to manage noise effectively.
- We release SQuAD-TR, a large-scale Turkish Question Answering dataset, that was obtained by automatically translating SQuAD2.0. This resource facilitates building efficient and effective OpenQA systems in Turkish and serves as an example for other languages.

In Section 4.2.1, we introduce SQuAD-TR, an automatic translation of SQuAD2.0 [50] into Turkish using Amazon Translate.[19] SQuAD-TR serves as our noisy training data. For gold assessments, we rely on the XQuAD dataset [3] that provides a human-translated version of the SQuAD1.0 [8] dev set into Turkish (and nine other languages). Our core OpenQA system is a variant of the ColBERT-QA model of Khattab et al. [12], but it is built entirely over Turkish resources and is trained on SQuAD-TR. We show that the proposed methodology and resources can lead to highly effective standard QA systems (See Section 4.3.1) and OpenQA systems (See Section 4.2.4.2) for Turkish, and using only a few hundreds gold question–answer pairs is sufficient for the robust evaluation of QA systems. We make our code, models, and the dataset publicly available.[20]

[19] https://aws.amazon.com/tr/translate
[20] https://github.com/boun-tabi/SQuAD-TR

4.2. Datasets

4.2.1. SQuAD-TR

Inspired by previous work using machine translation as a stepping stone to obtain multilingual resources (Section 2.4), we translated SQuAD2.0 [50] to Turkish using Amazon Translate. We translated the titles, context paragraphs, questions, and answer spans in the original dataset. As a natural consequence, we needed to remap the starting positions of the answer spans, since their positions were not maintained in the translated paragraphs. This is needed not only due to linguistic variation between the source and target languages [59] but also because the translation task is inherently context dependent [63]. A text span may have totally different translations depending on its context. This is a challenging issue for obtaining consistent translations, particularly for Turkish due to the context-dependent morphological variation of Turkish words, as exemplified in Table 4.1. The problem with mapping all of the answer spans after translation is that it requires a substantial amount of time and manual work. However, it is still possible to recover part of them automatically, so we mapped the answer spans automatically in the target translations, as in much related work in different languages [59, 63, 66, 67].

In this automatic post-processing step, we first looked for spans of text in the context paragraph that exactly matched the answer text. If we found such a span, we kept that answer text along with its starting position in the translated text following the previous works [59, 63, 66, 67]. For answer texts without matching spans, we searched for the spans of text that approximately matched with the target answer text using character-level edit distance [169].[21] We use different edit distance values based on the length of the answer text. For answer texts with lengths shorter than 4 characters, we try to match all spans that are 1-edit distance away from the answer text. For all other answer texts, we search for all spans that are up to 3-edit distance away from the answer text and select all of the longest spans of texts that approximately match the target answer text. Table 4.1 shows examples of the answer spans that are recovered as a result of this post-processing.

[21] We used the implementation in the Python `regex` package: `https://pypi.org/project/regex/2021.4.4`

Table 4.1. Examples for the answer spans that are recovered in SQuAD-TR-TRAIN.

	Language	Context span	Question	Answer Text (Before post processing)	Answer Text (After post processing)
Example 1 *(Edit distance=1)*	Turkish	...Görünüşü, o yılki MTV Video Müzik Ödülleri'nin MTV tarihinde en çok izlenen yayın haline gelmesine ve **12.4 milyon** izleyiciyi çekmesine yardımcı oldu;...	2011 MTV Müzik Ödülleri'ni kaç kişi izledi?	12,4 milyon *(12.4 million)*	12.4 milyon *(12.4 million)*
	English	...Her appearance helped that year's MTV Video Music Awards become the most-watched broadcast in MTV history, pulling in **12.4 million** viewers;...	How many people watched the 2011 MTV Music Awards?	12.4 million	—
Example 2 *(Edit distance=2)*	Turkish	...Kariyerindeki en uzun süreli Hot 100 single'ı olma başarısına ulaşan "Halo"un ABD'deki başarısı, Beyoncé'nin **2000'li** yıllarda diğer kadınlardan daha fazla listede ilk on single elde etmesine yardımcı oldu....	Hangi on yıl boyunca, Beyonce'ın diğer kadınlardan daha fazla şarkısı vardı?	2000'ler *(2000s)*	2000'li *(2000s)*
	English	...The album featured the number-one song "Single Ladies (Put a Ring on It)" and the top-five songs "If I Were a Boy" and "Halo". Achieving the accomplishment of becoming her longest-running Hot 100 single in her career, "Halo"'s success in the US helped Beyoncé attain more top-ten singles on the list than any other woman during the **2000s**....	For which decade, did Beyonce have more top ten songs than any other woman?	2000s	—
Example 3 *(Edit distance=3)*	Turkish	...Amerika Kayıt Endüstrisi Birliği (RIAA), Beyoncé'yi 2000'lerin en iyi sertifikalı sanatçısı olarak toplamda **64 sertifikayla** listeledi....	2000'lerde kaç tane müzik sertifikası aldı?	64 sertifikasyon *(64 certifications)*	64 sertifikayla *(with 64 certificates)*
	English	...The Recording Industry Association of America (RIAA) listed Beyoncé as the top certified artist of the 2000s, with a total of **64 certifications**....	How many music certifications has she received in the 2000s?	64 certifications	—

This approximate matching is generally successful. However, for 25,528 question–answer pairs in SQuAD-TR-TRAIN, neither exact nor approximate matching returns a span in the translated paragraph. We excluded these question–answer pairs from SQuAD-TR-TRAIN and made them available in a separate file. This resulted in 259 paragraphs having no question–answer pairs. We excluded those paragraphs from SQuAD-TR-TRAIN as well. Similarly, we excluded 3,582 question–answer pairs from the SQuAD-TR-DEV dataset, but we did not need to exclude any paragraphs from SQuAD-TR-DEV, as all paragraphs had at least one question–answer pair where the answer text has a matching span in the paragraph.

With this procedure, we obtained the training and evaluation splits of SQuAD-TR, namely SQuAD-TR-TRAIN and SQuAD-TR-DEV, respectively. We used SQuAD-TR-TRAIN as a training dataset but did not use SQuAD-TR-DEV for evaluation in our research. We share it for future work. For evaluation, we instead used the Turkish split of XQuAD [3], namely XQuAD-TR, which helped maximize the validity of our assessment results, since it is a high-quality, human-translated test set.

Table 4.2. The basic statistics for the training and dev splits of SQuAD-EN and SQuAD-TR datasets along with the XQuAD-TR dataset.

Language	Dataset	Articles	Paragraphs	Question Count		
				Answerable	Unanswerable	Total
English	SQuAD-EN-TRAIN	442	19035	86821	43498	130319
	SQuAD-EN-DEV	35	1204	5928	5945	11873
Turkish	SQuAD-TR-TRAIN	442	18776	61293	43498	104791
	SQuAD-TR-DEV	35	1204	2346	5945	8291
	XQuAD-TR	48	240	1190	0	1190

Table 4.2 provides basic statistics of SQuAD-TR and XQuAD-TR along with the training and dev splits of the original SQuAD2.0 dataset (SQuAD-EN), noted as SQuAD-EN-TRAIN and SQuAD-EN-DEV, respectively. The number of articles is identical for the SQuAD-EN and SQuAD-TR datasets, whereas the SQuAD-TR-TRAIN dataset has fewer paragraphs and answerable questions than SQuAD-EN-TRAIN due to the excluded paragraphs and questions. Similarly, the SQuAD-TR-DEV dataset has fewer answerable questions than SQuAD-EN-DEV, for the same reason. As a matter of course, the number of unanswerable questions did not change in any split of the SQuAD-TR dataset, as the original unanswerable questions remain unanswerable after translation. We release SQuAD-TR publicly. [22]

4.2.2. Knowledge Source

As we discussed above, in OpenQA, evidence passages are not given to the reader along with the questions, but rather are retrieved from a large corpus. Thus, we first need to prepare a knowledge source containing the passages to be retrieved. We used the Turkish Wikipedia as the main part of our knowledge source. We obtained the passages in our knowledge base by extracting "contents" and "titles" from Turkish Wikipedia articles. However, we observed that the majority of the target information available in SQuAD2.0 [50] was not actually available in Turkish Wikipedia due to two main issues.

[22] https://github.com/boun-tabi/SQuAD-TR

One of these issues occurs when the article containing the target information in English Wikipedia is actually missing in Turkish Wikipedia. As an example, SQuAD2.0 has 50 question–answer pairs targeting 25 paragraphs about Canada's national public broadcaster "CBC Television" [23] referenced as an article in the English Wikipedia. However, there is no corresponding article for the same entity in Turkish Wikipedia but rather on "TRT",[24] which is Türkiye's national public broadcaster. Therefore, all the information required to answer the questions about the Canadian CBC Television is missing in Turkish Wikipedia. Another issue happens when the target article is actually available in Turkish Wikipedia with information-rich content but is missing the target information due to cultural bias. For example, SQuAD2.0 dataset has a question "When was the first known use of the word 'computer'?" targeting a passage in the English Wikipedia article "Computer".[25] The corresponding article "Bilgisayar"[26] in the Turkish Wikipedia[27] does not have any information about the etymological origin of the English word "computer", but instead the origin of its Turkish translation "bilgisayar". Asai et al. [88] succinctly describe the issues behind these two examples as "information scarcity" and "information asymmetry", which can be commonly called the "missing information" in the knowledge source of the target language.

The missing information issues will probably resolve gradually as the Turkish Wikipedia grows over time in terms of the number of articles and their quality. However, it is worth noting that this expansion may also introduce noise into the system, particularly when new articles act as distractors for the questions. To quantify the overall effect of the expansion of the knowledge source on the success of the OpenQA models, we used two different dumps of the Turkish Wikipedia with the dates spanning about 2 years,[28] which we call `Wiki-TR-2021` and `Wiki-TR-2023`.

The missing information issues will understate the performance of the retriever models in the OpenQA systems if not mitigated properly. To mitigate these issues, we appended

[23] `https://en.wikipedia.org/wiki/CBC_Television`
[24] `https://tr.wikipedia.org/wiki/TRT`
[25] `https://en.wikipedia.org/wiki/Computer`
[26] `https://tr.wikipedia.org/wiki/Bilgisayar`
[27] `Wiki-TR-2021`.
[28] We used the data dumps of May 31st, 2021 and May 1st, 2023

the target context passages of the SQuAD-TR-TRAIN and XQuAD-TR [3] datasets to the Turkish Wikipedia articles to complete our knowledge source. It should be noted that we do not append answer texts, but rather only the "contexts" and "titles". In this way, we made the target passages in our knowledge source available to our models while ensuring the validity of our experimental protocol. As a result, the total number of passages in our knowledge source increased slightly with the addition of 19,117 unique passages in the SQuAD-TR-TRAIN and XQuAD-TR datasets to the existing articles in the Turkish Wikipedia dump used.

We split the combined passages of varying lengths in the knowledge source into equal chunks of passages using an enhanced whitespace tokenizer, as in the DPR model [1]. The original DPR model for English segments the passages into 100-word chunks resulting in 142 tokens (subwords) on average when the BERT [25] tokenizer is used. Turkish sentences produce about 1.3 times longer sequence of tokens with the same number of words due to the very rich suffixing morphology of Turkish. For this reason, unlike Karpukhin et al. [1], we split the passages into 75 words instead of 100 words, as 100-word segments in Turkish run a high risk of being truncated by the ColBERT model [83], which accepts up to 180 tokens for documents by default. After splitting the combined passages into equal chunks of 75 words, we obtained a total of 1.7M and 2.1M passages for the Wikipedia dumps dated 2021 and 2023, respectively.

Table 4.3. Basic statistics for the knowledge sources used in our study and the one used in the DPR model of Karpukhin et al. [1].

					Passage Length	
	Language	**Short Name**	**Wikipedia Date**	**Passage Count**	**Word Count (Max)**	**Token Sequence Length (Avg)**
DPR	English	Wiki-EN-2018	Dec 20, 2018	21,015,324	100	142
Ours	Turkish	Wiki-TR-2021	May 31, 2021	1,719,277	75	136
		Wiki-TR-2023	May 01, 2023	2,192,776	75	136

The resulting combined passages then served as the knowledge sources in our study. The basic statistics of these knowledge sources and the one used in the original DPR model for English are given in Table 4.3.

4.2.3. Models

In this section, we share the background information about the retriever and reader models we used in our study. The same reader models are used for both the standard QA formulation and the OpenQA formulation in our experiments.

4.2.3.1. Retriever Models. In our study, we experimented with three retriever models, namely BM25, DPR, and ColBERT-QA. These models represent distinct approaches to document retrieval, ranging from traditional probabilistic relevance methods to modern neural network-based architectures. Each retriever model has been carefully selected to highlight specific strengths, from efficiency and simplicity to advanced contextual understanding. Below, we provide a detailed overview of each retriever model, their architectures, and their operational mechanisms.

Okapi BM25: The Okapi BM25 model, often abbreviated as BM25, is a probabilistic relevance algorithm that has been widely adopted for many years [170]. BM25 seeks to address core limitations of TF-IDF. For example, TF-IDF tends to be biased toward long documents. BM25 addresses this deficiency by incorporating document length normalization in addition to the conventional term frequency and inverse document frequency components.

The BM25 algorithm is formally defined as follows. Let q denote a query, d a document, q_i the i'th term of q, $f(q_i, d)$ the frequency of q_i in document d, $|d|$ the length of document d, and *avgdl* the average document length in the document collection. BM25 has two hyperparameters: $k_1 > 0$ adjusts the impact of term frequencies, and $0 \leq b \leq 1$ adjusts the document length penalty. The BM25 score for a document d with respect to a given query q is calculated as:

$$\mathit{BM25}(q, d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, d)}{f(q_i, d) + k_1 \cdot \left(1 - b + b \cdot \dfrac{|d|}{\mathit{avgdl}}\right)} \tag{4.1}$$

$$\text{IDF}(q_i) = \log \frac{N}{df_{q_i}} \tag{4.2}$$

where N is the total number of documents in the collection and df_{q_i} is the number of documents in the collection containing the term q_i.

In summary, the BM25 algorithm calculates a relevance score for a query q and document d based on the occurrence of the query terms in the document and the document collection, taking into account the document length to ensure a fair assessment.

DPR: The DPR model [1] employs a BERT-based dual-encoder architecture for the retriever component within an end-to-end OpenQA system. DPR has two BERT-based encoders: one for queries (denoted as $E(q)$) and another for documents (denoted as $E(d)$). Unlike similar dual-encoder setups that share an embedding layer [171], DPR uses separate word embedding layers for each encoder. The DPR encoders extract the representation from the built-in `[CLS]` token and output a fixed-size vector representation (dimension $d = 768$ for the BERT-base models). These encoders are utilized within the retriever component of an end-to-end OpenQA system during the training and test time.

The DPR encoders are trained by optimizing the negative log likelihood of the positive passage, d^+, against a batch of negative passages, $\mathbb{B} = \{d_1^-, d_2^-, \ldots, d_m^-\}$ for a given query q using the following loss function:

$$\text{NLL}(q, d^+, \mathbb{B}) = -\log \frac{e^{score(q,d^+)}}{e^{score(q,d^+)} + \sum_{i=1}^{m} e^{DPR(q,d_m^-)}} \tag{4.3}$$

where $DPR(q, d)$ is defined as:

$$DPR(q, d) = E(q) \cdot E(d) \tag{4.4}$$

which measures similarity between the query and document vectors. After training, the passages are encoded using the document encoder and indexed offline. The scoring function

in Equation 4.4 is also used for ranking the documents at inference time.

A lightweight retriever like BM25 provides a training bootstrap dataset for DPR training containing positive passages along with negative passages which are also referred as "hard-negatives" by Karpukhin et al. [1]. A crucial aspect of DPR involves learning the similarity between questions and passages by employing in-batch negatives distinct from hard-negatives. These in-batch negatives are composed of relevant passages from other questions within the same training batch and employed alongside hard-negatives. The primary strength of utilizing in-batch negatives lies in expanding the number of training examples effectively while keeping the memory footprint minimal. A clear strength of DPR is that it can encode all passages offline and store them in a fixed index, and query and document storing can be very fast. Its main weakness is that allows for only very minimal interactions between queries and documents.

ColBERT-QA: The ColBERT-QA system of Khattab et al. [12] is an OpenQA system built on top of the ColBERT retriever model [83]. In ColBERT-QA, the retriever is iteratively fine-tuned using weak supervision from the QA dataset so that it can perform task-specific retrieval. ColBERT-QA standardly uses an extractive reader, though its fine-tuned retriever is compatible with a wide range of reader designs.

The hallmark of the ColBERT model is its "late interaction" mechanism: both queries and passages are separately encoded into sequences of token-level vectors corresponding roughly to the output states of a BERT encoder [25]. Given a query q encoded as a sequence of token-level vector representations $[q_1, \ldots, q_m]$ and a passage d encoded as $[d_1, \ldots d_n]$, ColBERT computes the relevance score for q and d as

$$\textit{ColBERT}(q, d) = \sum_{i=1}^{m} \text{MaxSim}(q_i, d) \tag{4.5}$$

where $\text{MaxSim}(q_\text{i}, d)$ is defined as

$$\text{MaxSim}(q_\text{i}, d) = \sum_{i}^{n} \max_{\{d_j\}_{j=1}^{n}} q_i \cdot d_j \tag{4.6}$$

That is, we calculate the similarity of every pair of vectors q_i and d_j and sum the scores only for the highest scoring d_k for each q_i ("MaxSim").

The scoring function serves a dual purpose, being used not only during the training process but also in testing. During training, ColBERT optimizes the cross-entropy loss in a binary classification task using the scores, *ColBERT*(q, d^+) and *ColBERT*(q, d^-), for the triple (q, d^+, d^-) where q denotes the query, and d^+ and d^- denote the positive and negative passages with respect to the query, respectively [12]. For testing, this scoring function is the basis for ranking documents with respect to queries. The architecture allows all passages in the knowledge source to be encoded off-line and indexed for fast comparisons with query representations.

As a pure retriever, ColBERT achieves state-of-the-art results across a wide variety of IR benchmarks [172] and it can be implemented in a low-latency, space-efficient manner [173]. ColBERT-QA is a powerful example of recent general-purpose approaches to OpenQA, and so we base our models on this architecture. To adapt the model to Turkish, we made only language-specific adjustments (Section 4.2.4.2).

4.2.3.2. Reader Models. In both standard or OpenQA formulation of QA system, the reader model is responsible for extracting precise answers from the passages retrieved by the retriever. Unlike retrievers, which narrow down the search space, reader models focus on interpreting the textual context of passages to locate the most accurate and contextually relevant answers.

Our study employs several reader models, ranging from the foundational BERT to its advanced multilingual and language-specific variants such as mBERT, BERTurk, and XLM-RoBERTa. These models leverage pre-trained contextual representations to process and understand complex linguistic patterns, ensuring high-quality answer extraction.

In what follows, we describe the architecture, training methodology, and adaptations of each reader model used in our experiments, highlighting their suitability as a reader model for both standard QA and OpenQA formulations.

BERT: The BERT model [25] has emerged as a revolutionary NLP model, fundamentally altering how contextual understanding is achieved within sentences by employing Transformers [4] as its core architecture. One key landmark of BERT is that it is pretrained on large text corpora through self-supervision and adapts its representations to specific NLP tasks to deliver state-of-the-art results across a wide range of applications.

During BERT's pre-training, it uses two main objectives. The Next Sentence Prediction (NSP) objective involves predicting if one sentence follows another. The Masked Language Modeling (MLM) objective involves randomly masking or replacing words in input sentences and training BERT to predict the original words. BERT generates a contextual representation for every input token. For answer-span extraction, we follow Devlin et al. [25] in adding a span-classification head that predicts the start and end positions of the answer for a given question within a given passage.

mBERT and BERTurk: Since its inception, various BERT variants have emerged to suit specific linguistic contexts. The multilingual BERT model (mBERT [3]) is trained on multilingual data, enabling it to handle multiple languages without fine-tuning on language-specific data. Additionally, there are language-specific BERT models, like BERTurk [140] for Turkish, trained on extensive language-specific data to better capture nuanced characteristics of those languages. These variants showcase BERT's adaptability and versatility across diverse linguistic contexts, enhancing its utility for a wide range of natural language processing tasks.

In addition to the data-driven adaptations, BERT has also undergone architectural modifications, resulting in improved variants such as RoBERTa [174], XLM-RoBERTa [65], ELECTRA [175], and DeBERTa [176].

XLM-RoBERTa: Conneau et al. [65] represents an extension of the RoBERTa model [174], which is an optimized version of BERT designed to improve pre-training objectives and hyper-parameters. The key distinction of RoBERTa [174] compared to BERT lies in its optimized pre-training approach. Unlike BERT, which employs static masking patterns and the NSP task during pre-training, RoBERTa utilizes dynamic masking and removes the NSP task. These

modifications, among others, enable RoBERTa to generalize more effectively and outperform BERT on several downstream NLP tasks.

XLM-RoBERTa takes this a step further by focusing on cross-lingual understanding. As a cross-lingual version of RoBERTa, it is trained to understand text in multiple languages. This is particularly valuable for multilingual applications where a single model is needed to handle text in different languages without fine-tuning on language-specific data. Its versatility and effectiveness make it a valuable tool for multilingual NLP applications, where understanding and processing text across different languages is essential.

4.2.4. Standard QA and OpenQA Formulations

In this section, we provide a detailed description of the methodologies and settings followed during the execution of our standard QA and OpenQA experiments as well as any other pertinent details necessary for the reproducibility and transparency of our research findings.

4.2.4.1. Standard QA Formulation. To help establish an upper-bound for OpenQA in Turkish, we first conducted a series of standard QA experiments. Artetxe et al. [3] established a baseline for these experiments with an mBERT model [25] trained on `SQuAD-EN-TRAIN` [50] and tested on `XQuAD-TR` [3] as a crosslingual QA application. We extended this experiment in two ways. First, we changed the training dataset to `SQuAD-TR-TRAIN` while keeping all other aspects of Artetxe et al. [3]'s system fixed; the goal of this experiment is to begin to understand `SQuAD-TR-TRAIN` as a training resource. Second, we changed mBERT to BERTurk [140] to see the effects of pairing an in-language model with an in-language dataset. Third, we substituted BERTurk with XLM-RoBERTa [65] to understand the effect of this architectural improvement.

For these experiments, we finetuned the BERTurk, mBERT and XLM-RoBERTa models with the same hyperparameters using `SQuAD-TR-TRAIN` and `XQuAD-TR` as the training and test datasets. We used a batch size of 16, without gradient accumulation, on a single NVIDIA

Tesla V100 GPU. We applied 3×10^{-5} as the learning rate, used a maximum length of 384, with a document stride of size 128, and trained each model for 5 epochs using Huggingface's `transformers` library [177], Version 4.14.0.dev0.[29]

In all our reader models, we use standard evaluation metrics from the literature [8]: Exact Match (EM) and F1 scores. EM is the percentage of the predicted answer texts matching at least one of the ground-truth answer texts in an exact manner. F1 is the average of the maximum overlap ratio between predicted answer tokens and ground truth answer tokens. While EM gives no credit to predictions that have no exact match in any of the ground truth answer texts, F1 gives partial credit to those predictions that have at least one partially matching ground truth answer token. We calculated the evaluation metrics on `XQuAD-TR` as our test set.

4.2.4.2. OpenQA Formulation. In this section, we turn to OpenQA for Turkish. We establish baselines using BM25 and DPR [1] as examples of sparse and dense retrievers. Then, we share the results of our proposed system based on ColBERT-QA [12]. We first review the main components of our system, the retriever and the reader. We conduct the experiments for each Wikipedia dump as the knowledge source separately, which allows us to observe the overall effect of the growth in the knowledge source.

Retriever: The first step in building the DPR [1] and ColBERT-based [83] retrievers in our experiments involves a handful of steps that are specific to Turkish but that may have more general utility for cross-linguistic applications:

(i) Both DPR and ColBERT use the WordPiece tokenizer of the original BERT-base model [25] in English. We replace these tokenizers with the WordPiece tokenizer of the BERTurk cased model [140], which was pretrained on a large Turkish corpus.

(ii) The original tokenizer of the ColBERT model repurposes "unused" tokens in the tokenizer as query and document markers, which are available in the tokenizer of the

[29]The choice of specific values for the hyperparameters in our study is primarily aimed at establishing an initial reference point for future studies within a constrained budget.

original BERT-base model in English. As the BERTurk tokenizer did not have such unused tokens, we use alternative tokens for the document and question marker tokens. For the query marker we use a "blush" emoji (U+1F60A) and for the document marker we use a "smiley" emoji (U+1F603), as they are unlikely to occur in the Wikipedia articles yet likely to be present in various non-English BERT models.

(iii) Both DPR and ColBERT originally initialize their weights using those of the original BERT model in English. We use the BERTurk weights to initialize the DPR and ColBERT weights before starting the training step. For languages without high-quality language-specific embeddings like BERTurk, one might use multilingual embeddings here instead.

In light of the above steps, the resulting retrievers might more properly be called the DPRTurk and ColBERTurk retrievers. In the interest of clarity, we will continue to refer to them as the DPR and ColBERT models.

To train the retriever, we proceed in two phases, as outlined in Figure 4.1. In phase 1 (top row of the figure), we build our baseline retriever models. We rely on BM25 to index our knowledge sources, using `pyserini` [178][30] and `anserini` [179][31] wrappers for the Apache Solr search engine. We customize Apache Solr for Turkish by incorporating the Zemberek[32] plugin [180] as a morphological stemmer for Turkish. In addition to the BM25 retriever, we use the DPR model to index our knowledge sources and build our baseline dense retriever.

In our experiments, the BM25 retriever provides the bootstrap dataset to train DPR and ColBERT-QA. With this lightweight BM25 retriever, we create a dataset of triples (q, d^+, d^-), where q is a question in `SQuAD-TR-TRAIN`, d^+ is a positive passage containing the target answer span for q, and d^- is a negative passage that does not contain the target answer span for q. Both d^+ and d^- are from the top k results retrieved from the BM25 index. Specifically, we create the dataset of triples by pairing every d^+ with every other d^- where d^+ is from the top k^+ results and d^- is from the top k^- results ($k^+ \leq k^-$) obtained for each question q.

[30] `https://github.com/castorini/pyserini`
[31] `https://github.com/castorini/anserini`
[32] `https://github.com/iorixxx/lucene-solr-analysis-turkish`

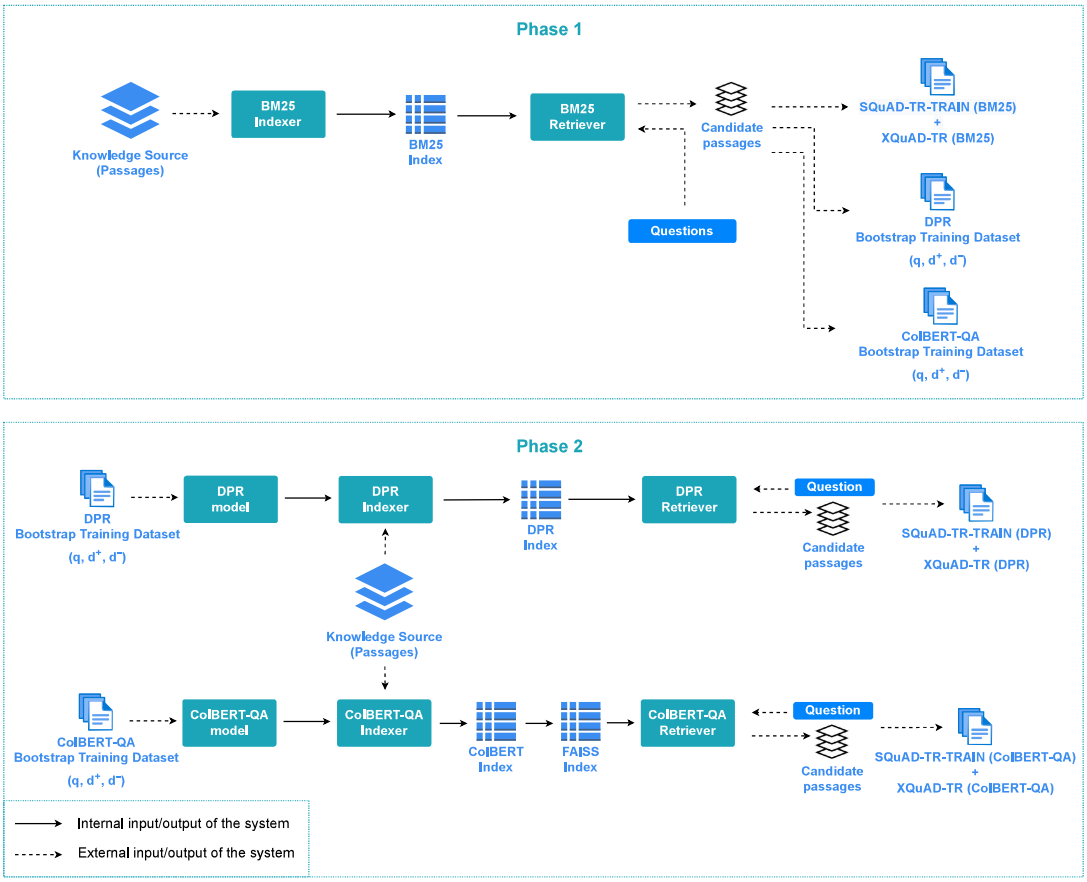


Figure 4.1. System overview diagram for the OpenQA retriever component. We assume the knowledge source is independent of the system for the sake of clarity in the figure.

