TURKISH QUESTION GENERATION MODEL

Senior Design Project I

Alp Gokcek   **Erdal Sidal Dogan**

2020

MEF UNIVERSITY
FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

TURKISH QUESTION GENERATION MODEL

Senior Design Project I

Alp Gokcek   Erdal Sidal Dogan

Advisor: Asst. Prof. Seniz Demir

2020

MEF UNIVERSITY
FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

Project Title   : TURKISH QUESTION GENERATION MODEL
Student(s) Name   : Alp Gokcek, Erdal Sidal Dogan
Date   : 04/12/2020

I hereby state that the design project prepared by Alp Gokcek, Erdal Sidal Dogan has been completed under my supervision. I accept this work as a "Senior Design Project".

04/12/2020
Asst. Prof. Seniz Demir

I hereby state that I have examined this senior design project by Alp Gokcek, Erdal Sidal Dogan . This work is acceptable as a "Senior Design Project".

04/12/2020
Prof. Muhittin Gokmen
Head of the Department of
Computer Engineering

ACADEMIC HONESTY PLEDGE

In keeping with MEF University Student Code of Conduct, I pledge that this work is my own and that I have not received inappropriate assistance in its preparation. I further declare that all resources are explicitly cited.

NAME	DATE	SIGNATURE
Alp Gokcek	04/12/2020	
Erdal Sidal Dogan	04/12/2020	

ABSTRACT

TURKISH QUESTION GENERATION MODEL

Alp Gokcek   Erdal Sidal Dogan

MEF UNIVERSITY
Faculty of Engineering
Department of Computer Engineering

Advisor: Asst. Prof. Seniz Demir

DECEMBER, 2020

In the last decade, computers and software we use has gained humanitarian characteristics. We talk with them, create chatbots that answer the question of the user as a real assistant and many more. All of this features are invigorated by the Artificial Intelligence research which has showed significant improvements over the past decade.

With the emergence of the Deep Learning (DL) systems, other sub-fields of AI such as Natural Language Processing (NLP) made leaps in short amount of time as well. Today, results of the studies of these fields are ubiquitous.

Development of such applications require extensively large datasets for the training of DL models. These applications are helping us in many ways. They enable us to solve our problems with less effort in shorter amount of time, reduce the work loads on humans etc.

Development such applications require extensively large datasets. Datasets are crucial for a DL model to be trained. Capability and the reliability of these applications are highly dependent on the data they have been trained on.

In this project, we aim to develop a Question Generation Model in Turkish

Language.

Furthermore, for an application to be developed in any language, training dataset must contain the examples from that language as well. Sourcing such data in languages such as English, German, French etc. is not a hard task. However, labeled data in Turkish language is scarce, hence, development of smart applications such as question bots, personal assistants etc. in Turkish requires significantly more effort comparing to other languages.

---

[1]Massive Open