More specifically, we selected a maximum of three positive passages from the top $k = 20$ results for both the DPR and ColBERT-QA models. In the case of DPR, we selected the top most negative passage out of top 20 results and used it as the hard-negative passage in a training example following Karpukhin et al. [1]. As for ColBERT-QA, we paired each positive passage with a negative passage from the top 100 results, per the method outlined by Khattab et al. [12].

For both of the Turkish Wikipedia dumps used as the knowledge source, the resulting bootstrap training dataset for the DPR model and the ColBERT-QA model contains 64K and 6M triples respectively, for 86K questions, where $k^+ = 3$ and $k^- = 20$. It is worth mentioning that the DPR model amplifies the size of its training bootstrap dataset by incorporating the in-batch negatives during training. Additionally, it can be noted that we could use all question–

answer pairs in SQuAD-TR that were originally labeled as answerable before translating SQuAD2.0. The reason for also including those question–answer pairs that we excluded from SQuAD-TR-TRAIN (Section 4.2.1) is that the retriever model, unlike the reader, does not require the location of the answer span in the context. Therefore, the retriever model can utilize all question–answer pairs in SQuAD-TR-TRAIN.

In phase 2 (bottom row of Figure 4.1), we use our BM25-derived datasets to train a DPR model and a ColBERT-QA model, and then we index our knowledge source using this retriever. These indexers compute the passage representations using DPR and ColBERT-QA to project them into an embedding space where the question and passage representations are close to each other if the passage has an answer for the question.

We trained the DPR model for Turkish on 6 NVIDIA RTX A6000 GPUs in parallel. We used a batch size of 128 without gradient accumulation, aligning with the configuration yielding the best reported scores [1]. Other parameters were set to their default values as specified in the original implementation. We indexed all the passages in the knowledge source, encoded by the resulting DPR model, using FAISS [181] in flat mode as the default configuration.

For training the ColBERT-QA model in Turkish we used a single NVIDIA RTX A6000 GPU with a maximum document length of 180 and a batch size of 32 without gradient accumulation. Then, we indexed all the passages in the knowledge source once again, this time using ColBERT-QA. Following Khattab et al. [12], we further reindexed ColBERT-QA indexed document embeddings using FAISS [181] in IndexIVFPQ mode with a 16384 partition and a sample rate of 0.3 to speed up the retriever component.

Since there is no state-of-the-art model for OpenQA in Turkish yet, we compare the retriever and reader performance of our models with the performance of models based on the baseline BM25 and DPR retrievers. It is important to note that each retriever determines its own versions of the SQuAD-TR-TRAIN and XQuAD-TR [3] datasets specific to that retriever and to the knowledge source it uses. For each retriever, we retrieve the top k passages for each

question in SQuAD-TR-TRAIN and XQuAD-TR to set a context passage for that question from the top k results. In both the training and testing phases, we use the first positive result out of top 5 retrieved results. Should there be no positive result among these top 5 results, we resort to the first result, regardless of whether it is positive or negative.

The success of the retriever component sets an upper bound for the reader. Following previous works (DrQA [182]; DPR [1]; REALM [81]; ColBERT-QA [12]), we evaluated the success of the retriever component by means of "Success@k", also noted as "S@k", which is the ratio of the questions having a positive passage among the top k results retrieved from the index. We evaluated "S@k" values for $k \in \{1, 5, 20\}$. We supplemented "S@k" with another metric, "Count@k", also noted as "C@k", which is the average number of positive passages among the top k results retrieved for each question.

Turkish has different morphological characteristics from English, as it is an agglutinative language and has more morphological variants of each word. For this reason, the evaluation scores depend heavily on the tokenization scheme that is used when evaluating the results. Therefore, we used three different tokenization schemes: whitespace, morphological, and "enhanced whitespace". Whitespace tokenization calculates the "S@k" and "C@k" values after splitting the retrieved passage and answer text into tokens whenever it finds a whitespace character. The morphological tokenization scheme segments the passage and answer texts into a list of stems by stripping all suffixes in all words before calculating the "S@k" and "C@k" values. The enhanced whitespace tokenization, which is the standard tokenizer of the DPR model, breaks the text into tokens not only when it encounters a whitespace character but also whenever it finds a list of predefined punctuations. We used the uncased version for all tokenization schemes to bring them in line with the output of our morphological tokenizer (Zemberek [29]), which was uncased out of the box.

While the performance of the retriever models on the evaluation dataset (XQuAD-TR) offers useful insights, a better assessment in the context OpenQA comes from evaluating the retrievers on the training dataset (SQuAD-EN-TRAIN). This is because we leverage the retrievers' outputs from the training dataset to prepare the retriever-specific training datasets

for the respective readers. It is important to emphasize that the performance of the retrievers on SQuAD-TR-TRAIN and XQuAD-TR may vary based on their ability to handle the machine-translated texts and human-translated texts. Hence, we also assess the retrievers' performance on SQuAD-TR-TRAIN.

Reader: In line with the three sets of OpenQA retrievers obtained in the retriever step for each knowledge source with different Wikipedia dumps, we build three sets of reader components in our OpenQA system in Turkish, depending on which retriever their training and test sets are based on. Figure 4.2 summarizes this process. We used the XLM-RoBERTa [65] model along with its original tokenizer for each reader, and we finetuned them using the retriever and knowledge source specific versions of SQuAD-TR-TRAIN, namely SQuAD-TR-TRAIN (BM25,YYYY), SQuAD-TR-TRAIN (DPR,YYYY) and SQuAD-TR-TRAIN (ColBERT-QA,YYYY), where YYYY $\in \{2021, 2023\}$ denotes the year of the Wikipedia dump.

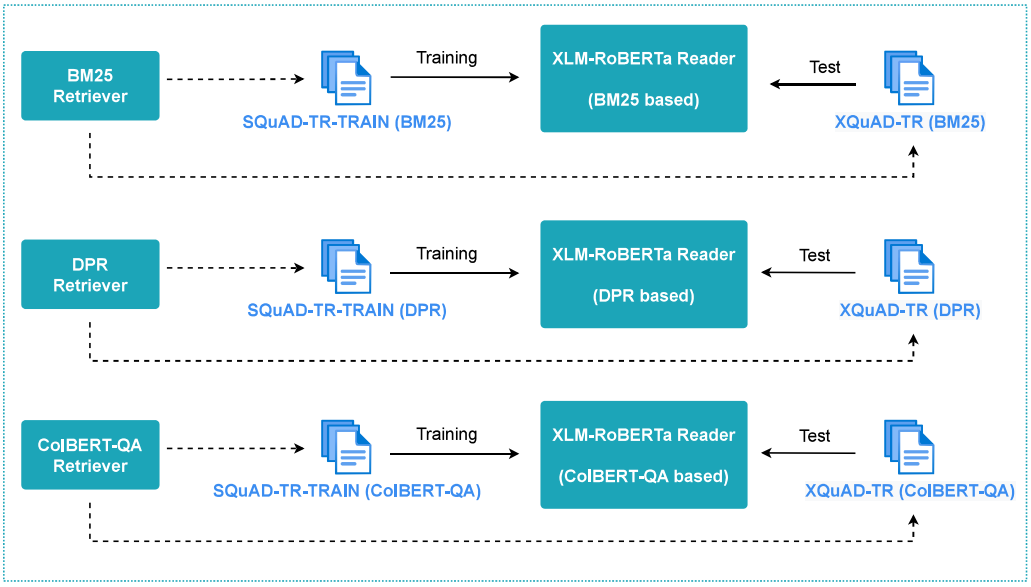


Figure 4.2. Overview diagram for the OpenQA reader component. For the sake of clarity in the figure, we assume the knowledge source that retrievers are based on is the same across all systems.

We used the same hyperparameters as described in Section 4.2.4.1 to train and evaluate the BERTurk model on the retriever-specific datasets as shown in Figure 4.2. We also calculated EM and F1 scores on the open versions of XQuAD-TR [3] for each reader model.

Subsampling Test Sets for OpenQA: One of our central goals is to efficiently create OpenQA systems. Machine translation costs are already manageable and controlled. However, creating gold test sets can lead to unexpectedly high costs, especially if the goal is to have thousands or tens of thousands of examples. Thus, a key question for us is: "How small can our test sets be?"

To begin to address this question, we ran a series of experiments in which we subsampled the OpenQA test sets that we obtained using Wiki-TR-2023. To create test subsets, we randomly picked samples in sizes of 100, 200, 500, and 1000 from the complete test datasets, alongside using the full test datasets. We then evaluated each retriever's performance on the subsampled questions of different sizes. Similarly, for the readers, we obtained prediction results of the readers on these subsampled examples to measure the readers' performance across subsets of various sizes. This process was repeated 20 times for each size of the subsampled reader datasets, aiming to maintain a diverse representation of the entire test set for each subset size, thereby ensuring the reliability of the findings derived from the evaluation process.

4.3. Experimental Results

In this section, we report the experimental results from both standard QA and OpenQA formulations. Additionally, we offer an overview of the resource usage patterns of the models involved.

4.3.1. Standard QA Results

Table 4.4 summarizes the results of the experiments for standard QA, where we experiment with mBERT, BERTurk and XLM-RoBERTa as readers and SQuAD-EN-TRAIN and

SQuAD-TR-TRAIN as training data. The results show that XLM-RoBERTa yields the highest scores, even outperforming an in-language model, BERTurk, when trained on an in-language dataset SQuAD-TR-TRAIN. The largest performance gap occurs when SQuAD-EN-TRAIN is replaced with SQuAD-TR-TRAIN, indicating that in-language datasets are essential for high-performing standard QA models, even if the datasets are machine-translated and potentially noisy.

Table 4.4. Reader results for the standard formulation of QA task tested on XQuAD-TR.

Reader Model	**Training Dataset**	**EM**	**F1**
mBERT [3] - *Baseline*	SQuAD-EN-TRAIN	–	55.40
mBERT	SQuAD-TR-TRAIN	50.00	64.76
BERTurk	SQuAD-TR-TRAIN	51.17	67.78

The use of an in-language model instead of a multi-lingual one has a smaller positive impact on the performance when the model architecture remains the same. This aligns with other recent findings in the literature [43, 183].

4.3.2. Open QA Results

The OpenQA results presented in this section evaluate the effectiveness of the retriever and reader models within the OpenQA framework. By leveraging a combination of traditional and neural retrieval methods, alongside state-of-the-art reading architectures, we aim to understand how these models perform individually and collectively in extracting answers correctly.

We also explore the impact of different tokenization schemes, training datasets, and knowledge sources on the performance of the retrievers. Additionally, we analyze how the interaction between retriever and reader models influences overall system performance, offering insights into the advantages and limitations of each approach.

4.3.2.1. Retriever Results. The performance results of each retriever on XQuAD-TR are shown in Table 4.5. The results indicate that ColBERT [83] is markedly more effective than the baseline BM25 and DPR models, even though the BM25 retriever is empowered with a morphological stemmer as described in Section 4.2.4.2. We observe a performance improvement over the BM25 and DPR [1] models independent of the tokenization scheme. The results suggest that the ColBERT-QA retriever will give the reader module a better chance at finding correct answers.

Table 4.5. Retriever results for the OpenQA formulation of QA task evaluated on "XQuAD-TR". All tokenizers are uncased. The highest values in each column are shown in **bold**. For equal pairs, the larger ones on more significant digits are underlined.

		Whitespace			Morphological			Enhanced Whitespace		
Retriever Model	**Knowledge Source**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**
BM25 - *Baseline - Sparse*	Wiki-TR-2021	42.79/0.43	58.91/0.85	66.64/1.17	45.97/0.46	62.27/0.95	69.92/1.39	56.30/0.56	73.53/1.11	82.10/1.55
	Wiki-TR-2023	41.68/0.42	58.15/0.85	66.22/1.18	44.62/0.45	61.43/0.95	69.66/1.42	55.21/0.55	72.52/1.10	81.18/1.55
DPR - *Baseline - Dense*	Wiki-TR-2021	40.59/0.41	56.72/0.80	63.78/0.93	43.69/0.44	59.92/0.89	67.56/1.35	52.10/0.52	70.67/1.03	79.32/1.48
	Wiki-TR-2023	38.15/0.38	54.37/0.77	62.61/0.92	40.92/0.41	57.48/0.86	66.22/1.32	48.99/0.49	68.23/0.99	77.90/1.46
ColBERT-QA	Wiki-TR-2021	58.99/0.59	**70.34**/1.05	72.77/1.38	62.27/0.62	**74.03**/1.15	78.07/1.64	75.88/0.76	**88.23**/1.38	92.10/1.83
	Wiki-TR-2023	**60.50/0.61**	**70.34/1.07**	**74.54/1.44**	**63.78/0.64**	73.87/**1.18**	**78.32/1.72**	**77.05/0.77**	87.81/**1.40**	**92.18/1.91**

Comparing the baseline retriever models, we noticed a substantial performance advantage of BM25 over the DPR retriever. This finding is consistent with the reported performance of DPR [1] on the English SQuAD 1.1 [8] dataset. This is attributed to the fact that the annotators of the SQuAD datasets [8, 50] tended to formulate questions with significant lexical overlap with their passages, thereby providing an advantage to BM25. Karpukhin et al. [1] also point out the skewed distribution of the target Wikipedia passages compared to the vast number of Wikipedia articles in the knowledge source as another contributing factor. This performance discrepancy also suggests that the DPR model is less effective in mitigating the inherent noise present in the knowledge source. To address this issue, they suggested a hybrid approach that combines the outcomes of BM25 and DPR to achieve a result that surpasses each individually. Given that both models yield higher $S@k$ values as k increases, these models can benefit from an effective reranker tailored to the noisy data for further improvement [184]. When we compare the baseline models with the ColBERT retriever, we observe that ColBERT achieves markedly superior performance than the baselines on

XQuAD-TR. This indicates the outstanding effectiveness of ColBERT retriever in handling and ranking examples characterized by lexical overlap as well as those requiring deep semantic understanding all while suppressing noise in the knowledge source and the training dataset.

The results also indicate that the morphology-unaware "enhanced whitespace tokenizer" identifies the correct results better than the morphological tokenizer for all values of S@k and C@k, suggesting that computationally-intensive morphological stemming can be avoided when evaluating QA systems in Turkish. Although the negative effect of morphological stemming may be surprising given the rich morphology of Turkish, this result is in line with previous literature [43].

As another perspective, we observed consistently diminishing results in the S@k values in the BM25 and DPR models when utilizing the newer Turkish Wikipedia dump and evaluating on XQuAD-TR. However, in the same scenario, we noted a consistent increase in most of the S@k and C@k scores for the ColBERT-QA model. It seems that, for DPR and BM25, the benefits of adding more relevant passages were outweighed by the interference effects from negative passages. In contrast, ColBERT-QA seems to be better able to suppress these interfering factors and benefit from the additional relevant data.

Table 4.6. Retriever results for the OpenQA formulation of QA task evaluated on "SQuAD-TR-TRAIN". All tokenizers are uncased. The highest values in each column are shown in **bold**. For equal pairs, the larger ones on more significant digits are underlined.

		Whitespace			Morphological			Enhanced Whitespace		
Retriever Model	**Knowledge Source**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**
BM25 - *Baseline - Sparse*	Wiki-TR-2021	19.04/0.19	28.10/0.43	35.09/0.73	25.93/0.26	37.38/0.60	45.43/1.02	28.91/0.29	40.53/0.65	48.39/1.07
	Wiki-TR-2023	18.65/0.19	27.66/0.43	34.87/0.73	25.42/0.25	36.85/0.59	45.19/1.02	28.31/0.28	39.95/0.65	48.04/1.09
DPR - *Baseline - Dense*	Wiki-TR-2021	27.11/0.27	36.14/0.56	41.11/0.91	37.18/0.37	47.70/0.78	52.47/1.29	41.73/0.42	51.91/0.86	56.01/1.37
	Wiki-TR-2023	26.71/0.27	35.82/0.56	40.99/0.91	36.64/0.37	47.39/0.77	52.37/1.29	41.17/0.41	51.54/0.85	55.85/1.06
ColBERT-QA	Wiki-TR-2021	**33.52/0.34**	**40.27/0.69**	44.14/1.07	**46.07/0.46**	**52.73**/0.95	56.22/1.50	**51.48/0.51**	**56.97**/1.05	59.87/1.61
	Wiki-TR-2023	33.18/0.33	40.25/**0.69**	**44.40/1.10**	45.75/**0.46**	52.72/**0.96**	**56.39/1.56**	51.13/**0.51**	56.82/**1.06**	**59.91/1.67**

While the performance of the retriever models on XQuAD-TR offers valuable insights, there is a need for a more effective method to explore how the retriever models interact with the reader model to improve the overall system performance. Given that the information

transfer between the retrievers and readers primarily occurs through the retriever-specific training dataset prepared for the readers, we investigated the retrievers' performance on SQuAD-TR-TRAIN, as shown in Table 4.6.

The most notable observation we initially make in Table 4.6 compared to Table 4.5 is the varying performance of DPR relative to BM25 on XQuAD-TR and SQuAD-TR-TRAIN. This finding shows that DPR yields more examples with positive passages for the reader's training dataset compared to BM25. Consequently, we observe that DPR performs better on the machine-translated dataset compared to BM25, whereas BM25 surpasses DPR on the human-translated dataset. ColBERT-QA continues to demonstrate consistently strong results in both datasets.

Another noticeable difference between Table 4.5 and Table 4.6 is the slight decline in performance of the retriever models on SQuAD-TR-TRAIN as the knowledge source expands, evident across nearly all S@k and C@k scores for all models, including ColBERT-QA. This outcome suggests that the resulting training datasets for readers may contain fewer examples with positive passages, thus offering fewer chances to improve the respective readers. It is also worth noting that a decrease in the number of examples containing positive passages could affect reader scores differently depending on whether it occurs in the test dataset or the training dataset. Specifically, a reduction in the number of examples with positive passages in the test set may directly impact reader scores negatively. However, if the number of training examples is already sufficient to saturate the overall reader performance, a decrease in positive passages in the training dataset may have a lesser impact on reader scores.

We also evaluated the retrievers in terms of the ranking of the answers returned, which is a widely used metric in the information retrieval domain. In order to assess this property, we computed mean reciprocal rank (MRR) for the questions correctly answered by all retrievers by returning the same relevant passage among their top k results at different ranks. All retrievers can return the same passage as the highest-ranked relevant passage among their top k results, even though the passage's exact rank may differ in each retriever's output. Table 4.7 shows the MRR scores of the retrievers on the examples where all retrievers return the same

relevant passage with the highest rank among their top k results. The scores are calculated on both SQuAD-TR-TRAIN and XQuAD-TR with Wiki-TR-2021 and Wiki-TR-2023 to reflect the ranking behaviors of the models on the training and test datasets as the knowledge source expands.

Table 4.7. The MRR scores of the retriever models evaluated on the ranks of the same relevant passages unanimously returned from all retrievers as the highest-ranked positive passages.

Retriever Model	Wiki-TR-2021		Wiki-TR-2023	
	SQuAD-TR-TRAIN	XQuAD-TR	SQuAD-TR-TRAIN	XQuAD-TR
BM25 - Baseline (Sparse)	0.8262	0.8879	0.8242	0.8806
DPR - Baseline (Dense)	0.9267	0.8474	0.9239	0.8320
ColBERT-QA	0.9867	0.9575	0.9868	0.9637

The MRR scores presented in Table 4.7 offer two noteworthy insights. First, the DPR model consistently retrieves positive passages with higher confidence compared to BM25, regardless of the dataset or knowledge source used. While this behavioral difference is somewhat aligning with the relative performance of these retrievers on SQuAD-TR-TRAIN (Table 4.6), it contradicts their relative performance on XQuAD-TR (Table 4.5). This observation warrants further investigation when assessing the performance of the corresponding readers of these models. Second, the MRR scores of BM25 and DPR models decrease when the knowledge source expands, whereas the performance of ColBERT-QA improves. Once again, this finding suggests that ColBERT-QA can effectively leverage the potential increase in noise associated with the expansion of the knowledge source to better distinguish positive passages.

Qualitative Analysis of the Retriever: In order to observe the strengths and weaknesses of the retrievers relative to each other, we manually analyzed the passages retrieved for the questions in the test set by the two top-performing retrievers on XQuAD-TR, ColBERT-QA [12] and BM25. The analysis revealed a number of factors that help to explain the performance differences between the sparse and dense retriever models [185].

One important factor is the TF-IDF-based scoring mechanism used in BM25, which results in the retrieval of irrelevant passages that excessively mention the uncommon content words in the questions. While this approach proves advantageous when there are only a few relevant candidate passages, it comes with significant side effects when the model needs to suppress multiple related passages for the question to return the actual relevant passage. The decrease in the S@k and C@k values for the BM25 retriever as the knowledge source expands (Table 4.5) indicates that BM25 is ineffective in inhibiting new irrelevant signals.

To highlight both positive and negative effects of TF-IDF approach, Table 4.8 and Table 4.9 depicts two question–answer pairs[33] and passages returned by each retriever. The first example shows that TF-IDF-based scoring misleads the retriever and causes it to retrieve irrelevant passages containing the content words. Conversely, in the second example, where there were limited relevant candidate passages available, BM25 identified the correct passage. This observation aligns with the qualitative analysis conducted by Karpukhin et al. [1] for English OpenQA, which compares the results of BM25 and DPR.

[33] Question numbers are the sequence numbers of the questions in XQuAD-TR.

Table 4.8. The "negative" effect of the TF-IDF approach used in BM25 are exemplified in Question 165. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results.

	Turkish	**English**
Question 165	Bir öğretmenlik sertifikasının geçerli olduğu en uzun süre nedir?	What is the longest time that a teaching certificate is good for?
Answer	on yıla	ten years
BM25	... için kullanılır. Genelde bu iptal bilgilerinin izlenmesinde kullanılır. Subject (Özne): Sertifikanın ait olduğu varlık: bir cihaz, birey, ya da kurum. Issuer (Sağlayıcı): Bilgileri doğrulayan ve sertifikayı imzalayan kuruluş. Not Before (Önce Değil): Sertifikanın geçerli olduğu en erken saat ve tarihi. Not After (Sonra Değil): Sertifikanın geçerli olduğu en geç saat ve tarihi. Key Usage (Anahtar Kullanımı): Sertifikanın açık anahtarındaki geçerli kriptografik kullanım. Ortak alanlar arasında dijital imza doğrulaması, anahtar şifreleme ve sertifika imzalama bulunur. Extended ... Source: `Wiki-TR-2023`	It is used for monitoring the validity information. Subject: The entity to which the certificate belongs: a device, individual, or organization. Issuer: The organization that verifies the information and signs the certificate. Not Before: The earliest date and time at which the certificate is valid. Not After: The latest date and time at which the certificate is valid. Key Usage: The valid cryptographic usage in the public key of the certificate. Common fields include digital signature verification, key encryption, and certificate signing. Extended ...
ColBERT-QA	...yönelik gereksinimler, genelde tam zamanlı profesyonellere yönelik gereksinimler kadar sert değildir. İş Gücü İstatistikleri Bürosu, ABD'de 1,4 milyon ilkokul öğretmeni, 674.000 ortaokul öğretmeni ve 1 milyon lise öğretmeni istihdam edildiğini tahmin etmektedir. Amerika Birleşik Devletleri'nde her eyalet devlet okullarında öğretmenlik yapma lisansı almak için gereksinimleri belirler. Öğretim sertifikasyonu genelde üç yıl devam eder, ama öğretmenler **on yıla** varan uzunlukta sertifikalar alabilirler. Devlet okulu öğretmenlerinin bir lisans derecesine sahip olması şart koşulmakta ve öğretmenlerinin çoğunun eğitim ... Source: `XQuAD-TR`	...are generally not as rigorous as those for full-time professionals. The Bureau of Labor Statistics estimates that there are 1.4 million elementary school teachers, 674,000 middle school teachers, and 1 million secondary school teachers employed in the U.S. In the United States, each state determines the requirements for getting a license to teach in public schools. Teaching certification generally lasts three years, but teachers can receive certificates that last as long as **ten years**. Public school teachers are required to have a bachelor's degree and the majority ...

Table 4.9. The "positive" effect of the TF-IDF approach used in BM25 are exemplified in Question 484, respectively. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results.

	Turkish	**English**
Question 484	Siliya ne için kullanılır?	What are cilia used for?
Answer	hareket yöntemi	method of locomotion
BM25	1 milimetreden (0,039 in) 1,5 metreye (4,9 ft) kadar değişen boyutlarıyla taraklılar, ana **hareket yöntemi** olarak siliya ("kıl") kullanan en büyük kolonyal olmayan hayvanlardır. Çoğu türün tarak dizisi denen ve vücutları boyunca devam eden, ktene adı verilen taraksı siliya grupları taşıyan sekiz dizisi vardır ve böylece siliya vurduğunda her bir tarak alttaki tarağa dokunur. "Ktenofor", Yunanca'da "tarak" anlamına gelen κτεις (kök biçimi κτεν-) ile "taşıyan" anlamına gelen Yunanca son ek -φορος 'tan gelir ve "tarak taşıyan" . . . Source: XQuAD-TR	Ranging from about 1 millimeter (0.039 in) to 1.5 meters (4.9 ft) in size, ctenophores are the largest non-colonial animals that use cilia ("hairs") as their main **method of locomotion**. Most species have eight strips, called comb rows, that run the length of their bodies and bear comb-like bands of cilia, called "ctenes" stacked along the comb rows so that when the cilia beat, those of each comb touch the comb below. The name "ctenophora" means "comb-bearing", from the Greek κτεις (stem-form κτεν-) meaning "comb" and the Greek suffix -φορος meaning "carrying"...
ColBERT-QA	Silikon veya polisiloksan, siloksan'dan (-R2Si-O-SiR2-, burada R = organik grup) oluşan bir polimer'dir. Bunlar genellikle renksiz yağlar veya kauçuk benzeri maddelerdir. Silikonlar, dolgu macunlarında, yapıştırıcılarda, yağlayıcılarda, tıpta, pişirme kaplarında, ısı ve elektrik yalıtımında kullanılır. Bazı yaygın biçimler arasında silikon yağı, silikon gresi, silikon kauçuk, silikon reçine ve silikon kalafat bulunur. Daha kesin olarak polimerize edilmiş siloksan'lar veya polisiloksanlar olarak adlandırılan silikonlar, her silikon merkezine bağlı iki organik gruplu inorganik silikon-oksijen omurga zinciri'nden (· · · -Si-O-Si-O-Si-O-· · · oluşur. Genellikle . . . Source: Wiki-TR-2023	Silicone or polysiloxane is a polymer composed of siloxane (-R2Si-O-SiR2-, where R = organic group). These are typically colorless oils or rubber-like substances. Silicones are used in caulks, adhesives, lubricants, medicine, cooking utensils, and for thermal and electrical insulation. Some common forms of silicones include silicone oil, silicone grease, silicone rubber, silicone resin, and silicone caulk. More precisely, silicones, also called polymerized siloxanes or polysiloxanes, consist of inorganic silicon-oxygen backbone chains with two organic groups attached to each silicon center (· · · -Si-O-Si-O-Si-O-· · ·). They are generally . . .

Another factor is the ability of ColBERT-QA to represent questions better than BM25 and thus retrieve relevant passages more accurately. Two example questions are given in Table 4.10 and Table 4.11. In the first one, both models retrieved passages related to the Amazon (region) and its surrounding countries. However, only the passage retrieved by ColBERT-QA provided a specific numerical answer to the question "how many". In contrast, the passage retrieved by BM25 consisted of the correct list of the countries without an explicit

count, posing a challenge for the reader in extracting the correct answer span. In the second example, ColBERT-QA successfully identified the information need as a Nobel Prize winner who is also a member of a university alumni, and retrieved the relevant passage. In contrast, BM25 retrieved a generic passage about universities and the Nobel Prize, missing the specific person targeted as the information need.

Table 4.10. An example showcasing how ColBERT-QA can better capture the information needs implicit in questions. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined.

	Turkish	**English**
Question 439	Amazon Havzası'nda kaç ülke bulunmaktadır?	How many nations are within the Amazon Basin?
Answer	dokuz	nine
BM25	Amazon Havzası, Güney Amerika'nın Amazon Nehri ve kolları tarafından beslenen bölümüdür. Amazon drenaj havzası 6.300.000 kilometrekare (2.400.000 sq mi) bir alanı kaplamaktadır ve bu değer Güney Amerika kıtasının yaklaşık %35,5'ini oluşturmaktadır. Havza Bolivya, Brezilya, Kolombiya, Ekvador, Fransız Guyanası (Fransa), Guyana, Peru, Surinam ve Venezuela ülkeleri sınırları içinde yer almaktadır. Havzanın çoğu Amazon yağmur ormanları ile kaplıdır. Kapladığı 55 milyon kilometrekare (21×10^6 sq mi) alan ile tropikal orman alanı, dünyanın en büyük yağmur ormanıdır. Dematteis, Lou; ... Source: Wiki-TR-2023	The Amazon basin is the part of South America drained by the Amazon River and its tributaries. The Amazon drainage basin covers an area of about 6,300,000 km2 (2,400,000 sq mi), or about 35.5 percent of the South American continent. It is located in the countries of Bolivia, Brazil, Colombia, Ecuador, Guyana, Peru, Suriname, and Venezuela, as well as the territory of French Guiana. Most of the basin is covered by the Amazon rainforest, also known as Amazonia. With a 5.5 million km2 (2.1 million sq mi) area of dense tropical forest, it is the largest rainforest in the world. Dematteis, Lou; ...
ColBERT-QA	Amazon yağmur ormanı (Portekizce: Floresta Amazônica veya Amazônia; İspanyolca: Selva Amazónica, Amazonía veya genellikle Amazonia; Fransızca: Forêt amazonienne; Hollandaca: Amazoneregenwoud) İngilizce'de aynı zamanda Amazonia veya Amazon Jungle olarak da bilinir ve Güney Amerika'nın Amazon havzasının çoğunu kaplayan bir nemli geniş yapraklı ormanıdır. Bu havza 7.000.000 kilometre karelik alanı kaplamaktadır (2.700.000 mil kare) ve bunun 5.500.000 kilometre karesi (2.100.000 mil kare) yağmur ormanıyla kaplıdır. Bu bölge **dokuz** ulusa ait toprakları içermektedir. Ormanın çoğu yağmur ormanının %60'ı ... Source: XQuAD-TR	The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva Amazónica, Amazonía or usually Amazonia; French: Forêt amazonienne; Dutch: Amazoneregenwoud), also known in English as Amazonia or the Amazon Jungle, is a moist broadleaf forest that covers most of the Amazon basin of South America. This basin encompasses 7,000,000 square kilometres (2,700,000 sq mi), of which 5,500,000 square kilometres (2,100,000 sq mi) are covered by the rainforest. This region includes territory belonging to **nine** nations. The majority of the forest is contained within Brazil, with 60% ...

Table 4.11. An example showcasing how ColBERT-QA can better capture the information needs implicit in questions. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined.

	Turkish	**English**
Question 922	Hangi Nobel Ekonomi Ödülü kazananı aynı zamanda bir üniversite mezun üyesidir?	What Nobel Memorial Prize in Economic Sciences winner is also a university alumni member?
Answer	Milton Friedman	Milton Friedman
BM25	Üniversitelerine göre Nobel Ödülü sahipleri listesi, Nobel Ödülü kazananların (öğrenci veya mezun oldukları üniversitelere göre) birinci derecede eğitim gördükleri Üniversitelere göre listelenmiş halidir. Üniversiteler Nobel Ödülü kazananların sayısına göre doğru orantılı şekilde sıralanmıştır. Üniversitelere göre listeleme işlemi oldukça kapsamlı bir çalışma gerektirmiştir. Bu nedenle çok çeşitli kaynaklardan yararlanılmıştır. Ödülü kazanan birçok kişi farklı Üniversitelere geçiş veya doktora yapmıştır. Bu nedenle bazı ödül sahipleri birden fazla Üniversitede eğitim görmüş veya doktora yapmış olabilir. Aşağıdaki listede ödül ... Source: `Wiki-TR-2023`	The list of Nobel Prize laureates according to their universities is a compilation that categorizes the winners (based on whether they were students or graduates) according to the universities where they received their primary education. The universities are ranked in proportion to the number of Nobel Prize recipients. The process of listing the universities required extensive and comprehensive research, utilizing various sources. Many prize winners have transitioned to different universities or pursued doctoral degrees, resulting in some recipients having received education or completed their doctorates at multiple universities. The list below features the recipients of the award.
ColBERT-QA	...yer alır. Amerikalı ekonomist, sosyal kuramcı, politif filozof ve yazar Thomas Sowell de üniversitenin mezunları arasındadır. Ekonomide, tanınmış Nobel Ekonomi Ödüllü, ABD'nin Cumhuriyetçi Başkanı Ronald Reagan'ın Muhafazakar Britanya Başbakanı Margaret Thatcher'ın baş danışmanlarından biri olan **Milton Friedman**, Nobel ödüllü ve düzenleme tuzağı teorisini ileri süren George Stigler, ekonominin aile ekonomisi dalına önemli katkılar sunmuş olan Gary Becker, örgütsel karar verme konseptinin modern yorumundan sorumlu Herbert A. Simon, Nobel Ekonomi Ödüllü ilk Amerikalı olan Paul Samuelson ... Source: `XQuAD-TR`	...are. American economist, social theorist, political philosopher, and author Thomas Sowell is also an alumnus. In economics, notable Nobel Memorial Prize in Economic Sciences winners **Milton Friedman**, a major advisor to Republican U.S. President Ronald Reagan and Conservative British Prime Minister Margaret Thatcher, George Stigler, Nobel laureate and proponent of regulatory capture theory, Gary Becker, an important contributor to the family economics branch of economics, Herbert A. Simon, responsible for the modern interpretation of the concept of organizational decision-making, Paul Samuelson, the first American to win the Nobel Memorial Prize in Economic Sciences ...

During manual analysis, we also observed an intriguing aspect related to ColBERT-QA's WordPiece tokenization, which can have both positive and negative implications. Table 4.12 and Table 4.13 shows two example cases. In the first example, ColBERT-QA employed WordPiece tokenization to split the word "Amazonas" into the word pieces `[''Amazon'', ''##as'']`. This split allowed ColBERT-QA to correctly associate the word "Amazonas" with the word "Amazon" and successfully retrieve the relevant passage.

On the other hand, BM25 placed excessive emphasis on the term “Amazonas” and other content words in the question due to its lexical bias, leading to the retrieval of an entirely unrelated passage that contained these content words extensively.

Table 4.12. An example that illustrates how WordPiece tokenization can produce a “favorable” outcome, depending on its ability to resist the influence of lexical bias. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results.

	Turkish	**English**
Question 430	Kaç ülke isminde “Amazonas” bulunmaktadır?	How many nations contain “Amazonas” in their names?
Answer	Dört	four
BM25	Tarık el-Tayyib Muhammed Buazizi (29 Mart 1984 - 4 Ocak 2011), Tunuslu seyyar satıcı. 17 Aralık 2010’da kendisini yakarak intihar girişiminde bulundu. Bu olayın tesiri ile Tunus halkının ayaklanması üzerine 23 yıldır ülkeyi yöneten Zeynel Abidin Bin Ali ülkeden kaçmıştır. Bu olay aynı zamanda diğer Arap ülkelerindeki ayaklanmaları teşvik etmiştir. Ölümünden sonra Tunus’ta Yasemin Devrimi başlamıştır. 17 Ocak 2011’de başkent Tunus’un en ünlü caddesi olan 7 Kasım Caddesi’nin ismi (Zeynel … Source: `Wiki-TR-2023`	Tarık el-Tayyib Muhammed Buazizi (March 29, 1984 - January 4, 2011) was a Tunisian street vendor. On December 17, 2010, he set himself on fire in a suicide attempt. As a result of this incident, Zine El Abidine Ben Ali, who had been ruling the country for 23 years, fled from Tunisia. This event also inspired uprisings in other Arab countries. Following his death, the Jasmine Revolution began in Tunisia. On January 17, 2011, the name of 7 November Avenue, the most famous street in the capital city of Tunis (Zine El …
ColBERT-QA	…ile Brezilya sınırları içindedir, ardından %13 ile Peru, %10 ile Kolombiya, ve daha az oranlarla Venezuela, Ekvador, Bolivya, Guyana, Surinam ve Fransız Guyanası gelir. **Dört** ülkenin eyalet veya il isimlerinde “Amazonas” geçmektedir. Amazon gezegenin mevcut yağmur ormanlarının yarıdan fazlasını temsil etmektedir ve dünyadaki en büyük ve en çok biyoçeşitliliğe sahip tropik yağmur ormanı alanını içermektedir ve buna 16.000 türe ayrılan 390 milyar ağaç dahildir. Amazon yağmur ormanı (Portekizce: Floresta Amazônica veya Amazônia; İspanyolca: Selva Amazónica, … Source: `XQuAD-TR`	…is contained within Brazil, followed by Peru with 13%, Colombia with 10%, and with minor amounts in Venezuela, Ecuador, Bolivia, Guyana, Suriname and French Guiana. States or departments in **four** nations contain “Amazonas” in their names. The Amazon represents over half of the planet’s remaining rainforests, and comprises the largest and most biodiverse tract of tropical rainforest in the world, with an estimated 390 billion individual trees divided into 16,000 species. The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva Amazónica …

Table 4.13. An example that illustrates how WordPiece tokenization can produce an "unfavorable" outcome, depending on its ability to resist the influence of lexical bias. The content words in the questions and the corresponding correctly matched terms in the passages are shown underlined. The dashed lines represent the incorrectly matching terms (false positives) that adversely affect the results.

	Turkish	**English**
Question 941	Huihui neydi?	What was huihui?
Answer	Müslüman tıbbı	Muslim medicine
BM25	Batı tıbbı, bazen huihui ya da **Müslüman tıbbı** olarak adlandırıldığı Yuan meclisinin Nestûrî Hristiyanları tarafından Çin'de de uygulanmıştır. Nestûrî hekim Tercüman İsa, 1963 yılında, Kubilay'ın saltanatı döneminde Batı Tıbbı Ofisini kurmuştur. İki imparatorluk hastanesinde çalışan doktorlar imparatorluk ailesi ve meclisin üyelerini tedavi etmekten sorumluydu. Çinli hekimler, hümoral sistemi, geleneksel Çin tıbbının altında yatan yin-yang ve wuxing felsefesine karşı geldiği için Batı tıbbına karşı çıkıyorlardı. Batı tıbbı çalışmalarının bilinen bir Çin tercümesi yoktur ama Çinlilerin İbn-i . . . Source: XQuAD-TR	Western medicine was also practiced in China by the Nestorian Christians of the Yuan court, where it was sometimes labeled as huihui or **Muslim medicine**. The Nestorian physician Jesus the Interpreter founded the Office of Western Medicine in 1263 during the reign of Kublai. Huihui doctors staffed at two imperial hospitals were responsible for treating the imperial family and members of the court. Chinese physicians opposed Western medicine because its humoral system contradicted the yin-yang and wuxing philosophy underlying traditional Chinese medicine. No Chinese translation of Western medical works is known, but Chinese had Avicenna . . .
ColBERT-QA	Tüzükleri, işbirliği yapmayan çocuk işçiler için hapis şartlarını öngörmüştür. Hong Kong gibi güneydoğu Asya kolonilerinde Mui Tsai () gibi çocuk işçiliği kültürel bir gelenek olarak rasyonelleştirildi ve İngiliz yetkililer tarafından göz ardı edildi. Hollanda Doğu Hindistan Şirketi yetkilileri, çocuklarının işçi tacizlerini "bu, bu çocukları daha kötü bir kaderden kurtarmanın bir yolu" ile mantıklı hale getirdiler. Zambiya'dan Nijerya'ya uzanan bölgelerdeki Hıristiyan misyon okulları da çocuklardan çalışma gerektirdi ve karşılığında laik eğitim değil din eğitimi sağladı. Başka . . . Source: SQuAD-TR-TRAIN	In southeast Asian colonies, such as Hong Kong, child labour such as the Mui Tsai (), was rationalised as a cultural tradition and ignored by British authorities. The Dutch East India Company officials rationalised their child labour abuses with, "it is a way to save these children from a worse fate." Christian mission schools in regions stretching from Zambia to Nigeria too required work from children, and in exchange provided religious education, not secular education. Elsewhere . . .

However, WordPiece tokenization can be a liability as well. In the second example, despite the word “Huihui” being a proper noun, ColBERT-QA tokenized it as [‘‘Hu’’, ‘‘##ih’’, ‘‘##u’’, ‘‘##i’’], resulting in retrieving an irrelevant passage. The same effect can also be seen in Question 484 in Table 4.9, where ColBERT-QA matched the words “Silikon” [‘‘Sili’’, ‘‘##kon’’] and “Siliya” [‘‘Sili’’, ‘‘##ya’’] due to their common prefix and incorrectly retrieved a passage on “Silikon” (Silicone) for a question about “Siliya” (Cilia), which are completely different concepts.

Table 4.14. Example passages chosen by each retriever for a given question from SQuAD-TR-TRAIN, and put into the training set of the corresponding readers.

	Turkish	**English**
Question 178	2009'da Beyonce ikinci dünya turuna başladı ve ne kadar hasılat etti?	In 2009, Beyonce started her second world tour and grossed how much money?
Answer	119,5 milyon	119.5 million
BM25	The Beyoncé Experience, Amerikalı şarkıcı Beyoncé'nin üçüncü konser turnesi. Beyoncé, 4 Eylül 2006'da ikinci solo albümü B'Day'i yayımlamıştı. Albümün getirdiği başarı onu ilk dünya turnesine götürdü. Turnenin adının "B'Day World Tour" olması düşünülmüştü ancak Beyonce'nin ilk dünya turu deneyimi olduğu için "The Beyonce Experience Tour' olmasına karar verildi. Sony Music stüdyolarında gerekli çalışmalar, provalar, dansçı seçimleri, koreografiler, sahne dekoru ayarlamaları yapıldıktan sonra Mart'ta genel hazırlıklar başladı. Normalde Avrupa'da başlayacak olan turne bazı aksaklıklar yüzünden Japonya'dan ... Source: Wiki-TR-2023	The Beyoncé Experience is the third concert tour by American singer Beyoncé. Beyoncé had released her second solo album, B'Day, on September 4, 2006. The success of the album led her to embark on her first world tour. The tour was initially planned to be called the "B'Day World Tour," but it was decided to name it "The Beyonce Experience Tour" as it was Beyoncé's first experience with a world tour. After necessary work, rehearsals, dancer selections, choreography, and stage set adjustments were made at Sony Music studios, general preparations began in March. The tour, which was originally planned to start in Europe, (began) from Japan due to some setbacks. ...
DPR	...Kadın Video kategorisini kazanamaması, Kanye West'in töreni kesintiye uğratmasına ve Beyoncé'nin kendi kabul konuşması sırasında Swift'in ödülünü yeniden sunumunu gerçekleştirmesine yol açtı. Mart 2009'da, Beyoncé I... Dünya Turu, 108 gösteriden oluşan dünya çapında konser turu, **119,5 milyon** dolar hasılat kazanıyor. 4 Nisan 2008'de, Beyoncé Jay Z ile evlendi. Üçüncü stüdyo albümü olan I Am'in dinleme partisinde bir video montajında evliliklerini açıkça açıkladı. Sasha Fierce, 22 Ekim 2008'de Manhattan'ın Sony Kulübünde. Ben... Sasha Fierce 18 Kasım ... Source: SQuAD-TR-TRAIN	...Not winning in the Female Video category led to Kanye West interrupting the ceremony and Beyoncé re-presenting Swift's award during her acceptance speech. In March 2009, Beyoncé's I... World Tour, a worldwide concert tour consisting of 108 shows, grossing **$119.5 million** in revenue. On April 4, 2008, Beyoncé married Jay Z. They openly announced their marriage in a video montage at the listening party for her third studio album, I Am... Sasha Fierce, on October 22, 2008, at Manhattan's Sony Club. I... Sasha Fierce November 18th. ...
ColBERT-QA	...Amerikalı şarkıcı Taylor Swift'in "You Belong with Me" şarkısının klibine giden En İyi Kadın Klibi kategorisinde ödül alamaması, Kanye West'in töreni durdurmasına ve Beyoncé'nin kendi ödül konuşmasını Swift'e vermesine yol açtı. Mart 2009'da *Beyonce*, 108 gösteriden oluşan ve $**119,5** **milyon** *hasılat* *yapan* *ikinci* *dunya* turnesi I Am... World *Tour*'u başlattı. Beyoncé, 2008 yapımı müzikal biyografik film Cadillac Records'ta blues şarkıcısı Etta James olarak başrolde yer alarak filmlerde rol almaya devam etti. Filmdeki performansı eleştirmenler tarafından ... Source: Wiki-TR-2023	...The failure of American singer Taylor Swift to win the Best Female Video category for her song "You Belong with Me" led to Kanye West interrupting the ceremony and Beyoncé giving her own award speech to Swift. In March 2009, *Beyonce* launched her *second* *world* *tour*, I Am... World tour, consisting of 108 shows and *grossing* $**119.5** **million** . Beyoncé continued to act in films, starring as blues singer Etta James in the 2008 musical biographical film Cadillac Records. Her performance in the film was praised by critics. ...

To further deepen this analysis, Table 4.14 shows sample passages selected by each retriever for a particular question sourced from SQuAD-TR-TRAIN. These passages are incorporated into the training set of the respective readers.

In this particular example, we see that BM25 fails to identify the relevant passage containing the answer for the question due to its lexical nature: the incorrect passage it selected contains numerous repetitions of one of the key content words in the query.

To analyze DPR and ColBERT-QA, we used BertViz [186] as a tool to visualize the output of BERT-based models.[34] When DPR encodes the question in Table 4.14, we see that the [CLS] token in the output layer is attending to the [CLS] token in the preceeding layer. This behavior continues across the other layers until the earlier layers, where the multi-head attention is distributed across all the words but focuses relatively more on the content words "Beyoncé", "ne kadar (how much)", "hasılat etti (grossing)" in certain heads. When DPR encodes the passage, a similar pattern is observed for the [CLS] token, where the attention of the [CLS] token in the earlier layers is spread across all tokens in the passage with loose emphasis on certain content words like "Beyoncé, dünya turu (world tour)", "hasılat (gross)", as underlined in the passage for DPR.

For ColBERT-QA, we also show in the table the pairs of question-passage tokens matched by means of MaxSim scoring in the passage. ColBERT-QA establishes a balance between lexical matching and semantic matching. For example, it is able to match certain question words such as "2009", "Beyoncé", "hasılat (gross)", "ikinci (second)", "dünya (world)" with the corresponding words in the paragraph in an exact manner, just like BM25 does. On the other hand, it can also match certain words and phrases such as "ne kadar (how much)" and "etti" (akin to "made") with the semantically related words and phrases "milyon (million)" and "yapan" (akin to "doing"), respectively, demonstrating semantic association. Unlike BM25, ColBERT-QA is unaffected by the excessive presence of a question token in the passage, as it only selects one passage token that is most similar to the question token. This

[34] The visual representations are omitted from the thesis due to concerns about their readability within the constraints of page size. However, we provide descriptions of the key insights from the visualizations in the text and make the full-sized visualizations available on our Github page: https://github.com/boun-tabi/SQuAD-TR

observation suggests a potential avenue for future research: modifying ColBERT-QA to allow the selection of multiple passage tokens for each question token, thereby viewing each scoring as a mini ranking function that operates on the tokens within a passage given a question token. We leave this exploration for a future work.

The example presented in Table 4.14 also demonstrates how the retriever models function in scenarios where potential noise is introduced by machine translation. In the question, the phrase "hasılat etti" is a flawed translation used in lieu of the correct translation "hasılat yaptı" for the English term "grossed". This discrepancy arises from the translation system's confusion between the words "etti" (akin to "made") and "yaptı" (akin to "did") owing to the difference between the use of these words in Turkish and English. For this question, DPR returns the target positive passage sourced from SQuAD-TR-TRAIN, which is a machine-translated text, while ColBERT-QA retrieves the equivalent passage from Wikipedia. BM25 returns a negative passage from Wikipedia due to its lexical term bias.

The passage returned by DPR has the expression "Beyoncé I... Dünya Turu" ("Beyoncé I... World Tour"), which is an erroneous translation. The correct translation for the original expression "Beyoncé I Am... World Tour" should have been "Beyoncé I Am... Dünya Turu", as seen in the corresponding Wikipedia passage retrieved by ColBERT-QA. In this particular example, we understand that each model handles the noise in the translated texts in a different way. BM25 completely disregards the translation error in the question term because it does not find a matching term in the retrieved passage. DPR successfully retrieves the target passage from SQuAD-TR-TRAIN as its fixed-size representations align with those of the question, without being affected by the translation error that exists in both the question and the passage. Meanwhile, ColBERT-QA retrieves the original Wikipedia passage by associating each question token with the most semantically relevant passage token, even in cases where the question token was inaccurately translated. For instance, the question token "etti" (akin to "made") was matched with the closest semantically related passage token "yapan" (akin to "doing"). In this respect, ColBERT-QA surpasses DPR in the quality of passages returned, by effectively managing the individual interactions between question and passage tokens, as seen in BM25. Furthermore, it also outperforms BM25 by leveraging its distributional

representation capability to match semantically related tokens.

4.3.2.2. Reader Results. Table 4.15 shows the results of the reader step of the OpenQA formulation. The results demonstrate that the reader trained on the dataset obtained by the ColBERT-QA [12] retriever using `Wiki-TR-2021` achieves around 27% (EM) / 23% (F1) improvement and around 26% (EM) / 22% (F1) improvement compared to the readers that use the baseline BM25 and DPR retrievers, respectively. This improvement changes to around 24% (EM) / 26% (F1) for BM25 and to approximately 33% (EM) / 29% (F1) for DPR when the retrievers use `Wiki-TR-2023`. Based on these findings, we hypothesize that the substantial degradation in the quality of the bootstrap training datasets for BM25 and DPR relative to ColBERT-QA, as the knowledge source grows over time, is the source of the increasing gap between the baseline models and ColBERT-QA. Similarly, the higher quality training dataset generated by ColBERT-QA for its reader, as the knowledge-source expands, contributes to narrowing the gap between the ColBERT-QA-based reader towards the upper bound standard QA reader results shown in Table 4.4. This is a striking finding that underscores the capacity of the retriever models to withstand the gradual expansion of the knowledge source over time.

Table 4.15. Reader results for the OpenQA formulation of QA task with respect to each of the retriever.

Reader Model	**Retriever Model**	**Training Dataset**	**Test Dataset**	**EM**	**F1**
XLM-RoBERTa	BM25 - Baseline (Sparse)	`SQuAD-TR-TRAIN (BM25,2021)`	`XQuAD-TR (BM25,2021)`	37.82	49.96
		`SQuAD-TR-TRAIN (BM25,2023)`	`XQuAD-TR (BM25,2023)`	37.31	48.59
XLM-RoBERTa	DPR - Baseline (Dense)	`SQuAD-TR-TRAIN (BM25,2021)`	`XQuAD-TR (BM25,2021)`	38.07	50.40
		`SQuAD-TR-TRAIN (BM25,2023)`	`XQuAD-TR (BM25,2023)`	34.96	47.36
XLM-RoBERTa	ColBERT-QA	`SQuAD-TR-TRAIN (ColBERT-QA,2021)`	`XQuAD-TR (ColBERT-QA,2021)`	46.47	61.22
		`SQuAD-TR-TRAIN (ColBERT-QA,2023)`	`XQuAD-TR (ColBERT-QA,2023)`	**47.98**	**61.63**

To further explore the effects of expanding the knowledge source, we inspected the reader outputs in relation to the source of the passages. Our aim in this analysis is to determine if the number of passages retrieved from the knowledge source increases or not as the knowledge source expands, and how this impacts the performance of the readers.

Table 4.16. Error analysis on the reader outputs on XQuAD-TR with respect to the source of the passages (SQuAD-TR or Wiki-TR) and the retriever module the reader is based on.

	Wiki-TR-2021						Wiki-TR-2023					
	SQuAD-TR			Wiki-TR			SQuAD-TR			Wiki-TR		
Base of Reader Model	**Correct**	**Incorrect**	**Subtotal**	**Correct**	**Incorrect**	**Subtotal**	**Correct**	**Incorrect**	**Subtotal**	**Correct**	**Incorrect**	**Subtotal**
BM25-based - Baseline (Sparse)	432	452	883	19	288	307	424	442	866	20	304	324
DPR-based - Baseline (Sparse)	441	506	947	12	231	243	409	514	923	7	260	267
ColBERT-QA-based	544	598	1142	9	39	48	559	560	1119	12	59	71

Table 4.16 shows the outputs of each reader categorized by the source of passages selected from Wiki-TR-2021 and Wiki-TR-2023. The BM25-based reader includes about 26% of passages from Wikipedia in their test datasets when using Wiki-TR-2021 and 27% when using Wiki-TR-2023. For DPR-based readers, these numbers are 20% for Wiki-TR-2021 and 22% for Wiki-TR-2023. However, for ColBERT-QA, these numbers are much lower: 4% from Wiki-TR-2021 and 6% from Wiki-TR-2023. When passages are from Wikipedia, the success rates of these readers are somewhat variable. For the BM25-based readers, the success rate is around 6% for both Wiki-TR-2021 and Wiki-TR-2023. For DPR-based readers, these numbers are 5% and 3%, respectively. For ColBERT-QA-based readers, they are 18% and 17%. Conversely, when passages are originated from XQuAD-TR, the success rates improve for all readers. Specifically, BM25-based readers achieve a success rate of 49% for both Wiki-TR-2021 and Wiki-TR-2023. DPR-based readers achieve up to 47% for Wiki-TR-2021 and 44% for Wiki-TR-2023. For ColBERT-QA, these figures reach 48% when using Wiki-TR-2021 and 50% when using Wiki-TR-2023, respectively.[35] Although success rates decline for passages retrieved from Wikipedia for all readers, only ColBERT-QA compensates for this drop with a larger increase in success rates for the examples where passages are from XQuAD-TR. These numbers explain why reader performance varies as the knowledge source expands. They support our findings from the retriever output that ColBERT-QA is not only a noise-resistant model, but it can also improve its performance as the noise in the knowledge source increases.

In addition, the OpenQA model for the ColBERT-QA-based reader achieves almost

[35] Table 4.8, Table 4.9, Table 4.10, Table 4.11, Table 4.12 and Table 4.13 provide specific examples illustrating the role of Wikipedia passages as distractors.

89% of the standard formulation QA reader results in terms of both EM and F1 scores. This result suggests that the OpenQA formulation is productive for low-resource and resource-constrained languages, since we can rely on machine-generated noisy training data and unstructured knowledge sources.

4.3.2.3. OpenQA Results with Subsampled Test Sets. Figure 4.3 summarizes the results of the experiments we conducted using subsampled datasets for our retrievers, and Figure 4.4 extends the protocols to our readers. In each panel, the x-axis tracks the number of assessment examples, and the y-axis shows our key metrics.

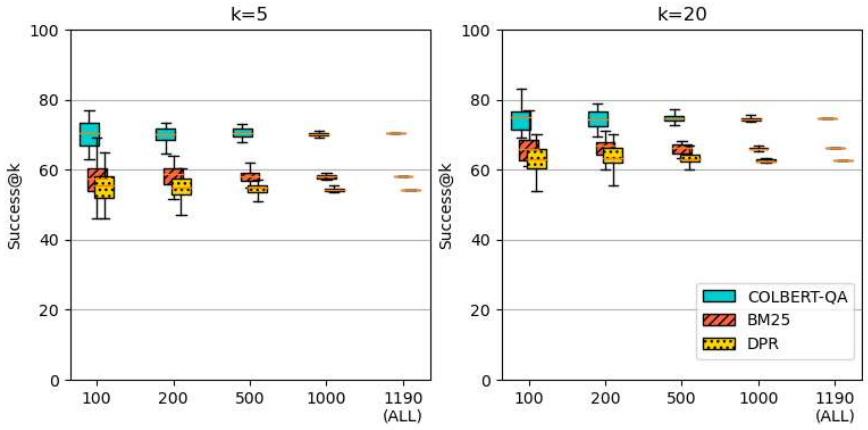


Figure 4.3. Retriever results for the OpenQA formulation for different sample portions of XQuAD-TR based on Wiki-TR-2023.

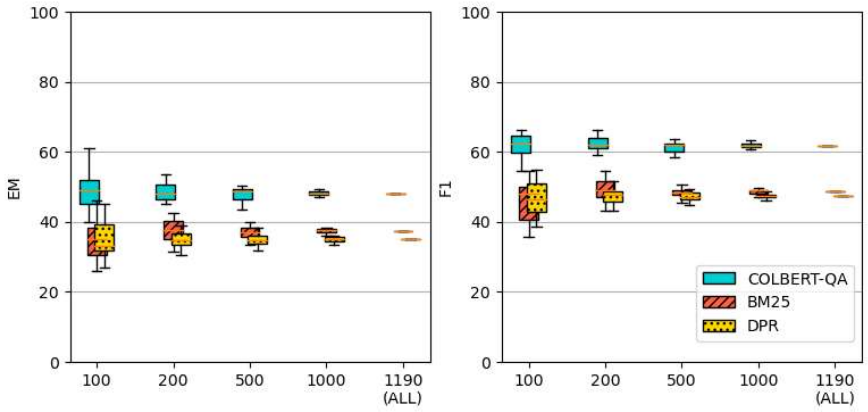


Figure 4.4. Reader results for the OpenQA formulation for different sample portions of XQuAD-TR based on Wiki-TR-2023.

Strikingly, with only 100 examples, we can already pretty clearly differentiate our BM25-based and DPR-based models from our ColBERT-QA-based models. By 200 examples, the systems are dramatically different on all metrics for BM25 and ColBERT-QA. As the test sets get larger, the variance of these measures gets tighter, as one would expect, but the core conclusions are unchanged beyond 200 examples for the models. The results of the BM25 and DPR models implicitly suggest that the number of examples needed to differentiate the benchmarked models would increase proportional to the competitiveness of the models with respect to each other.

In this setting, we are using the experiments to differentiate three systems, but the same logic would apply if we were seeking to determine whether a system had truly passed a lower-bound on performance that we set for a production system. Overall, these results show that OpenQA systems can be evaluated very efficiently. This opens the door to conducting multiple distinct evaluations of the same system, which could be crucial for piecing together a picture of how the system behaves overall.

4.3.3. Resource Requirements of the Models

When aiming to enhance the performance of the models, it is important to also account for their resource consumption, particularly when working with use cases that have limited resources. In this section, we present the resource footprints of the models we utilized in our study. The values are based on the model parameters specified in Section 4.2.4. Varying the model parameters and processing units such as batch size, query and document lengths, and parallel processing or multi-threading can lead to different results.

4.3.3.1. Resource Usage of Retriever Models. Table 4.17 shows the resource usage of the retriever models for each stage of the retrieval process. We observe that BM25 exhibits the lowest resource footprint across nearly all categories in each phase, offering the fastest indexing and retrieval speeds. In contrast, dense retrievers demonstrate varying resource consumption across different phases.

Table 4.17. Resource consumption of retriever models during training, indexing, and retrieval steps.

	Resource	BM25	DPR	ColBERT-QA
Training	*Num of CPU*	-	1.50	1.17
	Memory	-	13.42 GB	6.14 GB
	Model size	-	2.5 GB	1.3 GB
	Duration	-	6 hrs 13 mins	8 hrs 30 mins
Indexing	*Num of CPU*	1.27	1.07	2.76
	Memory	2.57 GB	19.90 GB	19.22 GB
	Index size	2 GB	6.5 GB	114 GB
	Speed (D/s)[*]	2610 D/s	315 D/s	619 D/s
	Duration	14 mins	116 mins	59 mins
Retrieval/Ranking	*Num of CPU*	1.01	3.59	1.73
	Memory	1.71 GB	54.41 GB	78.60 GB
	Speed (Q/s)[**]	35.77 D/s	16.30 D/s	5.49 D/s
	Duration	41 mins	90 mins	4 hrs 7 mins

[*] D/s: The number of documents indexed per second.

[**] Q/s: The number of questions the retriever can process per second.

In training, DPR uses more memory than ColBERT-QA, mainly because DPR depends on in-batch negatives and its performance is tied to batch size. Additionally, DPR's final model size is twice as large as that of ColBERT-QA because DPR uses two separate BERT models for query and document encoding. In contrast, ColBERT-QA employs a single shared BERT model for both encoders.

During indexing, BM25 demonstrates exceptional speed when indexing the knowledge source compared to dense retrievers. It also maintains a low memory footprint and minimal index size. DPR and ColBERT-QA have a similar memory footprint. DPR exhibits slower indexing performance relative to ColBERT-QA. (For all models, indexing speed can be boosted using multiple parallel processes.) Additionally, DPR produces a smaller index than ColBERT-QA, in line with the fact that DPR generates fixed-size vectors while ColBERT-QA produces a sequence of token embeddings.

In the retrieval and ranking step, BM25 maintains the minimum footprint across all categories while achieving the fastest speed. DPR and ColBERT-QA, on the other hand, both demand a considerable amount of memory because they load the entire index into memory during retrieval. The size of memory that DPR and ColBERT-QA consume is in line with the size of their index. Meanwhile, BM25 minimizes memory usage by accessing the index directly from disk.

4.3.3.2. Resource Usage of Reader Models. Table 4.18 outlines the resource footprints of the reader models in our study, highlighting their impact and trade-offs across different types of resources. The models were trained using GPUs and then tested exclusively on CPUs to mirror the resource demands of a typical real-world environment.

Table 4.18. Resource consumption of reader models during training and test steps categorized by the architecture of the model.

	Resource	**BERT-based**	**RoBERTa-based**
Train	*Num of CPU*	0.99	1
	Memory	11.05 GB	14.87 GB
	Model size	420 MB	1.1 GB
	Duration	3 hrs 24 mins	12 hrs 38 mins
Test	*Num of CPU*	3.89	3.93
	Memory	1 GB	2.79 GB
	Speed (Q/s)[*]	2.08 Q/s	2.07 Q/s
	Duration	9 mins	9 mins

[*] Q/s: The number of questions the readers can process per second.

The RoBERTa-based model takes approximately three times longer to train compared to BERT-based readers, and its resulting model size is significantly larger than that of BERT-based models. This is largely due to the inherent difference in the number of parameters between the two types of models. Nonetheless, the inference speeds of the models when

tested solely on CPUs are fairly similar, though the RoBERTa-based reader attains this speed only by using around three times the memory of BERT-base models. The inference speed also indicates that each CPU core can handle about two questions per second for these readers.

4.4. Discussion

In this section, we discuss the findings of our experiments to assess the effectiveness and limitations of the proposed approach. We begin by analyzing the results of the standard QA formulation in order to observe the potential performance for OpenQA. Then, we delve into the results for each component of the OpenQA formulation, namely, the retriever and reader modules. Finally, we discuss the required number of gold instances for a reliable evaluation of an OpenQA system.

Perhaps unsurprisingly, one of the best models in the standard QA results is the one that begins with BERTurk parameters and trains on our in-language dataset. However, using multilingual embeddings (mBERT) is competitive with this, and the only large gap in performance is between these two models and the mBERT model trained on English data. This aligns with other recent findings and shows that in-language training data is the real differentiator, even if it is potentially noisy MT data. Furthermore, XLM-RoBERTa outperforms BERTurk, highlighting the impact architectural improvements can have on performance. These results also point out the potential for developing in-language variants of RoBERTa, which can outperform XLM-RoBERTa, as observed in other languages [187, 188, 189].

The success of OpenQA systems highly depends on the performance of their retriever models. The baseline BM25 and DPR retrievers were able to capture at most, respectively, 56.30% and 52.10% of the target passages, whereas the ColBERT-QA retriever increased this to 77.05%. As such, the ColBERT-QA retriever model increased the likelihood of the reader model finding more relevant passages to answer questions. These results also indicate that there is a room for further improvement within the retrieval module to find more passages containing relevant answer spans for the questions.

Furthermore, the loss of some answer span locations within the target paragraphs after translation does not necessarily hinder the performance of the retriever models; in fact, weak supervision presents an advantage in such cases. This is because retrievers can instead fetch other suitable paragraphs from the knowledge source, resulting in noise-resistant retriever results. Consequently, this increases the number of examples with positive passages in training bootstrap datasets for the retriever models and training/test datasets for the reader models. The noise-robustness of retrievers in the weak supervision setting also means that we can rely on datasets of question–answer pairs, with no need to include or annotate passages. The superior performance of DPR- and ColBERT-QA-based reader models over the BM25-based reader model, particularly when employing `Wiki-TR-2021`, underscores how much weak supervision facilitates retriever models in benefiting from this augmentation more effectively. Notably, the substantial and consistent improvement in the ColBERT-QA-based reader model is evident both with `Wiki-TR-2021` and `Wiki-TR-2023`, and it indicates the superior capacity of ColBERT-QA to derive benefits from weak supervision compared to DPR.

Additionally, we note that expanding the knowledge source can potentially impact the performance of the OpenQA system in two ways. It may either enhance or degrade the performance of the OpenQA system, depending on the capacity of the retriever model to effectively navigate the increased noise in the expanded knowledge source to identify more positive passages. We observe that ColBERT-QA is more capable of navigating this challenge compared to BM25 and DPR models. Consequently, as the knowledge source gradually expands over time, ColBERT-QA leads to a significant enhancement in the overall success of its reader towards its upper limit capped by the standard QA reader results.

In the standard QA formulation, the reader models achieved a maximum EM score of 52.18%, which sets the upper bound for the OpenQA reader models. The ColBERT-QA based OpenQA reader results demonstrated that we can preserve almost 89% of this score without requiring gold passages as input. This relaxation of the input requirement provides a significant advantage for developing cost-effective QA systems in low-resource language contexts.

Although we do not require input gold passages at training time, we still need a labeled dataset at test time. Our experiments revealed that a few hundred evaluation examples may be enough to confidently differentiate the performance of models. Lowering the requirement for the number of gold examples is highly important, especially for low-resource language scenarios where the necessary human resources are not widely available, e.g. endangered Indigenous Languages [69, 72]. In other language scenarios, this result not only helps limit the cost of obtaining QA systems but also paves the way for obtaining multiple evaluation datasets with different characteristics to better reflect the overall picture of the models in production.

In summary, our proposed system does not require gold datasets during training, but instead utilizes existing unstructured knowledge sources and MT systems to create a machine-translated labeled training dataset for an OpenQA application in Turkish as a use case for low-resource language contexts. The potential noise in the resulting training dataset can be overcome with the help of the weak supervision used in the OpenQA formulation, which is resilient to noisy data. As a result, we have shown that a cost-effective QA system is feasible for low-resource language contexts when we shift our focus to the OpenQA formulation.

4.5. Adaptation and Generalization: Extending Our Method to Diverse Language Contexts

We presented a general purpose method for training OpenQA in "low resource contexts" and use Turkish QA as a case study for showing its efficacy. In this section, we detail the steps for adapting our method to different low-resource contexts other than the presented approach for Turkish. A summary of our overall method is shared in Table 4.19.

Table 4.19. Our general method for creating OpenQA systems in low-resource languages efficiently.

(i) **Building a QA dataset in a new language context.** Create an extractive reading comprehension dataset in the target language context as follows:
   - **Training dataset:** Translate an existing extractive QA dataset using automatic translation.
   - **Evaluation dataset:** Obtain an evaluation dataset containing as few as 200 question–answer pairs.

(ii) **Compiling a knowledge source.** Create a knowledge source by compiling passages answering the questions in the extractive QA dataset obtained in the previous step.

(iii) **Training an OpenQA retriever.** Train a neural retriever weakly supervised by the training bootstrap dataset derived from the training dataset and knowledge source prepared in the previous steps.

(iv) **Creating an OpenQA reader.** Train an off-the-shelf reader using the extractive QA training dataset, where the contexts are now provided by the OpenQA retriever.

Step 1. Building a QA dataset in a new language context: The main blocker for building OpenQA systems in low-resource contexts is the lack of a QA dataset itself. In our case study, we resolve this issue for Turkish by translating a readily available extractive QA dataset in English to Turkish using an automatic translation service. Similar strategies can be applied to different QA datasets for other languages given the availability of the translation tools between the original and the target languages.

When obtaining a QA dataset through machine translation, one can translate the contexts, questions, and answer spans contained in a typical extractive dataset to the target language. However, the standalone translation of some of the answer spans may be different from their corresponding translations in the context paragraphs, due to the contextual nature of translation. In our case study with Turkish, we employ approximate string matching as a post-processing step to recover some of these answer spans, taking into account the morphological complexity of the Turkish language. When adapting our method to a different target language,

one can devise different post-processing methods considering the linguistic characteristics of the target language. For example, a basic stemming and lemmatization algorithm could also be used for languages that have fewer inflectional forms than Turkish. Similarly, accent normalization could also be helpful as a post-processing step for languages with accent-rich alphabets, like Vietnamese [190, 191].

At the end of this process, we successfully obtained a QA dataset in the target language, which we used as our training data. In our case study, this training dataset included 81K examples, for which 61K had their answer spans recovered. The size of our training dataset can be used as a reference when creating training datasets for other low-resource contexts, whether or not translation tools are utilized.

Although we can bootstrap our training dataset using a translation service, we need a high quality evaluation dataset for reliable assessment. In our method, we utilize an external human-annotated dataset, XQuAD [3], that supports 10 other languages. Similar datasets can also be employed for different language (e.g. XQA [183]; MLQA [13]; MKQA [192]) and domain scenarios (e.g. e-commerce [161], medical [162, 163], legal [164], finance [165], customer service [166], space [167]). Nevertheless, the availability of such datasets remains limited and may lack support for the targeted low-resource context. In such scenarios, our analysis showcased in Section 4.3.2.3 suggests that a dataset consisting of merely 200 examples can still provide valuable insights.

Step 2. Compiling a knowledge source: We then create a knowledge source that will serve as the corpus for OpenQA in our target context. We used Wikipedia as our knowledge source but other knowledge sources can also be used depending on the use case and the availability of resources in the target low resource context.

One specific limitation we encountered when using Turkish Wikipedia was that the majority of target paragraphs from our QA datasets were missing in Turkish Wikipedia. To tackle this issue, we augmented our knowledge source by adding the paragraphs from our QA dataset. This allowed us to create an OpenQA model even under limited resources. Such an

approach can be applied to other scenarios with comparable limitations.

In order to optimize the performance of the retriever models, we segment the passages in the resulting knowledge source into smaller chunks, ensuring that the number of tokens per chunk fits into the context of the retriever models. While 100 words proved sufficient for English, we found it necessary to reduce this to 75 words for Turkish due to the tokenizer for Turkish generating longer token sequences. Likewise, the chunk size can be adjusted based on the linguistic characteristics of the target language and the requirements of the chosen retriever.

Step 3. Training an OpenQA retriever: We use BM25 as a lightweight retriever to prepare the bootstrap training datasets for the advanced retrievers. We customize the BM25 retriever by incorporating a morphological stemmer tailored for Turkish. The default or custom analyzers for other languages can be employed for similar adaptations [193]. When training the advanced retrievers in our OpenQA system, we used BERTurk [140] to customize the tokenizers of our chosen retrievers and for weight initialization, thereby customizing them for Turkish. Likewise, for other languages, comparable in-language BERT variants can be utilized for adaptation. In cases where an in-language BERT model is not accessible for the desired target language, the multilingual BERT model [3] offers a viable alternative.

Step 4. Creating an OpenQA reader: The last step of our method is training an OpenQA reader using our retriever along with our extractive QA training dataset. To do so, we take XLM-RoBERTa [65] as an off-the-shelf model in our target language, which showed superior performance in our standard QA experiments. Other models, such as in-language variants of BERT [25], ALBERT [194] and RoBERTa [174], can be also selected depending on the availability of resources and needs of the target language context. We fine-tune the reader in an extractive QA setting using the training dataset obtained from the output of the retrievers employed in our experiments.

4.6. Summary

In this chapter, we obtained an affirmative answer to our core research question, Can we develop cost-effective OpenQA systems for low-resource language contexts without requiring a gold training dataset? We further expanded our question to explore the minimum test set sizes required to reliably evaluate the performance of OpenQA models. Our findings help pave the way to transferring the rapid advancements made in English QA to non-English QA systems. Moreover, this new avenue not only allows one-way transfer of advancements, but also establishes a virtuous cycle between English OpenQA and non-English OpenQA systems, promoting mutual progress. Furthermore, the proposed methodology can also benefit certain domain-specific scenarios, even in high-resource languages like English, where data remains scarce.

We presented a general method for creating efficient and effective OpenQA systems for low-resource language contexts, and we illustrated the method with a case study of Turkish. Our overall method is summarized in Table 4.19 as a recipe. As part of this, we introduced SQuAD-TR, a Turkish QA dataset derived by automatically translating SQuAD2.0 [50]. We showed that SQuAD-TR can straightforwardly be used to train high quality OpenQA systems and benchmark different types of models, and we supported this assessment with detailed qualitative analysis. In addition, we provided evidence that the success of the OpenQA system is notably enhanced by expanding the knowledge source depending on the retriever's capability on navigating the potential increase in noise accompanying the knowledge source expansion, and we showed that a relatively small number of gold test cases may be sufficient to obtain confident assessments of the quality of these systems.

The key to creating these systems in non-English languages so efficiently is the move from standard QA to OpenQA. In doing this, we greatly simplify the process of creating gold examples, which has been a barrier for the advancement in QA systems for low-resource languages. In OpenQA, these datasets are just question–answer pairs, completely eliminating the necessity for answer span annotation. Consequently, these datasets can now be acquired through automatic translation from the abundant resources available in English. The OpenQA

task is also arguably more “relevant”, in that it comes much closer than standard QA to simulating the experience of searching a real-world knowledge store like the Web. Thus, we hope not only to have removed obstacles to creating QA systems for low-resource languages like Turkish, but we also hope to have helped motivate the OpenQA task more generally, as a step towards QA systems that can truly meet the information needs of real-world users. We publicly share our code, models, and data to encourage future research.

5. BİRİ: CONVERSATIONAL OPEN QUESTION ANSWERING FOR RESOURCE-CONSTRAINED LANGUAGES

5.1. Introduction

Effective Conversational QA and Conversational OpenQA systems require large scale QA datasets created in a conversational setting. Recently, researchers shared a few large scale datasets suitable to fuel advancements in Conversational QA [104, 125] as well as Conversational OpenQA [103]. However, these datasets are limited to the English language impeding progress in other languages, particularly those with low resources. Budur et al. [2] recently shared a cost-effective method mitigating the data scarcity issue when building OpenQA systems in low-resource languages. They shared `SQuAD-TR`, a dataset derived from SQuAD2.0 [50] through machine translation, and used `XQuAD-TR` for evaluation, a human translated version of the XQuAD dataset sourced from SQuAD1.0 [8]. While their approach demonstrated the possibility of constructing OpenQA systems with limited data resources, it still relies on single-turn information seeking queries, overlooking the necessity for multi-turn interactions.

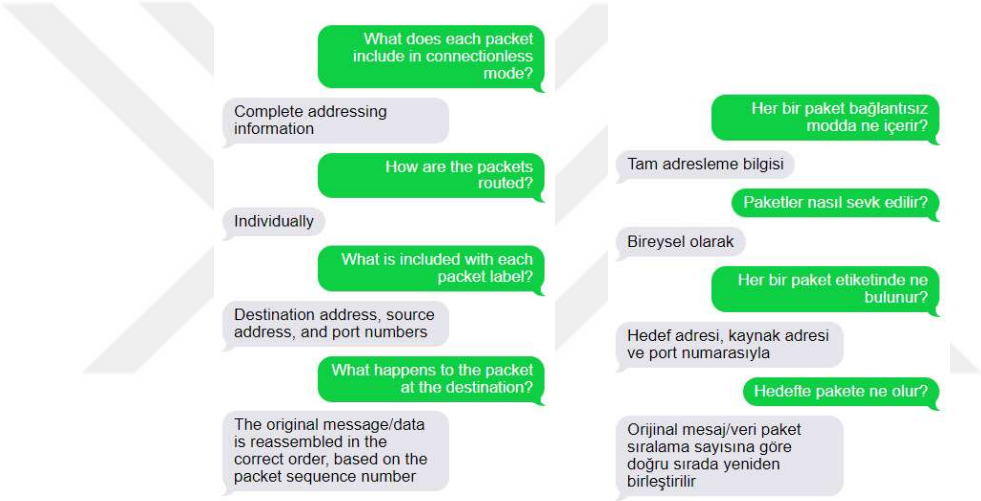
For an OpenQA system to effectively handle information seeking questions in a multi-turn interaction, it requires to retain information from preceding turns. To highlight this requirement, Table 5.1 shows a parallel instance from the XQuAD dataset in both English and Turkish. In this example, the term “packet” specifically denotes “a network packet”, but in an alternate context it could also refer to “a parcel”. While having content words like “connectionless mode” in the initial question aids in disambiguating the meaning of the term “packet”, subsequent questions lack such kind of strongly disambiguating words. This makes it challenging to locate answers for the subsequent questions, particularly in systems like OpenQA where the context is not provided.

Table 5.1. A sample entry in the XQuAD dataset where A_k denotes the answer span for the Question k in the corresponding language. In this example, the first question disambiguates the context of the term "packet" by adding content words like "connectionless mode". However, subsequent questions lack these contextual words, posing a challenge in answering those questions without the paragraph given as input.

	English	**Turkish**
Paragraph	In connectionless mode each packet includes ⟨complete addressing information⟩$_{A1}$. The packets are routed ⟨individually⟩$_{A2}$, sometimes resulting in different paths and out-of-order delivery. Each packet is labeled with a ⟨destination address, source address, and port numbers⟩$_{A3}$. It may also be labeled with the sequence number of the packet. ... At the destination, ⟨the original message/data is reassembled in the correct order, based on the packet sequence number⟩$_{A4}$.	Bağlantısız modda her bir pakette ⟨tam adresleme bilgisi⟩$_{A1}$ bulunmaktadır. Paketler ⟨bireysel olarak⟩$_{A2}$ sevk edilir ve bu bazen farklı yollara ve düzensiz teslime yol açmaktadır. Her bir paket bir ⟨hedef adresi, kaynak adresi ve port numarasıyla⟩$_{A3}$ etiketlenir. Aynı zamanda paketin sıra numarasıyla da etiketlenebilir. ... Hedefte, ⟨orijinal mesaj/veri paket sıralama sayısına göre doğru sırada yeniden birleştirilir⟩$_{A4}$.
Questions 1	What does each packet include in connectionless mode?	Her bir paket bağlantısız modda ne içerir?
2	How are the packets routed?	Paketler nasıl sevk edilir?
3	What is included with each packet label?	Her bir paket etiketinde ne bulunur?
4	What happens to the packet at the destination?	Hedefte pakete ne olur?

In this chapter, we introduce an OpenQA system that employs conversational context, specifically designed for languages with limited resources. We use `SQuAD-TR` and `XQuAD-TR` as our training and evaluation datasets, respectively. Although these datasets were not originally designed for Conversational QA, we treat questions about the same paragraph as part of a single conversation, distinguishing our approach from the previous studies that use datasets specifically crafted for Conversational QA and OpenQA settings, such as CoQA [125], QuAC [104], and OR-QuAC [103], which are not available in low-resource languages like Turkish. Our approach This enables the utilization of these datasets for the development of OpenQA systems, not just for handling ad-hoc questions but also for addressing multi-turn questions in conversational scenarios. To illustrate, Figure 5.1 shows how the example in Table 5.1 is reflected on a conversational context. This approach mitigates the need for large-scale Conversational QA datasets in resource-constrained languages. As a specific case

study, we implement our system for Turkish, a language that exhibits significant typological and grammatical differences from English, thereby reflecting various distinctions present in other languages as well. We integrated our approach into both the retriever and reader of our OpenQA system. Unlike previous research [112] which incorporated conversational signals only into the retriever, we employed BM25 and ColBERT-QA retriever, representing the sparse and dense retrievers respectively. We use BERTurk model as the common choice for the readers. Incorporating conversational signals consistently led to improved results in both our retriever and reader models. We publicly release our code and models.[36]



(a) (b)

Figure 5.1. Showing an XQuAD entry in a conversational context. (a) The example in English. (b) The example in Turkish.

The contributions of our work to the Conversational OpenQA field for languages with limited resources can be outlined as follows:

- We present a cost effective Conversational OpenQA system that effectively utilizes conversational context, tailored for languages with limited linguistic resources. Our methodology is applied to Turkish, demonstrating the adaptability and effectiveness of our approach in a language with considerable typological differences from English. This establishes a foundation for Conversational OpenQA applications in other resource-constrained languages.

[36] https://github.com/e-budur/convqa-tr

- We enhance both the retriever and reader components of the OpenQA systems by adding conversational context. This involves using BM25 and ColBERT-QA for the retrieval and BERTurk for the reader. We show that this enhancement leads to significant improvement in the overall performance of these models in Conversational OpenQA scenarios, even when working with a language facing resource constraints.
- By repurposing existing QA datasets, SQuAD-TR and XQuAD-TR, for Conversational OpenQA settings, we demonstrate the applicability of these datasets for handling multi-turn questions in conversational scenarios. Morover, we show the utility of machine-translated QA dataset SQuAD-TR on conversational setting, thereby reduce the need for large-scale gold Conversational OpenQA datasets in resource-constrained languages.
- To foster ongoing research and applications, we make our code and models publicly available, providing essential tools for the development and advancement of Conversational OpenQA systems in resource-constrained language settings.

The rest of the chapter is organized as follows: Section 5.2 explains the datasets and the methods used in this work in detail, and describes the evaluation metrics. Section 5.3 presents the results and insights into the findings obtained. Section 5.4 discusses how our method can be extended to other non-English languages. Finally, Section 5.5 concludes the chapter.

5.2. Methodology

In this section, we introduce the Conversational OpenQA system proposed in this work that uses a model-agnostic approach to keep track of the history of the conversation and we adapt SQuAD-TR and XQuAD-TR for conversational context. In what follows, we elaborate on the components and resources the system depends on.

5.2.1. Datasets

Building a Conversational OpenQA system requires a QA dataset acquired in a conversational context, which is often missing in languages with limited resources as in the case

of the Turkish language. To address this issue, we adapted SQuAD-TR [2] and XQuAD-TR [3] to simulate a conversational environment. Our approach involved treating all consecutive questions related to a specific passage as follow-up questions within a conversation as exemplified in Table 5.1. The example in the table was originally crafted for the XQuAD dataset with independent ad-hoc questions without any conversational context. However, the questions contain implicit reliance on the first question that contains the content words "connectionless mode" which disambiguates the meaning of the word "packet" into "network packet", which might otherwise refer to the word "parcel" as well. Handling each question supported by its conversational context helps resolve such disambiguities. Figure 5.1 shows how the questions of the same example are ordered in sequence in a conversational context. This approach presents an alternative method for obtaining a Conversational QA dataset for languages with resource constraints. After the SQuAD-TR and XQuAD-TR datasets are converted into conversational datasets as explained, we used the training split of SQuAD-TR to build the models and XQuAD-TR for evaluation. The basic statistics of the datasets are shown in Table 5.2.

Table 5.2. Statistics for the SQuAD-TR-TRAIN [2] and XQuAD-TR [3] datasets.

	SQuAD-TR-TRAIN	XQuAD-TR
# Articles	442	48
# Paragraphs ($\sim$ Conversations)	18776	240
# Answerable questions	61293	1190
# Unanswerable questions	43498	0
# Total Questions	104791	1190

To compare the results of the proposed Conversational OpenQA system with those reported by Budur et al. [2], we used the Wikipedia dump of May 01, 2023 as our knowledge source and followed the same procedure suggested to process it. For example, we appended the passages in SQuAD-TR and XQuAD-TR to the Wikipedia dump so that they are available for the retriever since the Turkish Wikipedia does not contain all the passages in SQuAD-TR and XQuAD-TR. Then, we splitted the merged passages into segments of 75 words to ensure they are of an appropriate size for inputting into the BERT-based models in our system. As a result, we obtained $2,192,776$ passages each with a maximum of 75 words.

5.2.2. Conversational OpenQA System

The Conversational OpenQA system proposed in this work is formed of two submodules, namely the retriever and the reader, as in previous studies [2, 12]. However, in contrast to previous studies, we utilize the conversational context in both modules to enhance their ability to answer the given question. We provide conversational context by appending the previous questions to the current question in chronologically reverse order to form the contextualized question.

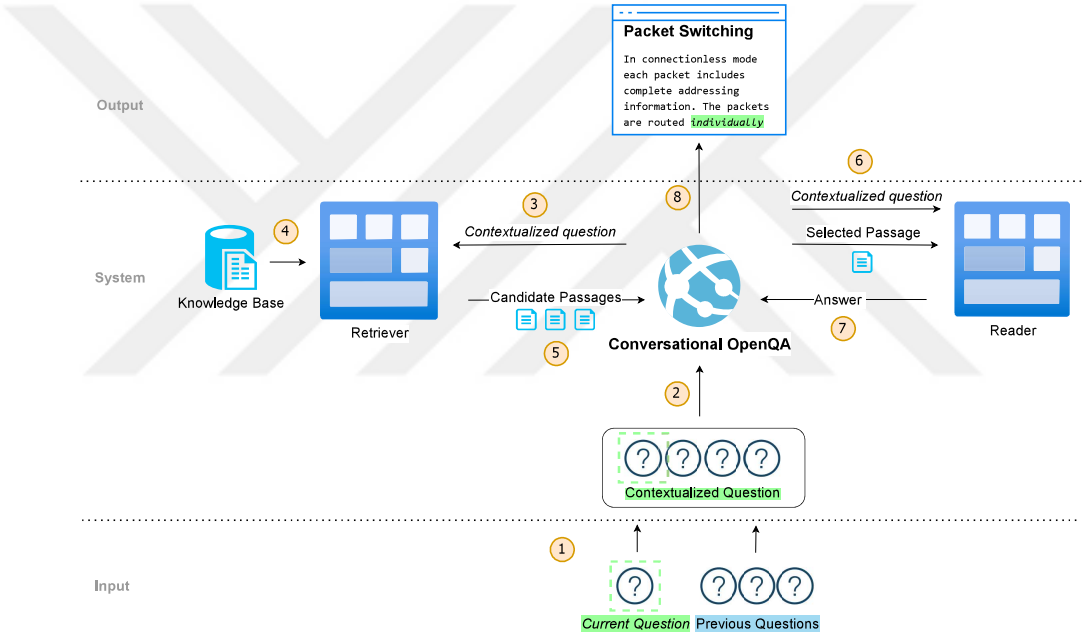


Figure 5.2. Overview of the Conversational OpenQA system with the numbers showing the information flow.

Figure 5.2 illustrates the information flow of the system with the numbered steps the numbered steps outlined below:

① The current question is given to the system with the previous questions as context.

② The contextualized question is derived by adding preceding questions to the current question in reverse chronological order.

③ The contextualized question is then given as input to the retriever model.
④ The retriever model retrieves passages relevant to the contextualized question.
⑤ The retriever model returns a ranked list of candidate passages.
⑥ The contextualized question, along with a passage selected from the candidate passages, is given as input to the reader model.
⑦ The reader model locates the answer in the selected passage for the given contextualized question.
⑧ The system returns the answer for the current question with the previous questions given as context.

In essence, given the contextualized question, the retriever model returns candidate passages as output. Then, the reader model takes both the contextualized question and a selected candidate passage as input, and returns an answer span in the passage that best fits the context of the conversation.

5.2.2.1. Task Formulation. Let q_k represent the k'th question in a conversation C having a series of questions $\{q_i\}_1^k$, each seeking a correct answer span a_k from a passage p_k having a relevancy score r_k in a large corpus $\mathcal{P} = \{p_i\}_1^N$ with N passages. The contextualized question $Cq_k = (q_k, D, q_{k-1}, D, \cdots, D, q_1)$ is formed by concatenating q_k to its preceding questions with a delimiter D.

5.2.2.2. Models. We adapt BM25 and ColBERT-QA to the conversational context for our retrievers in addition to their adaptations for Turkish outlined by Budur et al. [2], and pair each of them with a BERTurk model as their reader. The resulting conversation-aware models are named as ConvBM25, ConvCOLBERT-QA, and ConvBERT, respectively, to denote their use of conversational context. We briefly describe below how these models are further adapted to leverage the conversation history.

ConvBM25: The ConvBM25 retriever takes advantage of the complete conversational history by leveraging all the preceding questions within the conversation. The contextual

question for BM25 is formed as

$$Cq_k = (q_k \cdot \texttt{[SPACE]} \cdot q_{k-1} \cdot \texttt{[SPACE]} \cdot q_{k-2} \cdot \texttt{[SPACE]} \cdots q_1) \tag{5.1}$$

with the space character as the delimiter. We paired `pyserini` [178][37] with `anserini` [179][38] to build the underlying BM25 model. We adapted the BM25 model for Turkish by incoporating a Turkish morphological stemmer, namely Zemberek [29, 180], into the Solr instance in `anserini`. We used the default values for other parameters available in the underlying Solr instance.

The output of the BM25 retriever also serves as the bootstrap training dataset for ColBERT-QA following the previous studies [2, 12]. When building the training bootstrap dataset for ColBERT-QA, we create a set of (question, positive passage, negative passage) triples. Positive passages are those containing the answer span for the given question, while negative passages lack any text span matching the target answer text. To form the dataset, we select all positive passages from the top k results obtained from the BM25 retriever for a given question in `SQuAD-TR`. If there are no positive examples within the top k results or the question is labeled as unanswerable, we exclude that question from the bootstrap dataset. Each positive passage is then paired with a negative passage also from the top k results for the question.

ConvCOLBERT-QA: The contextualized question for the ConvCOLBERT-QA reader is formed as

$$Cq_k = (q_k \cdot \texttt{[SEP]} \cdot q_{k-1} \cdot \texttt{[SEP]} \cdot q_{k-2} \cdot \texttt{[SEP]} \cdots q_1) \tag{5.2}$$

with `[SEP]` token as the separator. Consequently, the input format for the underlying ColBERT-QA model is translated into $(\texttt{[Q]} \cdot Cq_k \cdot \texttt{[D]} \cdot p)$, where `[Q]` is the question marker, `[D]` represents the document marker, and p is the passage to calculate the similarity against

[37] `https://github.com/castorini/pyserini`
[38] `https://github.com/castorini/anserini`

the contextualized question Cq_k.

Given that Cq_k is used as the query input for ColBERT-QA, its built-in similarity function inherently incorporates the conversational history accordingly. Let $\{tq_i\}_{i=1}^{L}$ and $\{tp_j\}_{j=1}^{N}$ denote the output embeddings of ColBERT-QA for the tokens of Cq_{k} and p, respectively, where L is the number of tokens in Cq_{k} and N is the number of tokens in p. Then, the similarity between the contextualized question Cq_{k} and the passage p is calculated by the built-in similarity function of ConvCOLBERT-QA as follows:

$$MaxSim(Cq_{\mathrm{k}}, p) = \sum_{i}^{L} \max_{\{tp_j\}_{j=1}^{N}} tq_i \cdot tp_j \tag{5.3}$$

The similarity score for a question token is taken as its dot product value with a passage token that maximizes this value, and the scores of all question tokens are summed to arrive at the similarity between the question and the passage. In this formula, we include the entire conversation history when calculating similarity as opposed to Krasakis et al. [112], who compute the similarity solely based on the tokens of the last question. However, in practice, we let the oldest questions be truncated when the length of the contextualized question exceeds the input length limit for questions. We limit the input length for the question at 32, while the passage length limit is set to 180.

We customized ColBERT-QA for Turkish as in a previous study [2]. For this customization, we utilized the BERTurk base cased model to initialize the weights of the ColBERT-QA model. Given that the default marker tokens specified in ColBERT-QA are not present in the BERTurk tokenizer, we assigned the “blush” emoji (U+1F60A) to the query marker and the “smiley” emoji (U+1F603) to the document marker.

We trained ColBERT-QA for one epoch with a batch size of 32 without gradient accumulation on a single NVIDIA Tesla V100 GPU. In computing MaxSim for the query and document inputs, we employed L2 similarity. Our implementation of ColBERT-QA is derived from the original repository[39] with default values applied for all other parameters.

[39] `https://github.com/stanford-futuredata/ColBERT` - Commit id: `a69fe8ca`

ConvBERT: Like ConvCOLBERT-QA, the ConvBERT reader creates the contextualized question as

$$Cq_k = (q_k \cdot \texttt{[SEP]} \cdot q_{k-1} \cdot \texttt{[SEP]} \cdot q_{k-2} \cdot \texttt{[SEP]} \cdots q_1) \tag{5.4}$$

utilizing [SEP] as the separator token. Additionally, we combine the contextualized question and a candidate passage returned by the retriever using [SEP] as the separator. We give them as input to the BERTurk model as ([CLS] $\cdot\, Cq_k \cdot$ [SEP] $\cdot\, p$), where [CLS] denotes the built-in CLS token of BERT and p denotes the selected candidate passage to seek the answer from. However, as in ConvCOLBERT-QA, ConvBERT primarily benefits from the most recent historical questions as it needs to truncate the historical questions at the end of the sequence if the length of the contextualized question exceeds the maximum input length.

The context passages used in the training and evaluation datasets for the ConvBERT models are derived from the retriever outputs of the ConvBM25 and ConvCOLBERT-QA models. In the training dataset, the first positive passage among the top k retriever results is selected as the context passage for the reader. If no positive passage is identified within the top k candidate passages, the first passage is selected instead. In the evaluation dataset, we always select the first passage as the context of the question.

We initialized ConvBERT using BERTurk base cased model and trained it for 5 epochs with a batch size of 16 without gradient accumulation on a single NVIDIA Tesla V100 GPU. We applied a learning rate of $3X10^{-5}$ and used a maximum sequence length of 384 and a document stride set to 128. Our implementation is based on Huggingface's transformers library [177].[40] We excluded the unanswerable questions in SQuAD-TR during the training.

5.2.3. Evaluation Metrics

To evaluate the retrievers, we use "Success@k" (S@k) and "Count@k" (C@k) to enable comparison with the previous studies [1, 2, 12, 81, 182]. $S@k$ refers to the ratio of questions

[40] https://github.com/huggingface/transformers - Version: 4.14.0.dev0

that have at least one relevant passage from the top k results returned by the retriever. On the other hand, $C@k$ denotes the average number of the relevant passages out of the top k results returned by the retriever. The passage is considered as relevant if it contains any of the target answers via exact match. It should be noted that $S@k$ is a percentage value whereas $C@k$ is a positive real number between 0 and k.

Readers are evaluated with the Exact Match (EM) and F1 scores, which are frequently used in previous studies as the standard metrics for readers in QA systems [1, 2, 8, 12, 50, 81, 95, 103]. EM score is the percentage of the questions for which the predicted answer precisely matches any of the ground-truth answers. On the other hand, F1 score assigns partial credit if the predicted answer partially overlaps with a ground-truth answer. Note that the EM metric used in the reader evaluation is equivalent to the $S@1$ metric employed in the retriever evaluation, enabling a direct comparison between the performance of the retrievers and readers in OpenQA systems.

5.3. Experiment Results

In this section, we present the results of the retriever and reader models while sharing our analysis and insights into their performance. We compare our results with those reported by Budur et al. [2] that is currently the only work using the same dataset.

5.3.1. Retriever Results and Analysis

Table 5.3 presents the retriever results on the XQuAD-TR dataset, highlighting the impact of incorporating conversational history by comparing with the previous results that do not use conversational history [2]. Figure 5.3 illustrates these metrics to aid in benchmarking various retriever models. The results clearly show that including the conversational history helps increasing the performance of the retrievers significantly for both types of models across all $S@k$ and $C@k$ values. The consistent increase in $S@k$ and $C@k$ values implies that the retriever demonstrates improved efficacy in locating relevant passages when supported by conversational signals. Our empirical observation showed that there are also a few cases where

conversational history may inadvertently retrieve passages that create confusion. Detailed examples can be found in Table C.1 and Table C.2 in Appendix C, showcasing scenarios where conversational history improves retrieval accuracy or introduces challenges on the overall performance of the OpenQA system, respectively.

The increase is more evident between the BM25 and ConvBM25 models with an absolute performance gain of 18.82% and 0.19 for $S@1$ and $C@1$ values, respectively. In contrast, the ConvColBERT-QA model demonstrates a relatively more modest improvement of 4.54% and 0.05 for $S@1$ and $C@1$ values, respectively, over ColBERT-QA. We also observe that the performance gap between the BM25 and ColBERT-QA models narrowed down from 20.84% (55.21% vs. 76.05%) to 6.56% (74.03% vs. 80.59%) for $S@1$ and even further to 3.62% (92.18% vs. 95.80%) for $S@20$ when both models incorporated conversational history.

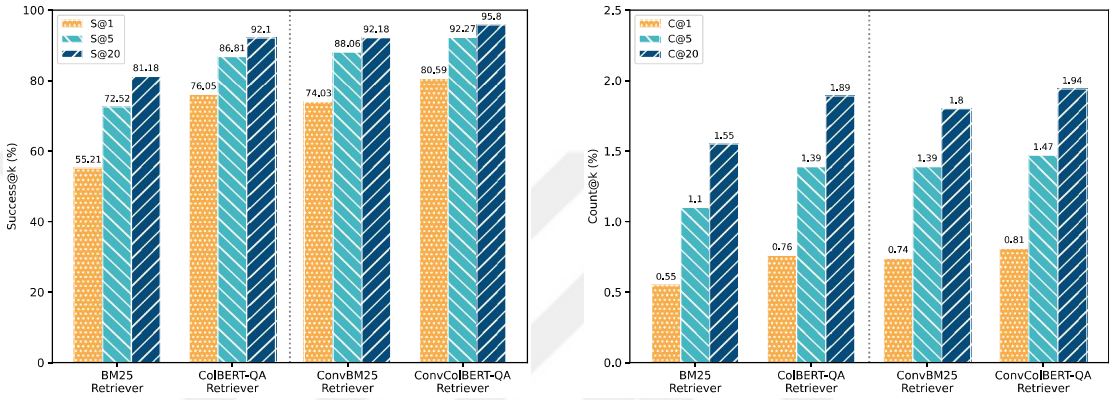
Table 5.3. Retriever results with/without historical questions used for training. The bolded figures indicate the highest values within their respective columns. The results are evaluated on XQuAD-TR.

Retriever Model	**History Used**	**S@1/C@1**	**S@5/C@5**	**S@20/C@20**
BM25 [2]	No	55.21/0.55	72.52/1.10	81.18/1.55
ColBERT-QA [2]	No	76.05/0.76	86.81/1.39	92.10/1.89
ConvBM25	Yes	74.03/0.74	88.06/1.39	92.18/1.80
ConvColBERT-QA	Yes	**80.59/0.81**	**92.27/1.47**	**95.80/1.94**

The high performance in $S@5$ and $S@20$ values can be propagated to $S@1$ values if a robust reranker module is applied to the retriever results. Previous studies support the importance of a reranker module in the Conversational OpenQA context [103, 195]. We will further explore the effect of the reranker module in the Conversational OpenQA context for low-resource languages as a future work.

One potential explanation for the severe performance increase in ConvBM25 compared to BM25 can be attributed to its ability to leverage the entire conversation history, whereas

ConvCOLBERT-QA (compared to ColBERT-QA) was constrained to using only the latest segment of the previous questions when the corresponding contextualized question exceeded the input length for the query due to design constraints. An alternative explanation might be the fact that BM25 had more room for improvement compared to ColBERT-QA and the additional history information provided a benefit to ConvBM25 more than to ConvCOLBERT-QA.



(a) (b)

Figure 5.3. Success metrics for the retrievers evaluated on XQuAD-TR, highlighting the contribution of contextual signals to the overall performance of the retrievers. The left segments of each figure, separated by a dotted line, show the retrievers without using conversational signals. The right segments of each figure show the retrievers where conversational signals are used. a) Success@k metrics. b) Count@k metrics.

It should be noted that an elevation in retriever scores may not necessarily translate into an overall performance improvement for question answering unless it is validated by the increase in the corresponding reader scores. It is important to determine if the retrieved passages are semantically more relevant or merely contain the answer text without the contextual relevance to the question. For instance, if the answer text is a year, that specific year may appear frequently in various contexts. Retrieving any passage containing the target year could enhance the retriever performance but might not necessarily contribute to improving the reader performance. Therefore, we need to assess the performance of the retriever scores also through their reader scores, as will be explained in the next subsection.

5.3.2. Reader Results and Analysis

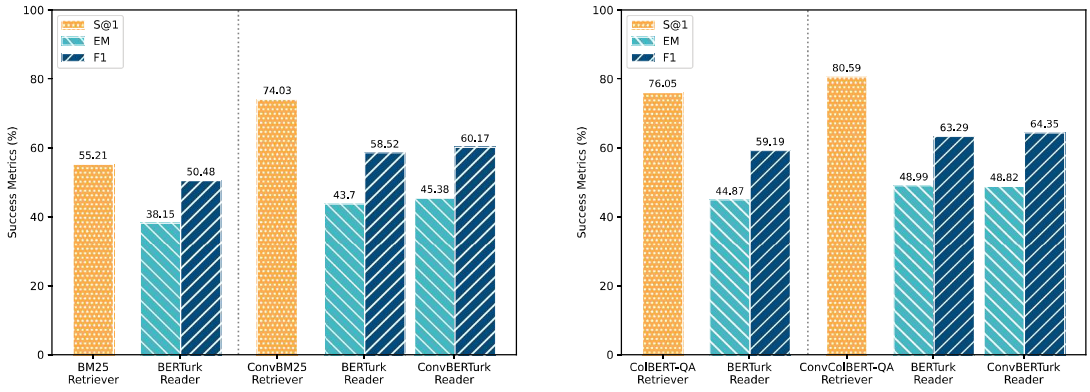
Table 5.4 shows the Conversational OpenQA (ConvOpenQA) reader scores based on both types of the retriever models and compares with the previous results reported by Budur et al. [2] for Standard QA and OpenQA readers that do not utilize conversational signals. Figure 5.4 also illustrates these results, accompanied with the corresponding retriever results, to facilitate comparison of success metrics among these OpenQA systems. The values in parentheses under Training Dataset and Test Dataset columns denote the name of the retriever used to derive the corresponding dataset. The values in parentheses under the EM/F1 columns show the percentage of the corresponding retriever's $S@1$ value achieved by the reader (e.g. the BM25-BERTurk EM score of 38.15 is 69.10% of the BM25 $S@1$ value of 55.21 in Table 5.3). Aligned with the retriever results, we observe a clear improvement also in the reader scores when conversational history is given as extra information. Although the precise mechanism by which the reader utilizes the conversational signal to improve answer prediction is not entirely clear, we hypothesize that leveraging conversational signals allows the reader to develop a more comprehensive understanding of the user's information needs. This, in turn, likely contributes to an overall improvement in the reader performance, though it may also occasionally affect negatively when previous questions introduce noise. Table C.3 and Table C.4 in Appendix C provide two illustrative examples where conversational history enhances or disrupts the reader's performance, respectively.

Table 5.4. Reader results with/without conversation history used for inference. The results are evaluated on the respective XQuAD-TR versions of the corresponding retriever model.

QA Type	Reader Model	Retriever Model	Training Dataset	Test Dataset	EM	F1
Standard QA	BERTurk [2]	–	SQuAD-TR-TRAIN	XQuAD-TR	51.17	67.78
	ConvBERTurk	–	SQuAD-TR-TRAIN	XQuAD-TR	52.94	68.78
OpenQA	BERTurk [2]	BM25	SQuAD-TR-TRAIN (BM25,2023)	XQuAD-TR (BM25,2023)	38.15 (69.10%)	50.48 (91.43%)
	BERTurk [2]	ColBERT-QA	SQuAD-TR-TRAIN (ColBERT-QA,2023)	XQuAD-TR (ColBERT-QA,2023)	44.87 (59.00%)	59.19 (77.83%)
ConvOpenQA	BERTurk	ConvBM25	SQuAD-TR-TRAIN (ConvBM25)	XQuAD-TR (ConvBM25)	43.70 (59.03%)	58.52 (79.05%)
	BERTurk	ConvColBERT-QA	SQuAD-TR-TRAIN (ConvColBERT-QA)	XQuAD-TR (ConvColBERT-QA)	48.99 (60.79%)	63.29 (78.53%)
	ConvBERTurk	ConvBM25	SQuAD-TR-TRAIN (ConvBM25)	XQuAD-TR (ConvBM25)	45.38 (61.30%)	60.17 (81.28%)
	ConvBERTurk	ConvColBERT-QA	SQuAD-TR-TRAIN (ConvColBERT-QA)	XQuAD-TR (ConvColBERT-QA)	48.82 (60.58%)	64.35 (79.85%)

When we use conversational history in the retriever but not in the reader, BM25-based readers exhibit an increase of 5.55%/8.04% in the EM/F1 scores (38.15% vs. 43.70% / 50.48% vs. 58.52%), while this improvement is saturated at 4.12%/4.10% (44.87% vs. 48.99% / 59.19% vs. 63.29%) for ColBERT-QA-based readers. Integrating conversation history into both readers and retrievers consistently improves the performance of BM25-based readers, with an additional increase of 1.68%/1.65% in the EM/F1 scores (43.70% vs. 45.38% / 58.52% vs. 60.17%). In contrast, ColBERT-QA-based readers experience a slight decrease of 0.17% in the EM score (48.99% vs. 48.82%) but an increase of 1.06% in the F1 score (63.29% vs. 64.35%). The notable performance improvement of BM25-based readers, despite sharing the same BERT architecture as ColBERT-QA-based readers, can be attributed to their relatively low scores in the OpenQA setting, thus having more room for improvement for the Conversational OpenQA setting.

Since the performance of a reader depends on the performance of the underlying retriever, each retriever establishes an oracle (upper limit) for its reader as indicated by the $S@1$ values in Table 5.3. The extent to which each reader attains its specific upper limit is shown near its EM/F1 scores within parentheses in Table 5.4. For instance, the EM score of the BM25 retriever and BERTurk reader setting (38.15%) is 69.10% of the $S@1$ score of the BM25 retriever in Table 5.3 (55.21%) which serves as the upper bound for the reader. These percentages offer insights into the marginal performance of each reader relative to the improvement in its retriever performance. As stated above, even though we observed a notable performance improvement in ConvBM25-based readers compared to BM25-based readers, we notice a decrease in the percentage at which ConvBM25-based readers reach their upper limits when conversational history is included (69.10% vs.59.03%/61.30%). This observation suggests that 1) the performance gain in ConvBM25-based readers is primarily attributed to the enhancement in their retriever scores, and 2) the ConvBM25 retriever may not necessarily offer candidate passages as effectively as the notable increase in its $S@1$ score compared to the BM25 retriever would suggest. In contrast, we observe that ConvCOLBERT-QA-based readers achieve an increasing percentage of their individual upper limits compared to ColBERT-QA-based readers (60.79%/60.58% vs. 59%), indicating that ConvCOLBERT-QA provides better examples for the reader, leading to further overall improvement.



(a) (b)

Figure 5.4. Success metrics for OpenQA systems evaluated on XQuAD-TR, highlighting the contribution of contextual signals to the overall performance. The left segments of each figure show systems without conversational signals, while the right segments show systems using these signals in retrievers or both retrievers and readers. (a) Success metrics for BM25-based OpenQA systems. (b) Success metrics for ColBERT-QA-based OpenQA systems.

To set a global upper limit for Conversational OpenQA readers, we integrated the conversation history into the StandardQA model, henceforth Conversational Standard QA model, denoted as the ConvBERTurk model in the first part in Table 5.4. Adding conversation signals in the Conversational Standard QA model led to an increase of 1.77%/1.00% in EM/F1 scores, respectively, over the Standard QA model scores. For a fair comparison, we compare the OpenQA models against the Standard QA model, and the Conversational OpenQA models against the Conversational Standard QA model. As such, the gap in EM/F1 scores between the BM25-/ConvBM25-based readers and their global upper limits dropped from $13.02\%/17.30\%$ (51.17% vs. 38.15% / 67.78% vs. 50.48%) to $7.56\%/8.61\%$ (52.94% vs. 45.38% / 68.78% vs. 60.17%), respectively. Conversely, ColBERT-QA-/ConvColBERTQA-based readers narrowed the gap between their global oracle further from $6.30\%/8.59\%$ (51.17% vs. 44.87% / 67.78% vs. 59.19%) to $4.12\%/4.43\%$ (52.94% vs. 48.82% / 68.78% vs. 64.35%) in the EM/F1 scores, respectively. These results prove the effectiveness of the conversational history as a pertinent information for OpenQA systems to improve their results toward their global upper limit.

5.4. Generalization to Other Non-English Languages

The methodology we presented in this chapter can be effectively extended to other low-resource languages by creating machine-translated datasets like SQuAD-TR for training and using human-translated datasets for evaluation, as outlined by Budur et al. [2]. These machine-translated datasets can serve as a stepping stone for low-resource languages, providing a solution for generating training data where human translation is not feasible due to resource constraints. Additionally, there are several multilingual evaluation datasets (e.g. XQuAD [3], MKQA [192], MLQA [13], XQA [183]) that can support many other non-English languages. These datasets not only provide benchmarks for evaluating model performance but also highlight the unique characteristics and challenges posed by diverse linguistic phenomena, such as syntax, morphology, and semantics. While we used Wikipedia as our knowledge source, other sources, such as domain-specific corpora, news articles, or structured databases, can also be employed. Similarly, various retriever models (e.g. RAG [82], DPR [1], ORQA [56], REALM [81]) and reader models (e.g. mBERT [25], XLM-R [65]) can be utilized based on resource availability in the target language use-case. Overall, the flexibility in the choice of datasets, knowledge sources and models enables the methodology to adapt to diverse linguistic contexts.

5.5. Summary

In this chapter, we demonstrated the feasibility of developing a robust Conversational Open QA system for low-resource languages, with Turkish serving as a case study. We used a model-agnostic approach for integrating conversational signals into OpenQA systems, which are applied in both the retriever and reader models. Our experiment results showed that integrating historical questions not only improves the retriever performance but also enhances the reader performance, underscoring the benefits of integrating conversational signals in OpenQA systems. Our findings also demonstrated that the inclusion of conversational signals further reduces the performance gap between OpenQA and Standard QA models, where the performance of Standard QA models serves as an upper bound since they are fed as input with passages that already include the answers.

The data scarcity problem for languages other than English has been a substantial impediment to progress in those languages. When developing Conversational OpenQA systems for low-resource languages, we addressed this challenge by repurposing the SQuAD-TR and XQuAD-TR datasets for the Conversational OpenQA setting. The results highlighted the practicality of utilizing automatically derived datasets to build a robust Conversational OpenQA system, even with limited resources.

Our findings highlight the effectiveness of our approach, even when applied to datasets not explicitly designed for multi-turn interactions or Conversational OpenQA scenarios. However, it is worth noting that these results may not fully represent the full potential achievable with datasets specifically crafted for conversational use. Additionally, while the proposed methodology is inherently language-agnostic, the generalizability of the findings to other non-English languages requires further experiments on those languages.

Our study specifically concentrates on the answerable questions in the SQuAD-TR dataset. In future work, we aim to enhance our approach by developing a Conversational OpenQA model for low-resource languages that also identifies unanswerable questions in SQuAD-TR using an integrated out-of-domain detection model. Moreover, we plan to translate a Conversational QA dataset [103, 104, 125] into languages with limited resources. This initiative will aim to make a Conversational QA dataset accessible for low-resource languages, thereby contribute to the further advancement of the field.

6. Q: WHAT IS THE ULTIMATE WISDOM OF A QA SYSTEM? A: I DO NOT KNOW.

> Being a lifelong learner isn't about taking pride in your knowledge. It's about having the humility to know what you don't know.
>
> Adam Grant

In the previous chapters, we tackled various challanges to obtain a robust Conversational OpenQA system for low-resource languages. We showed that the resulting system can answer the given questions from a large knowledge source dataset. However, what is missing in our system is to be able to simply say "I do not know" when the question requires a context out of the scope of its knowledge source.

In this chapter, we will briefly share a methodology to improve our QA system in a way that it recognizes its low-confidence on answering a given question.

6.1. Methods

Our method is based on the capability of Natural Language Inference (NLI) models to recognize the entailment and contradiction of the given hypothesis and premises statements. We use our NLI-TR dataset to pretrain a Turkish NLI model based on BERTurk, which serves as a baseline model that recognize entailment and contradiction between the given inputs. Then, we finetune NLI model to obtain our Out-Of-Domain (OOD) detection model that differentiate the unanswerable questions from answerable ones in our `SQuAD-TR` dataset. In this part, we use the question and its answer as the "hypothesis" statement and provide the context as the "premise" and expect the model to predict whether the context loosely entails the question for the expected answer. Figure 6.1 shows the high-level and low-level overview of our system for OOD detection, respectively.

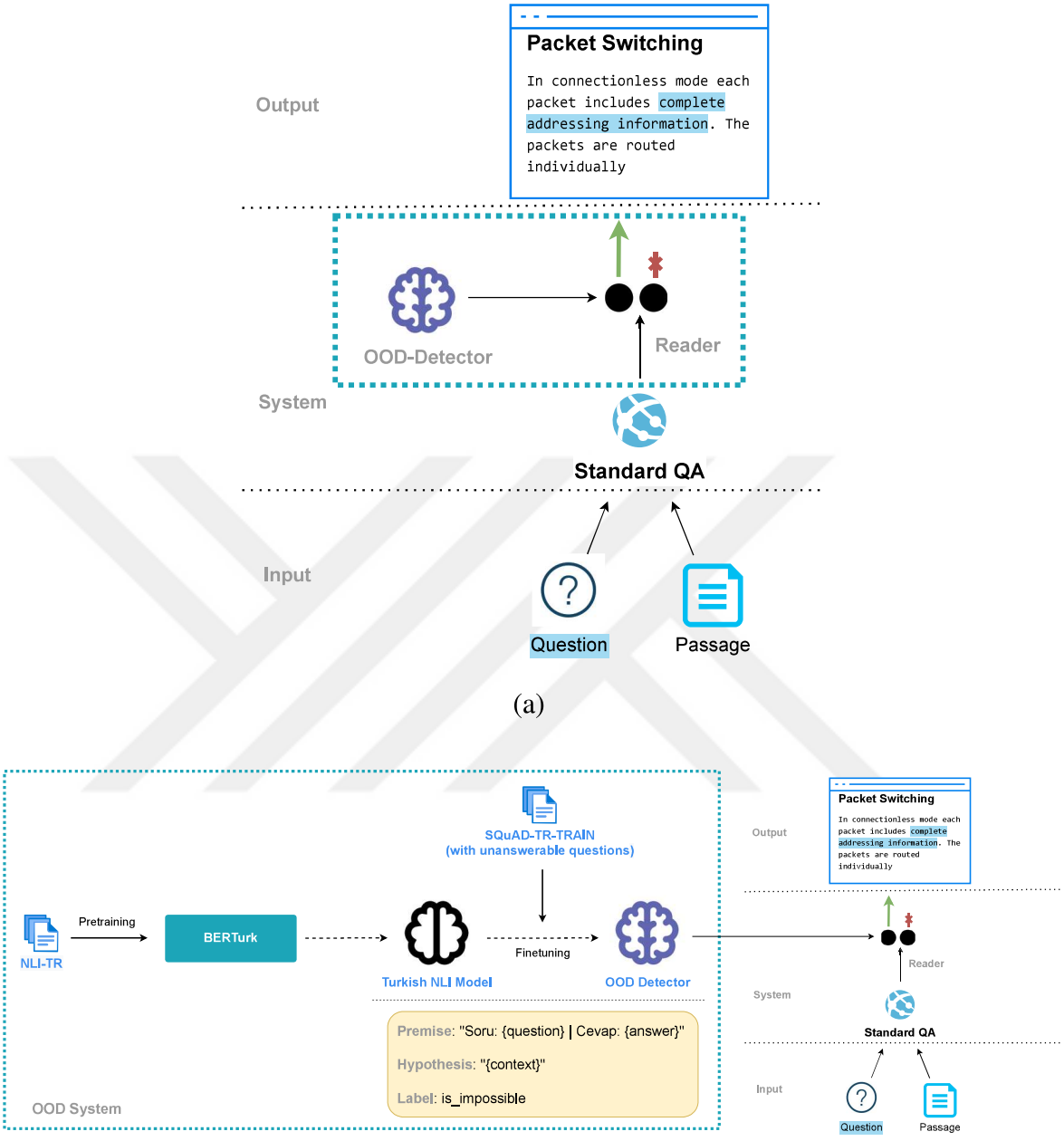


Figure 6.1. Out-of-Domain (OOD) Detector System Overview. (a) High-Level Overview. (b) Low-Level Overview.

Let (q, a, c, u) be a quadruple from `SQuAD-TR` denoting question, answer, context passage and unanswerability (`is_impossible`), respectively. We convert this quadruple into a triple (p, h, u) denoting premise, hypothesis and unanswerabiliy, respectively, where $p = q||a$ and $h = c$. Then we use the resulting triple to train a binary classifier. We use BERT as our binary classifier initialized with the weight of our pretrained NLI model.

For example, let the quadruple example in SQuAD-TR be as follows:

```
Question: Uzakdoğu koleksiyonlarında yaklaşık olarak kaç tane sanat eseri
          yer almaktadır?
Answer: 70.000'den fazla
Context: Uzakdoğu koleksiyonları, Doğu Asya ülkelerinden 70.000'den fazla
         sanat eserini içerir...
Unanswerable: False
```

Then, the resulting triple for our example will be as follows:

```
Premise: Uzakdoğu koleksiyonlarında yaklaşık olarak kaç tane sanat eseri
          yer almaktadır? | 70.000'den fazla
Hypothesis: Uzakdoğu koleksiyonları, Doğu Asya ülkelerinden 70.000'den fazla
         sanat eserini içerir...
Unanswerable: False
```

By using the resulting triples as explained above, we obtain our OOD model. Then, we incorporate our OOD model into our Standard QA model as an additional signal to help recognize the unanswerable questions.

In its original implementation by Devlin et al.[25], a “span logit” is calculated for each possible span of texts . Additionally, a baseline logit, a.k.a. “null logit” is derived from the [CLS] which has high logit values for the unanswerable questions due to the fact that the unanswerable questions uses the start index 0 by design. The original BERT model recognizes the unanswerable question if the “null logit” is higher than the “span logit”. We extend this logic by incorporating the output of our OOD model as an addtional condition to be satisfied to recognize the unanswerability of the example. That is, our model not only require “null logit” to be higher than the “span logit”, but also OOD model to give a softmax score greater than threshold, e.g. 0.5 for the given example.

6.2. Experiment Results

Table 6.1 shows the results of BERTurk models trained on SQuAD-TR-TRAIN and evaluated on SQuAD-TR-DEV, with and without OOD detection model incorporated. The evaluation metrics include EM/F1 scores in all (Overall), only answerable (Has Answer) and only unanswerable (No Answer) questions.

Table 6.1. Standard QA reader results with/without OOD detector, trained on SQuAD-TR-TRAIN evaluated on SQuAD-TR-DEV.

	Overall		**Has Answer**		**No Answer**	
Reader Model	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
BERTurk	67.08	70.12	47.87	**58.61**	74.67	74.67
BERTurk + OOD	**67.12**	**70.15**	47.87	58.57	**74.72**	**74.72**

The results illustrate that OOD model helps improve the overall results while it has a varying effect on “only answerable” and “only unanswerable” questions. The overall improvement is stemming from the significant improvement of “only unanswerable” questions, which cannot be offset by the performance degredation in “only answerable” questions. These results demonstrate that NLI-TR can be repurposed to build OOD detector model, which can be validated by SQuAD-TR dataset in a QA task.

6.3. Summary

In this chapter, we focused on enhancing a QA system for low-resource languages, specifically in its ability to recognize when it cannot answer a question. The ultimate goal of our approach is to enable the system to acknowledge its limitations by saying ”I do not know” when faced with questions beyond its knowledge scope.

Our approach leverages NLI models to detect entailment and contradiction between given hypotheses and premises. The NLI model is pretrained using the NLI-TR dataset on BERTurk and then fine-tuned for OOD detection using the SQuAD-TR dataset by converting question-answer-context tuples into premise-hypothesis pairs and training a binary classifier to distinguish unanswerable questions. Our system uses an additional condition where the output of OOD model must exceed a threshold for the question to be deemed unanswerable. Our results showed significant improvement on detecting unanswerable questions.

In summary, we presented a method to train a QA system to identify its limitations and effectively respond with "I do not know", improving its reliability and robustness in real-world applications.

7. DISCUSSION

In this study, we aimed to address the challenge of building robust Conversational Open Question Answering (OpenQA) systems for resource-constrained languages, with Turkish as the main case study. Through the development of synthetic datasets and the introduction of the BIRI system, we aimed to bridge the gap between high- and low-resource languages in the context of Conversational AI. This chapter reflects on the key findings, discusses the limitations of the work, and explores its broader implications, along with avenues for future research.

7.1. Key Findings

One of the most significant findings of our thesis is the demonstrated feasibility of adapting OpenQA systems to resource-constrained languages through a combination of neural machine translation and weak supervision. By translating well-established English NLI and QA datasets into Turkish and repurposing them for Domain-Aware Conversational OpenQA, we managed to obtain a substantial amount of labeled data in a cost-effective manner. Despite the noise introduced by automatic translation, human evaluations showed that these datasets were sufficient to train high-performing models, especially when combined with weak supervision techniques.

The development of the BIRI system, which incorporates conversational signals, demonstrated that integrating context from previous part of the dialogue significantly improves the performance of the OpenQA system. By considering conversation history, the system was better equipped to retrieve relevant passages and generate more accurate answers. This confirms the hypothesis that conversational context plays a pivotal role in the success of QA systems, especially when engaging with users through multi-turn interactions.

Another key innovation in our work is the integration of domain-awareness, facilitated through the use of a Natural Language Inference (NLI) model developed in a resource-

constrained setting. This NLI model served as an effective out-of-domain detector, enabling the system to identify when a question fell outside its knowledge scope. By recognizing these limits, the system could respond with "I do not know" when appropriate, significantly improving its reliability in real-world applications. This approach enhanced the system's ability to prioritize and filter relevant information, ensuring that it remained focused on domain-specific knowledge when required.

Furthermore, our research demonstrated that it is possible to achieve robust performance in OpenQA systems for low-resource languages with only a few hundred labeled examples. By leveraging weak supervision and using conversational signals, we demonstrated that even in scenarios where annotated data is scarce, it is still feasible to build functional and reliable systems suitable for real-world use cases.

7.2. Limitations

Besides the achievements, several limitations became apparent throughout the research process. One of the primary limitations is the reliance on machine translation to generate datasets. While neural machine translation provides a cost-effective way to produce synthetic data, it inevitably introduces noise and translation errors. These errors can result in mislabeled training data, which may affect the overall accuracy of the models. Although quality control measures were implemented, such as human evaluation of translated texts, the quality of the data remains a limiting factor for more precise applications.

Another limitation lies in the scalability of the approach to other resource-constrained languages. While the methods employed for Turkish are promising, their effectiveness in languages with more complex morphology or fewer available resources is not guaranteed. The linguistic and cultural diversity across languages may introduce new challenges that require tailored solutions. The focus on Turkish, although beneficial as a case study, leaves open the question of how well the system can be generalized to other low-resource languages.

Additionally, the lack of large-scale native conversational datasets for Turkish limited the potential of the BIRI system. The datasets used in our thesis were adapted from non-conversational sources, and while they served the purpose of our study, they were not originally designed for conversational contexts. As a result, the system's true potential in handling natural dialogues may not have been fully explored. A more robust evaluation would require truely conversational datasets to assess the system's performance in real-world settings.

7.3. Machine Translation as a Data Generation Tool

Machine translation has been the cornerstone of our approach due to its scalability and cost-efficiency. By translating large-scale, high-quality datasets from resource-rich languages like English into Turkish, we bypass the need for extensive, manually curated datasets. The scalability of this approach enables it to be adapted for other low-resource languages, making it a widely applicable method. However, the automatic translation process introduces vulnerabilities, especially when handling cultural nuances and idiomatic expressions. For example,[41] in English, mice in biological experiments are described as "the mice are fasting", but in Turkish, the equivalent phrase is "fareler aç bırakılır" (literally: "the mice are left hungry"), as the phrase "oruç tutmak" (fasting) carries a predominantly religious connotation in Turkish, and is not used to describe animals, especially since mice do not engage in fasting as humans do for religious or voluntary purposes. Similarly, the English idiom "apple of my eye" translates correctly into Turkish as "gözümün bebeği" (literally: "baby of my eye"), but a word-for-word translation might yield "gözümün elması" (literally: "apple of my eye"), which can be misinterpreted as "gözümün elmas-ı" (diamond of my eye) or "gözümün elma-sı" (apple of my eye), leading to ambiguity. Such errors are particularly problematic in low-resource languages, where cultural context plays a significant role and can undermine the quality of the generated datasets if left unchecked. Despite these challenges, our results demonstrate that the benefits of the machine translation-based approach—such as scalability and broad applicability—outweigh these limitations, allowing us to effectively generate useful datasets for low-resource languages.

[41] Credit: Hümeyra Bulut, *Registered Dietitian*.

7.4. Recent Data Generation Approaches

Beside machine translation, the scarcity of high-quality datasets for "low-resource language contexts" has recently led to the development of many other innovative data generation approaches as well. While it is not possible to list all methods in this continuously evolving field, a few notable ones can be highlighted to show a direction of growth for future work. These methods leverage various strategies, including not only advanced variants of machine translation but also techniques like synthetic data generation and retrieval-augmented methods, to address the bottleneck of limited training data.

One notable approach, which can be considered as an advanced variant of the naive machine translation approach, is NMT-Adapt, proposed by Ko et al. [196], which adapts high-resource neural machine translation (NMT) models to generate synthetic data for related low-resource languages without requiring parallel data. By utilizing monolingual data from low-resource languages and parallel data from linguistically related high-resource languages, this method employs adversarial training and back-translation to create synthetic parallel corpora. The cost-effectiveness of this method makes it attractive for languages with well-defined linguistic relationships, such as Catalan and Spanish, where leveraging similarities reduces the need for extensive parallel data. However, the approach is limited by its dependence on linguistic proximity; it is less effective for languages that do not have high-resource counterparts with similar syntactic or lexical structures. Consequently, its flexibility is limited, making it less applicable to a broad range of low-resource languages. In contrast, the method used in our thesis, which relies on machine translation of widely-used English datasets into a language without specific linguistic proximity, is more versatile across diverse language pairs.

An alternative approach is data augmentation through question generation, as explored by Chen et al. [197] for low-resource domains in English, such as privacy policies and technical support contexts. Their approach specifically targets the generation of diverse question-answer pairs leveraging existing datasets, by first prompting different answers, then generating varied questions, and finally filtering them to ensure relevance and consistency between the prompted answers and generated questions, briefly described as **P**rompt Answer,

Question Generation, and **Q**uestion Filter (PQQ). This method was found to outperform other augmentation strategies on low-resource domain datasets like PolicyQA [198] and TechQA [199], showing that careful question generation and filtering can produce high-quality data with minimal noise. This makes it a promising direction for future exploration in generating diverse question-answer pairs for resource-constrained languages as well. Compared to our machine translation approach, the PQQ method excels in controlling noise by using the question filter, ensuring the augmented data aligns closely with the original dataset. However, it is computationally intensive, as it requires specialized models for prompt-based answer generation and question filtering, making it potentially less scalable across multiple languages without significant fine-tuning. On the other hand, our method benefits from scalability and ease of implementation in new languages by leveraging relatively simpler NMT models, which typically require less resources than prompt-tuned LLMs.

More recently, Ziegler et al. [200] introduced CRAFT (Corpus Retrieval and Augmentation for Fine-Tuning), a scalable data generation method that retrieves relevant human-written documents from large web corpora and uses instruction-tuned large language models (LLMs) to create task-specific datasets. CRAFT can efficiently generate high-quality synthetic data starting from a few user-provided examples, significantly reducing the manual effort required for dataset creation. This approach is particularly powerful for generating data in low-resource language contexts, making it a viable alternative to traditional, labor-intensive data annotation processes. However, CRAFT's dependence on the quality of the underlying instruction-tuned LLMs introduces limitations, particularly when applied to low-resource domains where these LLMs may not perform optimally. Additionally, since CRAFT relies on retrieving human-written documents from large web corpora, it may struggle to find high-quality and relevant content for low-resource languages with limited web presence. In contrast, our study, which translates well-established high-resource language datasets like SQuAD into low-resource languages, bypasses the need for the web presence of contents in those languages.

Each of these approaches offers distinct advantages for addressing the challenge of low-resource language data generation. NMT-Adapt is highly effective for related language pairs, leveraging linguistic similarities to reduce the reliance on parallel corpora. The PQQ method

shines in its ability to control noise by generating diverse, high-quality question-answer pairs through careful filtering, but it is resource-intensive and may not scale well without significant prompt-tuning. CRAFT excels in scalability and flexibility, making it suitable for a range of tasks, though it faces challenges when dealing with low-resource domains with limited data presence on the web. In comparison, our approach—using machine translation to adapt high-resource datasets into low-resource languages provides a wide applicability to tasks like open-domain and conversational QA, though it is not without limitations such as the risk of translation noise. Our approach bridges the gap by balancing the trade-offs between scalability, task flexibility, and data quality across a variety of low-resource language contexts.

7.5. Implications of the Work

The implications of our work are far-reaching, particularly in the context of extending conversational AI technologies to a broader range of language contexts. By demonstrating the feasibility of developing OpenQA systems for resource-constrained language contexts, our thesis contributes to widening access to conversational AI beyond high-resource languages like English. The ability to build QA systems for underrepresented languages like Turkish ensures that advancements in AI can be inclusive, reaching populations that were previously underserved by such technologies.

In terms of information access, the ability to develop QA systems for languages with limited resources has the potential to transform how people access knowledge and services. For many communities, access to information in their native language is critical, yet the resources available are often limited. Systems like BIRI can play a key role in bridging this gap by providing enhanced access to information across various domains, such as legal, educational, and healthcare, in the user's own language.

Furthermore, the findings from our thesis underscore the importance and applicability of conversational context and domain awareness in QA systems. The integration of conversation history into both the retrieval and and reader steps improves the system's overall performance and highlights the evolving role of AI in simulating natural human dialogues. The domain-

awareness component ensured that the system was able to handle specific subject matters more precisely, which could have broad applications in areas where precision matters. These insights can have broad applications in areas where interactive systems are required to maintain context over extended conversations precisely, such as scalable effective virtual assistance in customer service, healthcare, and finance industries.

7.6. Future Work

While our study provides a strong foundation for building Conversational OpenQA systems in resource-constrained languages, several areas remain open for future exploration. One of the key areas for future work is improving the quality of machine translation for low-resource languages. Despite the successes seen in our thesis, translation errors still pose a significant challenge, particularly in cases where high precision is required. Exploring semi-supervised learning methods or human-in-the-loop approaches could enhance the accuracy of machine-translated datasets, making them more reliable for model training.

Another promising direction for future research is the expansion of our work to other low-resource languages. Although Turkish served as a successful case study, languages with different linguistic structures, such as those with a varying degree of complex morphologies or other linguistic characteristics, may present unique challenges. Investigating the applicability of our approach across a broader set of languages will be critical for better understanding the generalizability of the methods proposed here. To facilitate this, Section 5.4 provides a recipe to generalize the method, offering guidelines for adapting the techniques to diverse linguistic settings, which will ultimately contribute to building more inclusive and robust Conversational OpenQA systems for other low-resource languages worldwide.

Another potential future direction is the extension of Conversational OpenQA systems to Spoken Question Answering (SpokenQA) systems as well. Unlike other low-resource domains, SpokenQA systems face unique challenges, such as errors introduced by automatic speech recognition (ASR) and the need for real-time, speech-to-text processing, etc. One of the primary bottlenecks in developing robust SpokenQA systems is the scarcity of large,

labeled spoken datasets, especially for resource-constrained languages [201]. The automatic data generation techniques employed in our thesis, particularly the use of machine translation to obtain QA datasets, could be highly beneficial in addressing the data scarcity issue also in SpokenQA by generating synthetic spoken question-answer pairs without requiring gold-training dataset as demonstrated by Menevşe et al. [201] for Turkish. This approach not only improves the performance of SpokenQA systems in languages with limited resources but also reduces the cost and time associated with manual gold data collection for voice-based QA systems in low-resource languages.

In addition, future work should focus on integrating native, domain-specific conversational datasets into the training process. While synthetically adapted datasets provided valuable insights and a strong baseline, a deeper exploration of conversational QA in its natural form nuances of cultural context and linguistic diversity, would likely yield more effective models. The development and collection of large-scale conversational datasets in resource-constrained languages should remain a priority for advancing the field even further.

Another promising area for future research is analyzing the impact of the number of conversation turns used as context in conversational systems. While our thesis leverages conversational history to enhance the accuracy of responses, the extent to which increasing or decreasing the number of preceding turns affects system performance remains unexplored. For instance, a system that incorporates only the most recent turn might miss important nuances established earlier in the conversation, leading to less accurate or incomplete responses. Evaluating fine-tuning strategies for the optimal number of conversation turns for context could lead to significant improvements in the system's ability to provide relevant and coherent answers.

Lastly, as large language models continue to advance and becomes more cost-effective and feasible in many aspects, the integration of generative capabilities into Conversational OpenQA systems also presents an exciting opportunity. Once its resource-intensive requirements are mitigated for low-resource languages, generative models could allow for more detailed and contextually rich answers, enhancing the user experience in dialogue systems.

Combining retrieval-based systems with cost-effective generative models could be a promising area of exploration in improving the overall quality of Conversational OpenQA systems in low-resource languages.

8. CONCLUSION

In this thesis, we explored the feasibility and efficiency of Conversational Open Question Answering task, particularly in low-resource languages. In Chapter 3, we developed a large Turkish Natural Language Inference (NLI) dataset through machine translation, demonstrating that machine translation (MT) can effectively address data scarcity in Turkish NLP studies. In Chapter 4 we extended our approach to create cost-effective Open Question Answering (OpenQA) systems for low-resource languages. In Chapter 5 delves into the conversational aspect of OpenQA systems for low-resource languages, showcasing the benefits of integrating conversational signals and addressing data scarcity challenges through repurposing existing datasets. Chapter 6 examines whether QA systems can genuinely express uncertainty for unanswerable questions. Chapter 7 discusses the overall findings and other insights.

As an artifact of our thesis, we present SQuAD-TR, a Turkish Question Answering dataset that is obtained by automatically translating SQuAD2.0. This resource facilitates building efficient and effective OpenQA systems in Turkish, measuring their improvements, and sets an example for other languages. Additionally, we release NLI-TR, the first large scale NLI dataset for Turkish.

In conclusion, we demonstrated the feasibility of building Domain-Aware Conversational Open Question Answering systems for resource-constrained languages, using Turkish as a primary case study in our thesis. Through the use of machine translation, weak supervision, and conversational signals, we were able to develop a robust Conversational OpenQA system capable of handling multi-turn interactions. We further showed that NLI model can be used to assist a QA model in recognizing its limits and respond with “I do not know”, enhancing its real-world reliability and robustness. Although there are limitations in terms of data quality and scalability, the findings of this research provide a solid foundation for future advancements in this area. With further exploration into better data generation techniques and the integration of cost-effective generative models, the future of Domain-Aware Conversational OpenQA systems in resource-constrained languages looks promising.

REFERENCES

1. Karpukhin, V., B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering", B. Webber, T. Cohn, Y. He and Y. Liu (Editors), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, Online, 2020.

2. Budur, E., R. Özçelik, D. Soylu, O. Khattab, T. Güngör and C. Potts, "Building Efficient and Effective OpenQA Systems for Low-Resource Languages", *Knowledge-Based Systems*, Vol. 302, p. 112243, 2024.

3. 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.

4. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention Is All You Need", *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

5. Young, S. J., "Probabilistic Methods in Spoken–Dialogue Systems", *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, Vol. 358, No. 1769, pp. 1389–1402, 2000.

6. Gao, J., M. Galley and L. Li, "Neural Approaches to Conversational AI", Y. Artzi and J. Eisenstein (Editors), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 2–7, Melbourne, Australia, 2018.

7. Dalton, J., S. Fischer, P. Owoicho, F. Radlinski, F. Rossetto, J. R. Trippas and H. Zamani, "Conversational Information Seeking: Theory and Application", *Proceedings of the 45th International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, p. 3455–3458, Madrid, 2022.

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. Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le and S. Petrov, “Natural Questions: A Benchmark for Question Answering Research”, *Transactions of the Association for Computational Linguistics*, Vol. 7, pp. 452–466, 2019.

10. 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 (Long Papers)*, pp. 1601–1611, Vancouver, 2017.

11. Bui, M.-Q., V. Tran, H.-T. Nguyen and L.-M. Nguyen, “How State-Of-The-Art Models Can Deal With Long-Form Question Answering”, M. L. Nguyen, M. C. Luong and S. Song (Editors), *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pp. 375–382, Hanoi, Vietnam, 2020.

12. Khattab, O., C. Potts and M. Zaharia, “Relevance-guided Supervision for OpenQA with ColBERT”, *Transactions of the Association for Computational Linguistics*, Vol. 9, pp. 929–944, 2021.

13. Lewis, P., B. Oguz, R. Rinott, S. Riedel and H. Schwenk, “MLQA: Evaluating Cross-lingual Extractive Question Answering”, D. Jurafsky, J. Chai, N. Schluter and J. Tetreault (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7315–7330, Online, 2020.

14. Mehdad, Y., M. Negri and M. Federico, “Towards Cross-Lingual Textual Entailment”, R. Kaplan, J. Burstein, M. Harper and G. Penn (Editors), *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 321–324, Los Angeles, California, 2010.

15. Negri, M. and Y. Mehdad, “Creating a Bi-lingual Entailment Corpus through Translations with Mechanical Turk: $100 for a 10-day Rush”, C. Callison-Burch and M. Dredze (Editors), *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pp. 212–216, Los Angeles, 2010.

16. Negri, M., L. Bentivogli, Y. Mehdad, D. Giampiccolo and A. Marchetti, “Divide and Conquer: Crowdsourcing the Creation of Cross-Lingual Textual Entailment Corpora”, R. Barzilay and M. Johnson (Editors), *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 670–679, Edinburgh, Scotland, UK., 2011.

17. Mehdad, Y., M. Negri and M. Federico, “Using Bilingual Parallel Corpora for Cross-Lingual Textual Entailment”, D. Lin, Y. Matsumoto and R. Mihalcea (Editors), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1336–1345, Portland, Oregon, USA, 2011.

18. Real, L., A. Rodrigues, A. Vieira e Silva, B. Albiero, B. Thalenberg, B. Guide, C. Silva, G. de Oliveira Lima, I. C. S. Câmara, M. Stanojević, R. Souza and V. de Paiva, “SICK-BR: A Portuguese Corpus for Inference”, *Computational Processing of the Portuguese Language: 13th International Conference*, p. 303–312, Canela, Brazil, 2018.

19. Rodrigues, R. C., J. R. da Silva, P. V. Q. de Castro, N. F. F. da Silva and A. da Silva Soares, “Multilingual Transformer Ensembles for Portuguese Natural Language Tasks”, *Proceedings of the Avaliação de Similaridade Semântica e Inferência Textual 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese*, pp. 27–38, 2020.

20. Negri, M., A. Marchetti, Y. Mehdad, L. Bentivogli and D. Giampiccolo, “Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization”, E. Agirre, J. Bos, M. Diab, S. Manandhar, Y. Marton and D. Yuret (Editors), **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 399–407, Montréal, Canada, 2012.

21. Negri, M., A. Marchetti, Y. Mehdad, L. Bentivogli and D. Giampiccolo, “Semeval-2013 Task 8: Cross-lingual Textual Entailment for Content Synchronization”, S. Manandhar and D. Yuret (Editors), *Proceedings of the Seventh International Workshop on Semantic Evaluation*, pp. 25–33, Atlanta, Georgia, USA, 2013.

22. Fonseca, E. R., L. B. dos Santos, M. Criscuolo and S. M. Aluísio, “Visao Geral da Avaliaçao de Similaridade Semântica e Inferência Textual”, *Linguamática*, Vol. 8, No. 2, pp. 3–13, 2016.

23. Real, L., E. Fonseca and H. G. Oliveira, “The Avaliação de Similaridade Semântica e Inferência Textual 2 Shared Task: A Quick Overview”, *International Conference on Computational Processing of the Portuguese Language*, pp. 406–412, Evora, Portugal, 2020.

24. Conneau, A., R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk and V. Stoyanov, “XNLI: Evaluating Cross-Lingual Sentence Representations”, E. Riloff, D. Chiang, J. Hockenmaier and J. Tsujii (Editors), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2475–2485, Brussels, Belgium, 2018.

25. 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*, pp. 4171–4186, Minneapolis, Minnesota, 2019.

26. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest and A. M. Rush, “HuggingFace’s Transformers: State-of-the-Art Natural Language Processing”, *arXiv preprint arXiv:1910.03771*, 2020.

27. Alsentzer, E., J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann and M. McDermott, “Publicly Available Clinical BERT Embeddings”, A. Rumshisky, K. Roberts,

S. Bethard and T. Naumann (Editors), *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 72–78, Minneapolis, Minnesota, USA, 2019.

28. Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So and J. Kang, "BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining", *Bioinformatics*, Vol. 36, No. 4, pp. 1234–1240, 2020.

29. Akın, A. A. and M. D. Akın, "Zemberek, an Open Source NLP Framework for Turkic Languages", 2007, `https://github.com/ahmetaa/zemberek-nlp`, accessed on June 17, 2024.

30. Ozturel, A., T. Kayadelen and I. Demirsahin, "A Syntactically Expressive Morphological Analyzer for Turkish", H. Vogler and A. Maletti (Editors), *Proceedings of the 14th International Conference on Finite-State Methods and Natural Language Processing*, pp. 65–75, Dresden, Germany, 2019.

31. Sak, H., T. Güngör and M. Saraçlar, "A Stochastic Finite-State Morphological Parser for Turkish", K.-Y. Su, J. Su, J. Wiebe and H. Li (Editors), *Proceedings of the Association for Computational Linguistics-International Joint Conference on Natural Language Processing 2009 Conference Short Papers*, pp. 273–276, Suntec, Singapore, 2009.

32. Sak, H., T. Güngör and M. Saraçlar, "Resources for Turkish Morphological Processing", *Language Resources and Evaluation*, Vol. 45, No. 2, pp. 249–261, 2011.

33. Senel, L. K., B. Ebing, K. Baghirova, H. Schuetze and G. Glavaš, "Kardeş-NLU: Transfer to Low-Resource Languages with the Help of a High-Resource Cousin – A Benchmark and Evaluation for Turkic Languages", Y. Graham and M. Purver (Editors), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1672–1688, St. Julian's, Malta, 2024.

34. Solak, A. and K. Oflazer, "Parsing Agglutinative Word Structures and Its Application to Spelling Checking for Turkish", *The 14th International Conference on Computational Linguistics*, 1992.

35. Hakkani-Tür, D. Z., K. Oflazer and G. Tür, "Statistical Morphological Disambiguation

for Agglutinative Languages", *Computers and the Humanities*, Vol. 36, pp. 381–410, 2002.

36. Baykara, B. and T. Güngör, "Abstractive Text Summarization and New Large-Scale Datasets for Agglutinative Languages Turkish and Hungarian", *Language Resources and Evaluation*, Vol. 56, No. 3, pp. 973–1007, 2022.

37. Goldman, O. and R. Tsarfaty, "Morphology Without Borders: Clause-Level Morphology", *Transactions of the Association for Computational Linguistics*, Vol. 10, pp. 1455–1472, 2022.

38. Meeres, Y. and T. A. Pirinen, "Vowel Harmony Viewed as Error-Correcting Code", A. Ettinger, E. Pavlick and B. Prickett (Editors), *Proceedings of the Society for Computation in Linguistics 2021*, pp. 313–322, Online, 2021.

39. Çakmak, M. T., S. Acar and G. Eryiğit, "Word Alignment for English-Turkish Language Pair", N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis (Editors), *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pp. 2177–2180, Istanbul, Turkey, 2012.

40. Safaya, A., E. Kurtuluş, A. Goktogan and D. Yuret, "Mukayese: Turkish NLP Strikes Back", S. Muresan, P. Nakov and A. Villavicencio (Editors), *Findings of the Association for Computational Linguistics 2022*, pp. 846–863, Dublin, Ireland, 2022.

41. 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.

42. Pan, Y., X. Li, Y. Yang and R. Dong, "Multi-Task Neural Model for Agglutinative Language Translation", S. Rijhwani, J. Liu, Y. Wang and R. Dror (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 103–110, Online, 2020.

43. Budur, E., R. Özçelik, T. Gungor and C. Potts, "Data and Representation for Turkish Natural Language Inference", B. Webber, T. Cohn, Y. He and Y. Liu (Editors), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 8253–8267, Online, 2020.

44. Kann, K., "Acquisition of Inflectional Morphology in Artificial Neural Networks With Prior Knowledge", A. Ettinger, G. Jarosz and J. Pater (Editors), *Proceedings of the Society for Computation in Linguistics 2020*, pp. 144–154, New York, New York, 2020.

45. Yirmibeşoğlu, Z. and G. Eryiğit, "Detecting Code-Switching Between Turkish-English Language Pair", W. Xu, A. Ritter, T. Baldwin and A. Rahimi (Editors), *Proceedings of the 2018 Empirical Methods in Natural Language Processing Workshop on Noisy User-Generated Text: The 4th Workshop on Noisy User-Generated Text*, pp. 110–115, Brussels, Belgium, 2018.

46. Cambazoglu, B. B., M. Sanderson, F. Scholer and B. Croft, "A Review of Public Datasets in Question Answering Research", *Special Interest Group on Information Retrieval Forum*, Vol. 54, No. 2, 2021.

47. 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.

48. Hill, F., A. Bordes, S. Chopra and J. Weston, "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations", *4th International Conference on Learning Representations*, San Juan, Puerto Rico, 2016.

49. Bajgar, O., R. Kadlec and J. Kleindienst, "Embracing Data Abundance", *5th International Conference on Learning Representations*, Toulon, France, 2017.

50. 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, pp. 784–789, Melbourne, Australia, 2018.

51. Kamath, A., R. Jia and P. Liang, "Selective Question Answering under Domain Shift", D. Jurafsky, J. Chai, N. Schluter and J. Tetreault (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5684–5696, Online, 2020.

52. 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.

53. Jiang, Y., S. Bordia, Z. Zhong, C. Dognin, M. Singh and M. Bansal, "HoVer: A Dataset for Many-Hop Fact Extraction And Claim Verification", T. Cohn, Y. He and Y. Liu (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2020*, pp. 3441–3460, Online, 2020.

54. Dunn, M., L. Sagun, M. Higgins, V. U. Guney, V. Cirik and K. Cho, "SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine", ArXiv:1704.05179, 2017.

55. Yang, Y., W.-t. Yih and C. Meek, "WikiQA: A Challenge Dataset for Open-Domain Question Answering", L. Màrquez, C. Callison-Burch and J. Su (Editors), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018, Lisbon, Portugal, 2015.

56. Lee, K., M.-W. Chang and K. Toutanova, "Latent Retrieval for Weakly Supervised Open Domain Question Answering", A. Korhonen, D. Traum and L. Màrquez (Editors), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, 2019.

57. Chandra, A., A. Fahrizain, Ibrahim and S. W. Laufried, "A Survey on non-English Question Answering Dataset", ArXiv:2112.13634, 2021.

58. Lim, S., M. Kim and J. Lee, “KorQuAD1.0: Korean QA Dataset for Machine Reading Comprehension”, ArXiv:1909.07005, 2019.

59. 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.

60. Efimov, P., A. Chertok, L. Boytsov and P. Braslavski, “SberQuAD – Russian Reading Comprehension Dataset: Description and Analysis”, *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference and Labs of the Evaluation Forum 2020, Proceedings*, p. 3–15, Thessaloniki, Greece, 2020.

61. 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.

62. 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.

63. 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, 2019.

64. Conneau, A. and G. Lample, “Cross-lingual language model pretraining”, *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 7057–7067, Red Hook, NY, USA, 2019.

65. Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, "Unsupervised Cross-Lingual Representation Learning at Scale", D. Jurafsky, J. Chai, N. Schluter and J. Tetreault (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, 2020.

66. 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.

67. 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.

68. Xue, L., N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua and C. Raffel, "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer", K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty and Y. Zhou (Editors), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 483–498, Online, 2021.

69. Ebrahimi, A., M. Mager, S. Rijhwani, E. Rice, A. Oncevay, C. Baltazar, M. Cortés, C. Montaño, J. E. Ortega, R. Coto-solano, H. Cruz, A. Palmer and K. Kann, "Findings of the AmericasNLP 2023 Shared Task on Machine Translation into Indigenous Languages", M. Mager, A. Ebrahimi, A. Oncevay, E. Rice, S. Rijhwani, A. Palmer and K. Kann (Editors), *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pp. 206–219, Toronto, 2023.

70. Mager, M., R. Bhatnagar, G. Neubig, N. T. Vu and K. Kann, "Neural Machine Translation for the Indigenous Languages of the Americas: An Introduction", M. Mager, A. Ebrahimi, A. Oncevay, E. Rice, S. Rijhwani, A. Palmer and K. Kann (Editors), *Pro-*

ceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas, pp. 109–133, Toronto, 2023.

71. Chen, W.-r. and M. Abdul-mageed, "Improving Neural Machine Translation of Indigenous Languages with Multilingual Transfer Learning", A. K. Ojha, C.-h. Liu, E. Vylomova, F. Pirinen, J. Abbott, J. Washington, N. Oco, V. Malykh, V. Logacheva and X. Zhao (Editors), *Proceedings of the Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages 2023*, pp. 73–85, Dubrovnik, Croatia, 2023.

72. Shode, I., D. I. Adelani, J. Peng and A. Feldman, "NollySenti: Leveraging Transfer Learning and Machine Translation for Nigerian Movie Sentiment Classification", A. Rogers, J. Boyd-Graber and N. Okazaki (Editors), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 986–998, Toronto, 2023.

73. Le, N. T. and F. Sadat, "Towards a Low-Resource Neural Machine Translation for Indigenous Languages in Canada", *Traitement Automatique des Langues*, Vol. 62, No. 3, pp. 39–63, 2021.

74. Zhu, F., W. Lei, C. Wang, J. Zheng, S. Poria and T.-S. Chua, "Retrieving and Reading: A Comprehensive Survey on Open-Domain Question Answering", ArXiv:2101.00774 [cs.AI], 2021.

75. Zhang, Q., S. Chen, D. Xu, Q. Cao, X. Chen, T. Cohn and M. Fang, "A Survey for Efficient Open Domain Question Answering", A. Rogers, J. Boyd-Graber and N. Okazaki (Editors), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 14447–14465, Toronto, 2023.

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

77. Nogueira, R., Z. Jiang, R. Pradeep and J. Lin, "Document Ranking with a Pretrained Sequence-to-Sequence Model", T. Cohn, Y. He and Y. Liu (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2020*, pp. 708–718, Online, 2020.

78. Li, C., A. Yates, S. MacAvaney, B. He and Y. Sun, “PARADE: Passage Representation Aggregation for Document Reranking”, ArXiv:2008.09093 [cs.IR], 2020.

79. Pradeep, R., R. Nogueira and J. Lin, “The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models”, ArXiv:2101.05667 [cs.IR], 2021.

80. Noraset, T., L. Lowphansirikul and S. Tuarob, “WabiQA: A Wikipedia-Based Thai Question-Answering System”, *Information Processing & Management*, Vol. 58, No. 1, p. 102431, 2021.

81. Guu, K., K. Lee, Z. Tung, P. Pasupat and M.-W. Chang, “REALM: Retrieval-Augmented Language Model Pre-Training”, *Proceedings of the 37th International Conference on Machine Learning*, pp. 3929–3938, Online, 2020.

82. Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”, *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 9459–9474, Red Hook, NY, USA, 2020.

83. Khattab, O. and M. Zaharia, “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT”, *Proceedings of the 43rd International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, p. 39–48, New York, 2020.

84. Yang, S. and M. Seo, “Designing a Minimal Retrieve-and-Read System for Open-Domain Question Answering”, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty and Y. Zhou (Editors), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5856–5865, Online, 2021.

85. Bajaj, P., D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary and

T. Wang, “MS MARCO: A Human Generated MAchine Reading COmprehension Dataset”, ArXiv:1611.09268, 2018.

86. Craswell, N., B. Mitra, E. Yilmaz, D. Campos and E. M. Voorhees, “Overview of the Text REtrieval Conference 2019 Deep Learning Track”, ArXiv:2003.07820 [cs.IR], 2020.

87. Bonifacio, L. H., I. Campiotti, V. Jeronymo, R. Lotufo and R. Nogueira, “mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset”, ArXiv:2108.13897, 2021.

88. Asai, A., J. Kasai, J. Clark, K. Lee, E. Choi and H. Hajishirzi, “XOR QA: Cross-lingual Open-Retrieval Question Answering”, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty and Y. Zhou (Editors), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 547–564, Online, 2021.

89. Yu, D., C. Zhu, Y. Fang, W. Yu, S. Wang, Y. Xu, X. Ren, Y. Yang and M. Zeng, “KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering”, S. Muresan, P. Nakov and A. Villavicencio (Editors), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 4961–4974, Dublin, Ireland, 2022.

90. Nie, L., W. Wang, R. Hong, M. Wang and Q. Tian, “Multimodal Dialog System: Generating Responses via Adaptive Decoders”, *Proceedings of the 27th Association for Computing Machinery International Conference on Multimedia*, p. 1098–1106, Nice, 2019.

91. Lozano, A., S. L. Fleming, C.-C. Chiang and N. Shah, *Clinfo.ai: An Open-Source Retrieval-Augmented Large Language Model System for Answering Medical Questions using Scientific Literature*, pp. 8–23, 2024.

92. Mao, Y., P. He, X. Liu, Y. Shen, J. Gao, J. Han and W. Chen, “Generation-Augmented Retrieval for Open-Domain Question Answering”, C. Zong, F. Xia, W. Li and R. Navigli

(Editors), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Long Papers)*, pp. 4089–4100, Online, 2021.

93. Wei, J., X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models", S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho and A. Oh (Editors), *Advances in Neural Information Processing Systems*, pp. 24824–24837, New Orleans, 2022.

94. Yao, S., J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models", *International Conference on Learning Representations*, Kigali, 2023.

95. Khattab, O., K. Santhanam, X. L. Li, D. Hall, P. Liang, C. Potts and M. Zaharia, "Demonstrate-Search-Predict: Composing Retrieval and Language Models for Knowledge-Intensive NLP", ArXiv:2212.14024, 2022.

96. Chase, H., "LangChain", 2022, `https://www.langchain.com`, accessed on June 17, 2024.

97. Liu, J., "LlamaIndex", 2022, `https://www.llamaindex.ai`, accessed on June 17, 2024.

98. McCann, B., N. S. Keskar, C. Xiong and R. Socher, "The Natural Language Decathlon: Multitask Learning as Question Answering", ArXiv:1806.08730, 2018.

99. Radford, A., J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models are Unsupervised Multitask Learners", 2019, `https://github.com/openai/gpt-2`, accessed on June 17, 2024.

100. Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language Models are Few-Shot Learners", H. Larochelle,

M. Ranzato, R. Hadsell, M. Balcan and H. Lin (Editors), *Advances in Neural Information Processing Systems*, pp. 1877–1901, 2020.

101. Khattab, O., A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia and C. Potts, "DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines", ArXiv:2310.03714, 2023.

102. Lee, H., A. Kedia, J. Lee, A. Paranjape, C. Manning and K.-G. Woo, "You Only Need One Model for Open-domain Question Answering", Y. Goldberg, Z. Kozareva and Y. Zhang (Editors), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3047–3060, Abu Dhabi, United Arab Emirates, 2022.

103. Qu, C., L. Yang, C. Chen, M. Qiu, W. B. Croft and M. Iyyer, "Open-Retrieval Conversational Question Answering", *Proceedings of the 43rd International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, p. 539–548, Online, 2020.

104. Choi, E., H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang and L. Zettlemoyer, "QuAC: Question Answering in Context", E. Riloff, D. Chiang, J. Hockenmaier and J. Tsujii (Editors), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2174–2184, Brussels, Belgium, 2018.

105. Qu, C., L. Yang, M. Qiu, W. B. Croft, Y. Zhang and M. Iyyer, "BERT with History Answer Embedding for Conversational Question Answering", *Proceedings of the 42nd International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, p. 1133–1136, Paris, 2019.

106. Fang, H.-C., K.-H. Hung, C.-W. Huang and Y.-N. Chen, "Open-Domain Conversational Question Answering with Historical Answers", Y. He, H. Ji, S. Li, Y. Liu and C.-H. Chang (Editors), *Findings of the Association for Computational Linguistics: Asia-Pacific Chapter of the Association for Computational Linguistics-International Joint Conference on Natural Language Processing 2022*, pp. 319–326, Online, 2022.

107. Adlakha, V., S. Dhuliawala, K. Suleman, H. de Vries and S. Reddy, "TopiOCQA: Open-domain Conversational Question Answering with Topic Switching", *Transactions of the Association for Computational Linguistics*, Vol. 10, pp. 468–483, 2022.

108. Li, Y., W. Li and L. Nie, "A Graph-Guided Multi-Round Retrieval Method for Conversational Open-domain Question Answering", ArXiv:2104.08443, 2021.

109. Coman, A., G. Barlacchi and A. de Gispert, "Strong and Efficient Baselines for Open Domain Conversational Question Answering", H. Bouamor, J. Pino and K. Bali (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2023*, pp. 6305–6314, Singapore, 2023.

110. Izacard, G. and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering", P. Merlo, J. Tiedemann and R. Tsarfaty (Editors), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 874–880, Online, 2021.

111. Yu, S., Z. Liu, C. Xiong, T. Feng and Z. Liu, "Few-Shot Conversational Dense Retrieval", *Proceedings of the 44th International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, p. 829–838, Online, 2021.

112. Krasakis, A. M., A. Yates and E. Kanoulas, "Zero-Shot Query Contextualization for Conversational Search", *Proceedings of the 45th International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, pp. 1880–1884, Madrid, 2022.

113. Schneider, P., M. Klettner, E. Simperl and F. Matthes, "A Comparative Analysis of Conversational Large Language Models in Knowledge-Based Text Generation", Y. Graham and M. Purver (Editors), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Short Papers)*, pp. 358–367, St. Julian's, Malta, 2024.

114. Baek, J., S. Jeong, M. Kang, J. Park and S. Hwang, "Knowledge-Augmented Language Model Verification", H. Bouamor, J. Pino and K. Bali (Editors), *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing, pp. 1720–1736, Singapore, 2023.

115. Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language Models are Few-Shot Learners", H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin (Editors), *Advances in Neural Information Processing Systems*, pp. 1877–1901, 2020.

116. Chowdhery, A., S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov and N. Fiedel, "PaLM: Scaling Language Modeling with Pathways", *Journal of Machine Learning Research*, Vol. 24, No. 240, pp. 1–113, 2023.

117. Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov and T. Scialom, "Llama 2: Open Foundation and Fine-Tuned Chat Models", ArXiv:2307.09288, 2023.

118. Uludoğan, G., Z. Balal, F. Akkurt, M. Turker, O. Gungor and S. Üsküdarlı, "TURNA: A Turkish Encoder-Decoder Language Model for Enhanced Understanding and Generation", *Findings of the Association for Computational Linguistics 2024*, pp. 10103–10117, Bangkok, Thailand, 2024.

119. Kesgin, H. T., M. K. Yuce, E. Dogan, M. E. Uzun, A. Uz, H. E. Seyrek, A. Zeer and M. F. Amasyali, "Introducing cosmosGPT: Monolingual Training for Turkish Language Models", ArXiv:2404.17336, 2024.

120. Ji, Z., N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto and P. Fung, "Survey of Hallucination in Natural Language Generation", *Association for Computing Machinery Computing Surveys*, Vol. 55, No. 12, pp. 1 – 38, 2023.

121. Wang, D. and L. Li, "Learning from Mistakes via Cooperative Study Assistant for Large Language Models", H. Bouamor, J. Pino and K. Bali (Editors), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10667–10685, Singapore, 2023.

122. Dou, Y., M. Forbes, R. Koncel-Kedziorski, N. A. Smith and Y. Choi, "Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text", S. Muresan, P. Nakov and A. Villavicencio (Editors), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 7250–7274, Dublin, Ireland, 2022.

123. Yao, Y., P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen and N. Zhang, "Editing Large Language Models: Problems, Methods, and Opportunities", H. Bouamor, J. Pino and K. Bali (Editors), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, Singapore, 2023.

124. Mueller, A., K. Narang, L. Mathias, Q. Wang and H. Firooz, "Meta-training with Demonstration Retrieval for Efficient Few-shot Learning", A. Rogers, J. Boyd-Graber and N. Okazaki (Editors), *Findings of the Association for Computational Linguistics 2023*, pp. 6049–6064, Toronto, 2023.

125. Reddy, S., D. Chen and C. D. Manning, "CoQA: A Conversational Question Answering Challenge", *Transactions of the Association for Computational Linguistics*, Vol. 7, pp. 249–266, 2019.

126. Elgohary, A., D. Peskov and J. Boyd-Graber, "Can You Unpack That? Learning to Rewrite Questions-in-Context", 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. 5918–5924, Hong Kong, China, 2019.

127. Altinok, D., "A Diverse Set of Freely Available Linguistic Resources for Turkish", A. Rogers, J. Boyd-Graber and N. Okazaki (Editors), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 13739–13750, Toronto, 2023.

128. Çöltekin, Ç., A. S. Doğruöz and Ö. Çetinoğlu, "Resources for Turkish natural language processing: A critical survey", *Language Resources and Evaluation*, Vol. 57, No. 1, pp. 449–488, 2023.

129. Uskudarli, S., M. Şen, F. Akkurt, M. Gürbüz, O. Gungor, A. Özgür and T. Güngör, "TULAP - An Accessible and Sustainable Platform for Turkish Natural Language Processing Resources", D. Croce and L. Soldaini (Editors), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 219–227, Dubrovnik, Croatia, 2023.

130. Ganitkevitch, J., B. Van Durme and C. Callison-Burch, "PPDB: The Paraphrase Database", L. Vanderwende, H. Daumé III and K. Kirchhoff (Editors), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 758–764, Atlanta, Georgia, 2013.

131. Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty and D. Zeman, "Universal Dependencies v1: A Multilingual Treebank Collection", N. Calzolari, K. Choukri,

T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk and S. Piperidis (Editors), *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pp. 1659–1666, Portorož, Slovenia, 2016.

132. Bojar, O., C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia and A. Tamchyna, "Findings of the 2014 Workshop on Statistical Machine Translation", O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post and L. Specia (Editors), *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, Baltimore, Maryland, USA, 2014.

133. Bowman, S. R., G. Angeli, C. Potts and C. D. Manning, "A large annotated corpus for learning natural language inference", L. Màrquez, C. Callison-Burch and J. Su (Editors), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, 2015.

134. Williams, A., N. Nangia and S. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference", M. Walker, H. Ji and A. Stent (Editors), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, 2018.

135. Munro, R., *Processing Short Message Communications in Low-Resource Languages*, Ph.D. Thesis, Stanford University, 2012.

136. Johnson, M., M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation", *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 339–351, 2017.

137. Hieber, F., T. Domhan, M. Denkowski, D. Vilar, A. Sokolov, A. Clifton and M. Post, "The Sockeye Neural Machine Translation Toolkit at Association for Machine Translation in the Americas 2018", C. Cherry and G. Neubig (Editors), *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Research Track)*, pp. 200–207, Boston, MA, 2018.

138. Tomasello, L., *Neural Machine Translation and Artificial Intelligence: What Is Left for the Human Translator?*, Ph.D. Thesis, Università di Padova, 2019.

139. Hieber, F., T. Domhan, M. Denkowski and D. Vilar, "Sockeye 2: A Toolkit for Neural Machine Translation", A. Martins, H. Moniz, S. Fumega, B. Martins, F. Batista, L. Coheur, C. Parra, I. Trancoso, M. Turchi, A. Bisazza, J. Moorkens, A. Guerberof, M. Nurminen, L. Marg and M. L. Forcada (Editors), *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pp. 457–458, Lisboa, Portugal, 2020.

140. Schweter, S., "BERTurk - BERT models for Turkish", 2020, `https://github.com/stefan-it/turkish-bert`, accessed on June 17, 2024.

141. Williams, A., N. Nangia and S. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

142. Young, P., A. Lai, M. Hodosh and J. Hockenmaier, "From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions", *Transactions of the Association for Computational Linguistics*, Vol. 2, pp. 67–78, 2014.

143. McGraw, K. O. and S. P. Wong, "Forming Inferences About Some Intraclass Correlation Coefficients.", *Psychological Methods*, Vol. 1, No. 1, p. 30, 1996.

144. Cicchetti, D. V., "Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology.", *Psychological Assessment*, Vol. 6, No. 4, p. 284, 1994.

145. Hallgren, K. A., "Computing Inter-Rater Reliability for Observational Data: an Overview and Tutorial", *Tutorials in Quantitative Methods for Psychology*, Vol. 8, No. 1, p. 23, 2012.

146. Krippendorff, K., "Estimating the Reliability, Systematic Error and Random Error of Interval Data", *Educational and Psychological Measurement*, Vol. 30, No. 1, pp. 61–70, 1970.

147. Büyük, O., "Context-Dependent Sequence-to-Sequence Turkish Spelling Correction", *Association for Computing Machinery Transactions on Asian and Low-Resource Language Information Processing*, Vol. 19, No. 4, pp. 1–16, 2020.

148. Kuyumcu, B., C. Aksakalli and S. Delil, "An Automated New Approach in Fast Text Classification (FastText): A Case Study for Turkish Text Classification without Pre-Processing", *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval*, p. 1–4, Tokushima, Japan, 2019.

149. Ozer, Z., I. Ozer and O. Findik, "Diacritic Restoration of Turkish Tweets with Word2vec", *Engineering Science and Technology, an International Journal*, Vol. 21, No. 6, pp. 1120–1127, 2018.

150. Can, B., "Unsupervised Learning of Allomorphs in Turkish", *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 25, No. 4, pp. 3253–3260, 2017.

151. Dehkharghani, R., Y. Saygin, B. Yanikoglu and K. Oflazer, "SentiTurkNet: a Turkish Polarity Lexicon for Sentiment Analysis", *Language Resources and Evaluation*, Vol. 50, No. 3, pp. 667–685, 2016.

152. Gulcehre, C., O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk and Y. Bengio, "On Using Monolingual Corpora in Neural Machine Translation", ArXiv:1503.03535, 2015.

153. Oflazer, K., "Two-Level Description of Turkish Morphology", *Literary and Linguistic Computing*, Vol. 9, No. 2, pp. 137–148, 1994.

154. Allauzen, C., M. Riley, J. Schalkwyk, W. Skut and M. Mohri, "OpenFst: A General and Efficient Weighted Finite-State Transducer Library", J. Holub and J. Žďárek (Editors), *Implementation and Application of Automata*, pp. 11–23, Prague, 2007.

155. Bellman, R., "On the Theory of Dynamic Programming", *Proceedings of the National Academy of Sciences*, Vol. 38, No. 8, pp. 716–719, 1952.

156. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, Cambridge, Massachusetts, 2016.

157. Kudo, T., “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates”, I. Gurevych and Y. Miyao (Editors), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 66–75, Melbourne, Australia, 2018.

158. Kudo, T. and J. Richardson, “Sentencepiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing”, ArXiv:1808.06226, 2018.

159. Sennrich, R., B. Haddow and A. Birch, “Neural Machine Translation of Rare Words with Subword Units”, K. Erk and N. A. Smith (Editors), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 1715–1725, Berlin, Germany, 2016.

160. Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman and N. A. Smith, “Annotation Artifacts in Natural Language Inference Data”, M. Walker, H. Ji and A. Stent (Editors), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Short Papers)*, pp. 107–112, New Orleans, Louisiana, 2018.

161. Shen, X., A. Asai, B. Byrne and A. De Gispert, “xPQA: Cross-Lingual Product Question Answering in 12 Languages”, S. Sitaram, B. Beigman Klebanov and J. D. Williams (Editors), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Industry Track)*, pp. 103–115, Toronto, 2023.

162. Pergola, G., E. Kochkina, L. Gui, M. Liakata and Y. He, “Boosting Low-Resource Biomedical QA via Entity-Aware Masking Strategies”, ArXiv:2102.08366, 2021.

163. Daniel, J. E., W. Brink, R. Eloff and C. Copley, “Towards Automating Healthcare Question Answering in a Noisy Multilingual Low-Resource Setting”, A. Korhonen, D. Traum and L. Màrquez (Editors), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 948–953, Florence, 2019.

164. Ghosh, S., C. K. R. Evuru, S. Kumar, S. Ramaneswaran, S. Sakshi, U. Tyagi and D. Manocha, “DALE: Generative Data Augmentation for Low-Resource Legal NLP”,

H. Bouamor, J. Pino and K. Bali (Editors), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8511–8565, Singapore, 2023.

165. Sun, K. and J. Pujara, "Low-Resource Financial QA with Case-Based Reasoning", *Knowledge Discovery and Data Mining Workshop on Robust NLP for Finance*, Long Beach, 2023.

166. Zheng, X., T. Liu, H. Meng, X. Wang, Y. Jiang, M. Rao, B. Lin, Y. Cao and Z. Sui, "DialogQAE: N-to-N Question Answer Pair Extraction from Customer Service Chatlog", H. Bouamor, J. Pino and K. Bali (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2023*, pp. 6540–6558, Singapore, 2023.

167. Darm, P., A. V. Miceli Barone, S. B. Cohen and A. Riccardi, "DISCOSQA: A Knowledge Base Question Answering System for Space Debris based on Program Induction", S. Sitaram, B. Beigman Klebanov and J. D. Williams (Editors), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Industry Track)*, pp. 487–499, Toronto, 2023.

168. Shen, X., S. Vakulenko, M. del Tredici, G. Barlacchi, B. Byrne and A. de Gispert, "Low-Resource Dense Retrieval for Open-Domain Question Answering: A Comprehensive Survey", ArXiv:2208.03197, 2022.

169. Levenshtein, V. I., "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals", *Soviet Physics Doklady*, Vol. 10, No. 8, pp. 707–710, 1966.

170. Robertson, S. and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond", *Foundations and Trends in Information Retrieval*, Vol. 3, No. 4, p. 333–389, 2009.

171. Nie, L., Y. Li, F. Feng, X. Song, M. Wang and Y. Wang, "Large-Scale Question Tagging via Joint Question-Topic Embedding Learning", *Association for Computing Machinery Transactions on Information Systems*, Vol. 38, No. 2, pp. 1–23, 2020.

172. Santhanam, K., O. Khattab, J. Saad-Falcon, C. Potts and M. Zaharia, "ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction", M. Carpuat, M.-C.

de Marneffe and I. V. Meza Ruiz (Editors), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, Seattle, 2022.

173. Santhanam, K., O. Khattab, C. Potts and M. Zaharia, "PLAID: An Efficient Engine for Late Interaction Retrieval", *Proceedings of the 31st Association for Computing Machinery International Conference on Information & Knowledge Management*, p. 1747–1756, Atlanta, 2022.

174. 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", ArXiv:1907.11692, 2019.

175. Clark, K., M.-T. Luong, Q. V. Le and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators", *International Conference on Learning Representations*, Online, 2019.

176. He, P., X. Liu, J. Gao and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention", *Proceedings of the International Conference on Learning Representations*, Online, 2021.

177. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest and A. Rush, "Transformers: State-of-the-Art Natural Language Processing", Q. Liu and D. Schlangen (Editors), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020.

178. Lin, J., X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep and R. Nogueira, "Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations", *Proceedings of the 44th Annual International Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval 2021*, pp. 2356–2362, Online, 2021.

179. Lin, J., M. Crane, A. Trotman, J. Callan, I. Chattopadhyaya, J. Foley, G. Ingersoll, C. Macdonald and S. Vigna, "Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge", *European Conference on Information Retrieval*, pp. 408–420, Springer, Padua, 2016.

180. Arslan, A., "DeASCIIfication Approach to Handle Diacritics in Turkish Information Retrieval", *Information Processing & Management*, Vol. 52, No. 2, pp. 326–339, 2016.

181. Johnson, J., M. Douze and H. Jégou, "Billion-Scale Similarity Search with GPUs", *IEEE Transactions on Big Data*, Vol. 7, No. 3, pp. 535–547, 2019.

182. Chen, D., A. Fisch, J. Weston and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions", R. Barzilay and M.-Y. Kan (Editors), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pp. 1870–1879, Vancouver, 2017.

183. Liu, J., Y. Lin, Z. Liu and M. Sun, "XQA: A Cross-lingual Open-domain Question Answering Dataset", A. Korhonen, D. Traum and L. Màrquez (Editors), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2358–2368, Florence, 2019.

184. Du, P., J.-Y. Nie, Y. Zhu, H. Jiang, L. Zou and X. Yan, "PReGAN: Answer Oriented Passage Ranking with Weakly Supervised GAN", ArXiv:2207.01762, 2022.

185. Chen, X., K. Lakhotia, B. Oguz, A. Gupta, P. Lewis, S. Peshterliev, Y. Mehdad, S. Gupta and W.-t. Yih, "Salient Phrase Aware Dense Retrieval: Can a Dense Retriever Imitate a Sparse One?", Y. Goldberg, Z. Kozareva and Y. Zhang (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2022*, pp. 250–262, Abu Dhabi, United Arab Emirates, 2022.

186. Vig, J., "A Multiscale Visualization of Attention in the Transformer Model", M. R. Costa-jussà and E. Alfonseca (Editors), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, Florence, 2019.

187. Scheible, R., F. Thomczyk, P. Tippmann, V. Jaravine and M. Boeker, “GottBERT: a Pure German Language Model”, ArXiv:2012.02110, 2020.

188. Martin, L., B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah and B. Sagot, “CamemBERT: a Tasty French Language Model”, D. Jurafsky, J. Chai, N. Schluter and J. Tetreault (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7203–7219, Online, 2020.

189. Le, H., L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier and D. Schwab, “FlauBERT: Unsupervised Language Model Pre-training for French”, 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. 2479–2490, Marseille, France, 2020.

190. Nguyen, V. H., H. T. Nguyen and V. Snasel, “Text normalization for named entity recognition in Vietnamese tweets”, *Computational social networks*, Vol. 3, pp. 1–16, 2016.

191. Le, K., H. Nguyen, T. Le Thanh and M. Nguyen, “VIMQA: A Vietnamese Dataset for Advanced Reasoning and Explainable Multi-hop Question Answering”, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk and S. Piperidis (Editors), *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6521–6529, Marseille, France, 2022.

192. Longpre, S., Y. Lu and J. Daiber, “MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering”, *Transactions of the Association for Computational Linguistics*, Vol. 9, pp. 1389–1406, 2021.

193. Clavié, B., “JaColBERT and Hard Negatives, Towards Better Japanese-First Embeddings for Retrieval: Early Technical Report”, ArXiv:2312.16144, 2023.

194. Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, “ALBERT: A

Lite BERT for Self-Supervised Learning of Language Representations.", *Proceedings of Eighth International Conference On Learning Representations*, 2020.

195. Liang, T., Y. Jiang, C. Xia, Z. Zhao, Y. Yin and P. S. Yu, "Multifaceted Improvements for Conversational Open-Domain Question Answering", ArXiv:2204.00266, 2022.

196. Ko, W.-J., A. El-Kishky, A. Renduchintala, V. Chaudhary, N. Goyal, F. Guzmán, P. Fung, P. Koehn and M. Diab, "Adapting High-resource NMT Models to Translate Low-resource Related Languages without Parallel Data", C. Zong, F. Xia, W. Li and R. Navigli (Editors), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Long Papers)*, pp. 802–812, Online, 2021.

197. Chen, A., Y. Sun, X. Zhao, R. Galindo Esparza, K. Chen, Y. Xiang, T. Zhao and M. Zhang, "Improving Low-resource Question Answering by Augmenting Question Information", H. Bouamor, J. Pino and K. Bali (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2023*, pp. 10413–10420, Singapore, 2023.

198. Ahmad, W., J. Chi, Y. Tian and K.-W. Chang, "PolicyQA: A Reading Comprehension Dataset for Privacy Policies", T. Cohn, Y. He and Y. Liu (Editors), *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2020*, pp. 743–749, Online, 2020.

199. Castelli, V., R. Chakravarti, S. Dana, A. Ferritto, R. Florian, M. Franz, D. Garg, D. Khandelwal, S. McCarley, M. McCawley, M. Nasr, L. Pan, C. Pendus, J. Pitrelli, S. Pujar, S. Roukos, A. Sakrajda, A. Sil, R. Uceda-Sosa, T. Ward and R. Zhang, "The TechQA Dataset", D. Jurafsky, J. Chai, N. Schluter and J. Tetreault (Editors), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1269–1278, Online, 2020.

200. Ziegler, I., A. Köksal, D. Elliott and H. Schütze, "CRAFT Your Dataset: Task-Specific Synthetic Dataset Generation Through Corpus Retrieval and Augmentation", ArXiv:2409.02098, 2024.

201. Ünlü Menevşe, M., Y. Manav, E. Arisoy and A. Özgür, “Dealing with Data Scarcity in Spoken Question Answering”, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti and N. Xue (Editors), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation 2024*, pp. 4449–4455, Torino, Italia, 2024.

APPENDIX A: ADDITIONAL EXPERIMENTS WITH NLI-TR

In this appendix, we present supplementary experimental results that expand upon the findings discussed in Chapter 3. These additional results focus on translation quality for NLI datasets in Turkish as well as evaluating various models on different Turkish NLI datasets. The experiments demonstrate the consistency and robustness of the NLI-TR dataset, further validating its use in resource-constrained language settings.

Table A.1. Sample translations from SNLI and MultiNLI into NLI-TR. Each premise is associated with a hypothesis from each of the three NLI categories.

		English	Turkish
SNLI	**Premise**	Several people are on stage preparing for a show.	Birkaç kişi sahnede gösteri için hazırlanıyor.
	Entailment	People are setting up for a show.	İnsanlar bir gösteri için hazırlanıyor.
	Contradiction	A house is being demolished.	Bir ev yıkılıyor.
	Neutral	A crew is getting ready for a rock concert.	Bir ekip rock konseri için hazırlanıyor.
MultiNLI	**Premise**	All rooms have color TV, alarm clock/radio, en-suite bathrooms, real hangers, and shower massage.	Tüm odalarda renkli TV, çalar saat/radyo, en-suite banyo, gerçek askılar ve duş masajı vardır.
	Entailment	All rooms also contain a ceiling fan and outlets for electronics.	Tüm odalarda ayrıca tavan vantilatörü ve elektronik prizler bulunmaktadır.
	Contradiction	You will not find a TV or alarm clock in any of the rooms.	Odaların hiçbirinde TV veya çalar saat bulunmamaktadır.
	Neutral	Color TVs, alarms, and hangers can be found in all rooms.	Tüm odalarda renkli TV'ler, alarmlar ve askılar bulunur.

The sample translations in Table A.1 illustrate the process of adapting English datasets, such as SNLI and MultiNLI, into NLI-TR through machine translation. Each premise is paired

with a hypothesis, reflecting the three NLI categories (entailment, contradiction, and neutral). These examples emphasize the alignment and consistency maintained during the translation process, serving as the foundation for evaluating Turkish Natural Language Inference tasks.

Table A.2. Accuracy of the cased models in Table 3.4 trained on the original SNLI and MultiNLI datasets in English. We used the same fine-tuning and evaluation procedures. BERT-En ranked the first and BERT-Multi ranked the second, emphasizing the importance of in-language training one-more time as in Section 3.3.1.

.

	SNLI	**MultiNLI**	
Model Name	**Test**	**Matched Dev**	**Mismatched Dev**
BERT-En	**90.13%**	**83.16%**	**83.95%**
BERT-Multi	89.02%	81.74%	82.13%
BERTurk	85.84%	75.16%	75.60%

The accuracy results of cased BERT models, trained on the original SNLI and MultiNLI datasets in English, are explored in Table A.2. These results extend the findings in Table 3.4 by training and finetuning on the models on the original datasets in English, rather than focusing exclusively on NLI-TR. It highlights the comparative performance of models trained with in-language versus cross-lingual data, offering additional insights into the benefits and limitations of each approach. These supplementary results underscore the value of in-language datasets for improving model performance, reinforcing the observations from earlier experiments.

Table A.3. Accuracy results of the models in Table 3.6 for machine translated XNLI.

Model Name	MultiNLI-TR		MultiNLI-TR^{XNLI}	
	XNLI-Dev-TR	**XNLI-Test-TR**	**XNLI-Dev-TR**	**XNLI-Test-TR**
BERT-En	70.11%	70.11%	68.42%	67.70%
BERT-Multi	75.85%	74.79%	74.69%	73.77%
BERTurk	**80.11%**	**79.52%**	**79.95%**	**78.40%**

We reproduce the results of the models in Table 3.6 for machine translated XNLI and we show the resuts in Table A.3. We note that, XNLI-Dev-TR, XNLI-Test-TR and MultiNLI-TR are translated with the same MT service, whereas MultiNLI-TR^{XNLI} used a different one. Though this might result in a positive bias for MultiNLI-TR models, we report the accuracy of MultiNLI-TR^{XNLI} models as well for the sake of completeness. The results in Table A.3 agree with the ones in Table 3.6 in Section 3.3.3, suggesting that machine translated sentences can be used to evaluate Turkish NLI models.

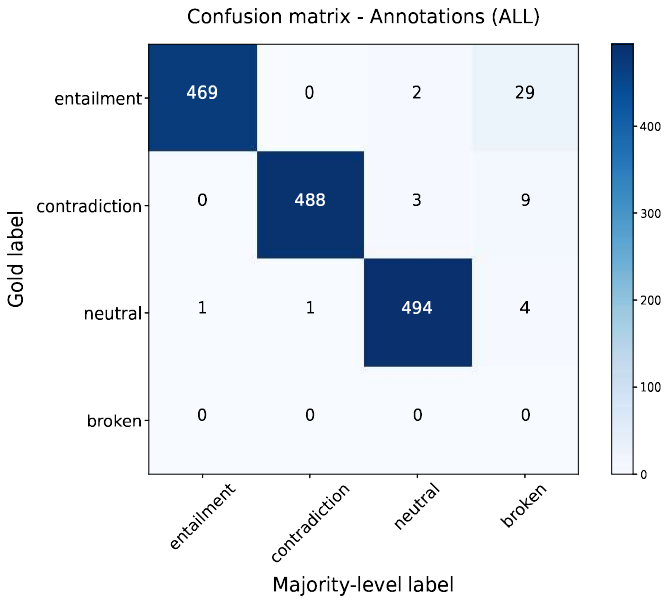


Figure A.1. Confusion matrix for the majority-level label consistency results of all annotations in Table 3.3. The label "broken" corresponds to the pairs which have either major translation error or no majority-level label.

The confusion matrix presented in Figure A.1 provides a detailed analysis of label consistency across annotations in the NLI-TR dataset. This figure extends the insights shared in Table 3.3, which evaluated the translation quality and label consistency of SNLI-TR and MultiNLI-TR datasets based on human assessment. By visualizing the agreement levels among annotators, Figure A.1 highlights specific areas of inconsistency, such as instances labeled as "broken" due to translation errors or the absence of majority-level agreement. These results not only complement the quantitative findings in the earlier table but also offer a clearer picture of how systematic errors in the dataset might impact model performance.

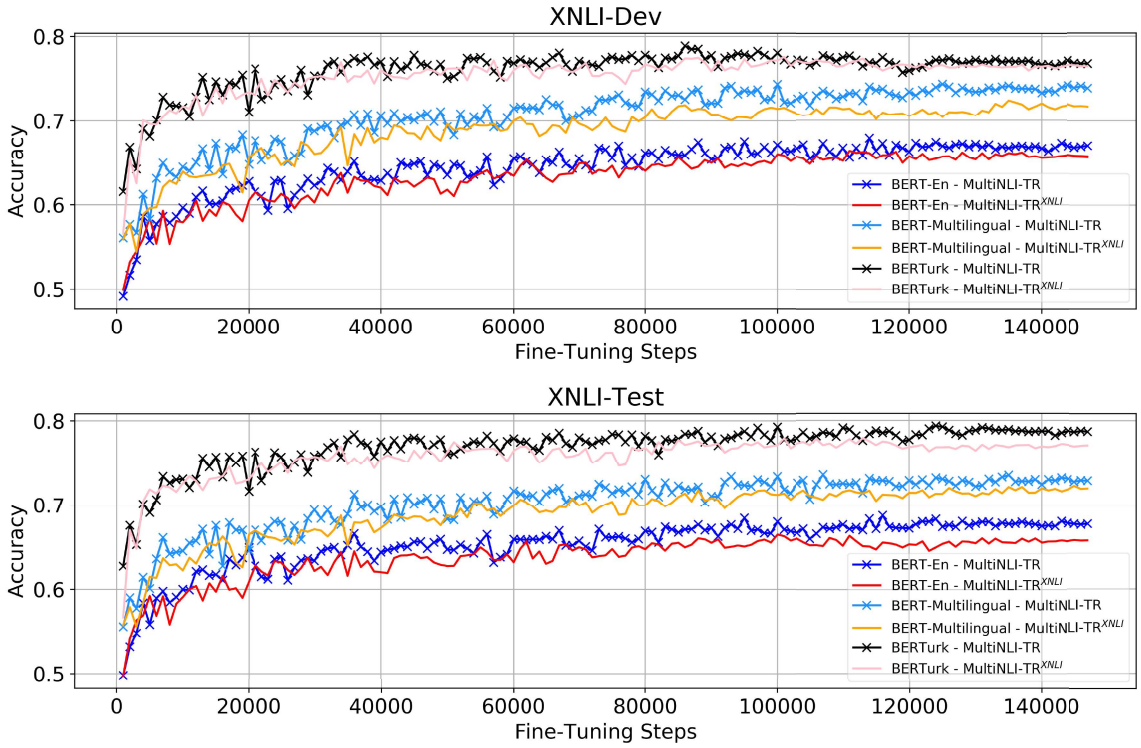


Figure A.2. XNLI-Dev and XNLI-Test accuracy of three transformer models trained on MultiNLI-TR and MultiNLI-TR$^{\text{XNLI}}$. The x-axis tracks the training set size. We find that models trained on MultiNLI-TR are superior to their MultiNLI-TR$^{\text{XNLI}}$ counterparts from the start of the training until the end.

The accuracy trends of three transformer models trained on MultiNLI-TR and MultiNLI-TR$^{\text{XNLI}}$datasets are visualized in Figure A.2, highlighting their performance across varying training set sizes. This analysis builds on the results in Table 3.6, which compared

the performance of NLI-TR when finetuned on two different machine-translated versions of MultiNLI dataset. By plotting the learning curves, the figure offers a detailed perspective on how models trained on MultiNLI-TR consistently achieve higher accuracy than those trained on MultiNLI-TR$^{\text{XNLI}}$, particularly in low-resource scenarios. These trends reinforce the importance of high-quality, in-language training data, aligning with earlier findings.

APPENDIX B: ALGORITHM AND FLOWCHART FOR BIRI

This appendix contains the algorithmic details and flowchart for the BIRI Conversational OpenQA system, which we introduced in Chapter 5. The algorithm and flowchart provide a clear, step-by-step overview of the system architecture, illustrating the information flow and interactions between its components. This additional information is crucial for understanding the operational aspects of BIRI.

The step-by-step process of the Conversational OpenQA system is formalized in Algorithm 1, detailing the logical flow of operations within the retriever and reader modules, as well as the integration of conversational signals. The algorithm aims to translate the conceptual overview illustrated in Figure B.1 into a precise procedural description.

Algorithm 1 Algorithm for the Conversational OpenQA system.

Input: $curQ$ — Current question, q_k

Input: $prevQs$ — Conversation history, $\{q_i\}_1^{k-1}$

$Cq_k \leftarrow$ CONCAT$(curQ, prevQs)$ ▷ Build contextualized question.

$passages \leftarrow$ RETRIEVE(Cq_k) ▷ Retrieve ranked list of candidate passages.

$\hat{p}_k \leftarrow passages[0]$ ▷ Select the first candidate passage.

$\hat{a}_k \leftarrow$ READ$(Cq_k, \hat{p}_k)$ ▷ Read candidate answer span from the selected passage.

return Candidate answer span $\hat{a}_k$

The flowchart depicted in Figure B.1 outlines the architecture and workflow of the Conversational OpenQA system developed in this study. The diagram serves as a detailed extension of the system overview provided in Figure 4.2, offering a step-by-step representation of the interaction between retriever, reader, and conversational context modules.

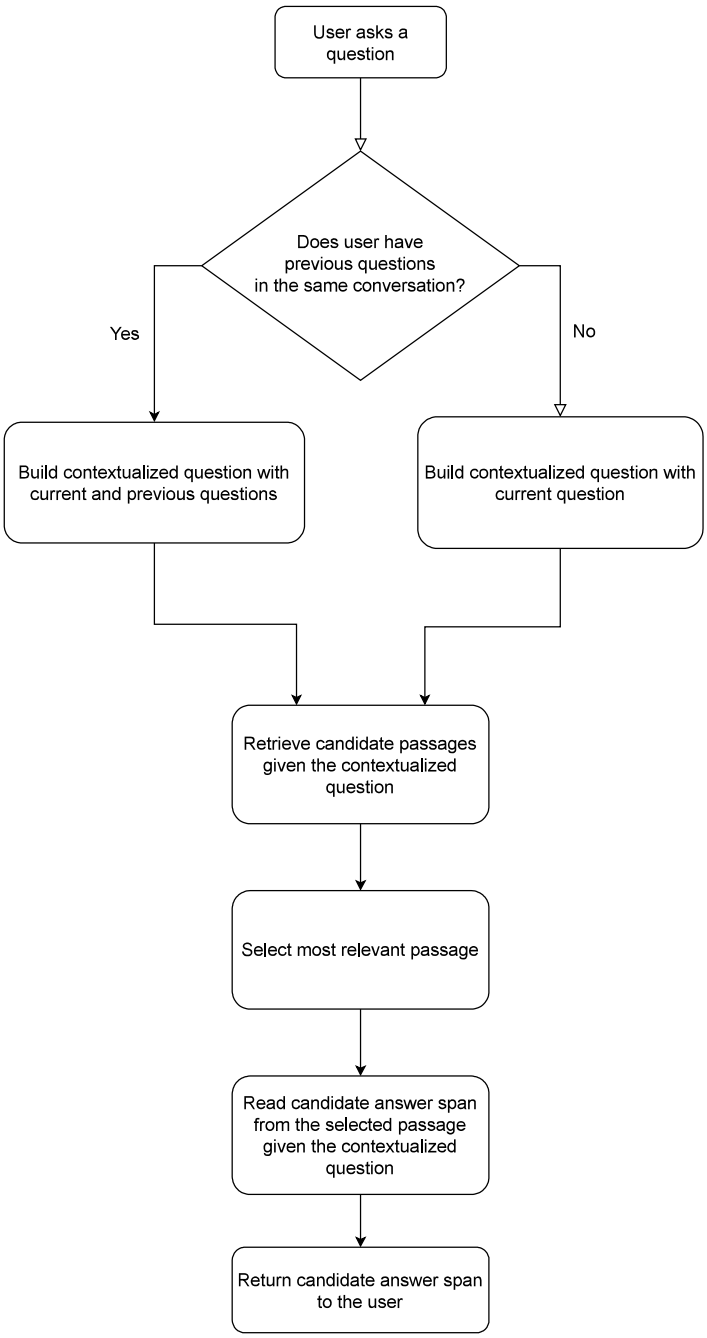


Figure B.1. The flowchart of the Conversational OpenQA system.

By illustrating the information flow and the incorporation of conversational signals at various stages, this flowchart emphasizes how the system adapts to the challenges of resource-constrained languages, providing a clearer understanding of the methodology.

APPENDIX C: DETAILED EXAMPLES FOR BIRI

In this appendix, we provide detailed examples of BIRI's performance in specific conversational contexts. These examples highlight the impact of incorporating conversational history into the system's retriever and reader components. By showcasing these detailed scenarios, we aim to give further insight into the practical benefits of conversational signals for improving OpenQA performance in resource-constrained languages.

In Table C.1, a query about the player with the most interceptions during the Super Bowl 50 season underscores the importance of leveraging conversation history. Specifically, in this particular example, the target passage is from the Wikipedia article titled "Super Bowl 50" with the current question focusing on the player with the most interceptions for the season in the context of the Super Bowl 50 game. Without utilizing conversation history (Retriever: ColBERT-QA), the retrieved passage lacks information specific to Super Bowl 50, therefore presenting an answer applicable to all games. Consequently, the reader (BERTurk) locates an incorrect answer (Paul Krause), as the predicted answer pertains to all games rather than the targeted Super Bowl 50 context. On the other hand, when leveraging conversation history (Retriever: ConvCOLBERT-QA), the correct passage is retrieved, enabling the reader (ConvBERTurk) to locate the correct answer (Kurt Coleman).

Table C.1. An example emphasizing the positive effect of using conversational history in the retrieval process for locating the relevant paragraph to respond to a query.

	Turkish	**English**
Current Question	Hangi oyuncu sezonun en çok top kapmasına sahiptir?	Which player had the most interceptions for the season?
Previous Questions (Latest first)	Thomas Davis kaç mecburi oyuncuyu düşürme almıştır?	How many forced fumbles did Thomas Davis have?
	Pro Bowl için kaç tane Panthers savunma oyuncusu seçilmiştir?	How many Panthers defense players were selected for the Pro Bowl?
	Panthers'lere sack etmede kim önderlik etmektedir?	Who led the Panthers in sacks?
	Panthers savunması 2015 yılında kaç top çalma ile kayda geçmiştir?	How many interceptions are the Panthers defense credited with in 2015?
	Bu sezon takımdaki en fazla sack etmeyi kim kaydetmiştir?	Who registered the most sacks on the team this season?
	Josh Norman kaç tane top çalmıştır?	How many balls did Josh Norman intercept?
	Luke Kuechly kaç tane top çalma kaydetmiştir?	How many tackles did Luke Kuechly register?
	Jared Allen'ın kaç tane kariyer sack edişi vardır?	How many career sacks did Jared Allen have?
	Panthers savunması kaç sayı bırakmıştır?	How many points did the Panthers defense surrender?
Retrieved Passages	• ... • çok top kapma: 81, ⟨**Paul Krause**⟩ (Reader answer), 1964–79 • Bir sezonda en çok top kapma: 14, Dick "Night Train" Lane, 1952 • Bir maçta en çok top kapma: 4, 19 oyuncu rekora ortaktır, en güncel olan DeAngelo Hall'dır (24 Ekim 2010) • En az bir top kapma yapılan arka arkaya maç sayısı: 8, Tom Morrow, 1962–63 • Kariyeri boyunca en çok top kapma sonrası mesafe kazanan: 1,590 yarda, Ed Reed, 2002–2013 • Kariyeri boyunca top kapma sonucu ortalama kapma başına en çok mesafe kazanan (asgari 30 top kapma) • ... Retriever: ColBERT-QA Source: `Wikipedia`	• ... • Most interceptions, career: 81, ⟨**Paul Krause**⟩ (Reader answer), 1964–79 • Most interceptions, season: 14, Dick "Night Train" Lane, 1952 • Most interceptions, game: 4, by 19 players, most recently DeAngelo Hall (October 24, 2010) • Most consecutive games with an interception: 8, Tom Morrow, 1962–63 • Most interception return yards, career: 1,590, Ed Reed, 2002–2013 • Most yards per interception return, career (minimum 30 interceptions) • ...
	... ikincisi olarak gösterilen Pro Bowl güvenliği için kariyerinin en yükseği olarak 7 top kapma ile takımı sürükleyen ⟨**Kurt Coleman**⟩ (Reader answer)'a ve 88 top çalma sayısıyla ve Pro Bowl köşe savunmacısı sezon boyunca bir kapalı köşe içinde gelişen ikisi kale çizgisini geçişten dönen dört top kapma geliştiren Josh Norman'a rol verdi. Retriever: ConvCOLBERT-QA Source: `XQuAD-TR`	... secondary featured Pro Bowl safety ⟨**Kurt Coleman**⟩ (Reader answer), who led the team with a career high seven interceptions, while also racking up 88 tackles and Pro Bowl cornerback Josh Norman, who developed into a shutdown corner during the season and had four interceptions, two of which were returned for touchdowns.

In Table C.2, we show an example illustrating the limitations of relying on conversational history during the retrieval process. In this particular example, the current question asks about the year Joseph Priestley recognized oxygen. Without conversational history (Retriever: ColBERT-QA), the retrieved passage directly mentions Priestley's discovery of oxygen in 1774, helping the reader locate the correct answer. However, when conversational history is used (Retriever: ConvCOLBERT-QA), the retrieved passage includes additional information about Carl Wilhelm Scheele's earlier discovery, which introduces ambiguity for the reader in identifying the precise answer. This example highlights how conversational history, while useful in most cases, can occasionally retrieve passages that complicate the task of the reader. It also emphasizes that strong retriever metrics alone do not necessarily translate into improved reader performance, or fully capture the overall effectiveness of an OpenQA system.

Table C.2. An example demonstrating the limitations of using conversational history in the retrieval process for locating a relevant passage for a question.

	Turkish	**English**
Current Question	Joseph Priestley hangi yılda oksijenin farkına vardı?	In what year did Joseph Priestley recognize oxygen?
Previous Question	Carl Wilhelm Scheele oksijeni ne zaman keşfetti?	When did Carl Wilhelm Scheele discover oxygen?
Retrieved Passages	• . . . • 1619 - İlk Afrikalı köleler, Virginia'nın Jamestown şehrine getirildiler. • 1773 - Deniz Harp Okulu (Tersane Hendesehanesi), Cezayirli Hasan Paşa tarafından İstanbul Kasımpaşa'da açıldı. • ⟨ **1774** ⟩ - İngiliz kimyacı Joseph Priestley, oksijen gazını (dioxygène, O2) keşfetti. (Reader answer) • 1798 - Nil Savaşı: Amiral Horatio Nelson komutasındaki Büyük Britanya Donanması, Abukir Körfezi'nde Fransız Donanması'nı yendi. • 1834 - Britanya İmparatorluğu'nda kölelik kaldırıldı. • 1840 - Ceride-i Havadis gazetesinin ilk sayısı basıldı. • 1876 - Colorado, 38. eyalet olarak ABD'ne kabul edildi. • . . . Retriever: ColBERT-QA Source: `Wikipedia`	• . . . • 1619 - The first African slaves were brought to Jamestown, Virginia. • 1773 - The Naval Academy (Tersane Hendesehanesi) was established in Istanbul's Kasımpaşa district by Algerian Hasan Pasha. • ⟨ **1774** ⟩ - English chemist Joseph Priestley discovered oxygen gas (dioxygen, O2). (Reader answer) • 1798 - Battle of the Nile: The British Royal Navy, commanded by Admiral Horatio Nelson, defeated the French Navy at Aboukir Bay. • 1834 - Slavery was abolished in the British Empire. • 1840 - The first issue of the Ceride-i Havadis newspaper was published. • 1876 - Colorado was admitted as the 38th state of the United States. • . . .
	Oksijen, birbirinden bağımsız olarak, Uppsala'da ⟨ **1773** ⟩ ya da önceki yıllarda Carl Wilhelm Scheele ve Wiltshire'da 1774 yılında Joseph Priestley tarafından keşfedildi, ancak çalışmaları ilk yayınlandığı için Priestley'e sıklıkla öncelik verilmektedir. Oksijen adı, 1777'de, oksijenle ilgili deneyleri o zamanki popüler yanma ve korozyonun flojiston kuramını geçersiz kılmaya yardımcı olan Antoine Lavoisier tarafından icat edildi. Adı, Yunan kökenli, asitlerin ekşi tadına atıf yapan oksys, "asit", kelime anlamıyla "keskin" ve -genes, "üretici", kelime anlamıyla "sebep. . . (Reader answer) Retriever: ConvCOLBERT-QA Source: `XQuAD-TR`	Oxygen was discovered independently by Carl Wilhelm Scheele, in Uppsala, in ⟨ **1773** ⟩ or earlier, and Joseph Priestley in Wiltshire, in 1774, but Priestley is often given priority because his work was published first. The name oxygen was coined in 1777 by Antoine Lavoisier, whose experiments with oxygen helped to discredit the then-popular phlogiston theory of combustion and corrosion. Its name derives from the Greek roots oxys, "acid", literally "sharp", referring to the sour taste of acids and - -genes, "producer", literally "begetter. . . (Reader answer)

An example showcasing the retrieval of the same target passage by both ColBERT-QA and ConvCOLBERT-QA is illustrated in Table C.3. In this example, the reader employing conversation history (ConvBERTurk) accurately identifies the target answer, whereas the one not utilizing conversation history (BERTurk) fails to locate the correct answer.

Table C.3. An example highlighting the positive impact of incorporating conversational history (previous questions) in the reader to find the answer based on a given question and the retrieved passage.

	Turkish	**English**
Current Question	Bağışıklık sisteminin hangi kısmının evrimi gerçekçenelilerin evrimsel atasında meydana gelmiştir?	Evolution of what part of the immune system occurred in the evolutionary ancestor of jawed vertebrates?
Previous Questions (Latest first)	İlkel çenesiz omurgalılar ne olarak adlandırılan bir dizi reseptöre sahiptir?	Primitive jawless vertebrates possess an array of receptors referred to as what?
	İlkel çenesiz omurgalıların iki örneği nedir?	What are two examples of primitive jawless vertebrates?
	Uyumlayıcı bağışıklık sisteminin hangi molekülleri sadece gerçekçenelilerde bulunur?	What molecules of the adaptive immune system only exist in jawed vertebrates?
Retrieved Passages	⟨**Uyumlayıcı bağışıklık sisteminin**⟩ (ConvBERTurk answer) evrimi, gerçekçenelilerin bir atasında meydana geldi. Uyumlayıcı bağışıklık sistemin klasik moleküllerinin çoğu (örneğin, immünoglobulinler ve T hücre reseptörleri) yalnızca gerçekçenelilerde bulunur. Fakat, farklı bir ⟨ **lenfosit** ⟩ (BERTurk answer) türevi molekülü ilk defa bofa balığı ve balık asalağı gibi çenesiz omurgalılarda keşfedilmiştir. Bu hayvanlar, gerçekçenelilerin antijen reseptörleri gibi, az sayıda (bir veya iki) genden üretilen, değişken lenfosit reseptörleri (VLR'ler) adında çok sayıda moleküle sahiptir. Bu moleküllerin patojenik antijenleri, antikorlara benzer bir şekilde ve aynı derecede özgünlük ile bağladığına inanılmaktadır.· · · Retrievers: ColBERT-QA, ConvCOLBERT-QA Source: XQuAD-TR	Evolution of the ⟨**adaptive immune system**⟩ (ConvBERTurk answer) occurred in an ancestor of the jawed vertebrates. Many of the classical molecules of the adaptive immune system (e.g., immunoglobulins and T cell receptors) exist only in jawed vertebrates. However, a distinct ⟨ **lymphocyte** ⟩ (BERTurk answer)-derived molecule has been discovered in primitive jawless vertebrates, such as the lamprey and hagfish. These animals possess a large array of molecules called Variable lymphocyte receptors (VLRs) that, like the antigen receptors of jawed vertebrates, are produced from only a small number (one or two) of genes. These molecules are believed to bind pathogenic antigens in a similar way to antibodies, and with the same degree of specificity.

While it is not entirely evident how the reader employs the conversational signal to enhance answer prediction, our hypothesis is that the reader has access to more information to capture the information need when conversational signals are leveraged. Consequently, its ability to locate the answer is likely improved, leading to greater accuracy.

Table C.4. An example where incorporating conversational history causes the reader to select an incorrect answer span, while the reader without conversational signals accurately identifies the correct answer.

	Turkish	**English**
Current Question	Q'yu hangi bakımdan tamamlamak gerçek sayıların alanını oluşturur?	Completing Q with respect to what will produce the field of real numbers?
Previous Questions (Latest first)	Genel küresel alanlar nereye veya nereden aktarılabilirler?	To what may general global fields be transferred to or from?
	Bir sayı p ile çarpıldığında norma ne olur?	What happens to the norm when a number is multiplied by p?
Retrieved Passages	Özellikle, bu norm, olağan mutlak değere (sonsuz asal olarak da ifade edilir) tam bir zıtlık içinde bir sayı p ile çarpıldığında küçülür. ⟨ **Mutlak değer** ⟩ (BERTurk answer) bakımından Q'yu tamamlamak (genel hatlarıyla boşlukları doldurmak) gerçek sayılar alanını oluştursa da ⟨ **p-sel** $\mid - \mid$**p normu** ⟩ (ConvBERTurk answer) bakımından tamamlamak, p-sel sayılar alanını oluşturur. Ostrowski teoremine göre, bunlar temelde Q'yu tamamlamanın olası tüm yöntemleridir. Q veya daha genel küresel alanlarla ile ilgili bazı aritmetik sorular tamamlanan (veya yerel) alanlara ileri ve geri olacak şekilde aktarılabilir. Bu yerel-küresel prensip asalların sayılar teorisindeki öneminin tekrar altını çizmektedir. Retrievers: ColBERT-QA, ConvCOLBERT-QA Source: XQuAD-TR	In particular, this norm gets smaller when a number is multiplied by p, in sharp contrast to the usual absolute value (also referred to as the infinite prime). While completing Q (roughly, filling the gaps) with respect to ⟨**the absolute value**⟩ (BERTurk answer) yields the field of real numbers, completing with respect to ⟨**the p-adic norm** $\mid - \mid$**p** ⟩ (ConvBERTurk answer) yields the field of p-adic numbers. These are essentially all possible ways to complete Q, by Ostrowski's theorem. Certain arithmetic questions related to Q or more general global fields may be transferred back and forth to the completed (or local) fields. This local-global principle again underlines the importance of primes to number theory.

This example in in Table C.4 shows that leveraging conversational history in the reader, while often beneficial, can occasionally lead to errors. In this example, the reader incorporating conversational signals (ConvBERTurk) selects an incorrect answer span (the p-adic norm $\mid - \mid$p), whereas the reader without conversational history (BERTurk) correctly identifies the answer (the absolute value). We hypothesize that this issue arises when specific words (e.g., "p", "norm") from previous questions distract the model. Although we observed this issue in only a few cases, it highlights how conversational signals can inadvertently introduce noise and distract the current question's focus, and ultimately affect the accuracy of the prediction.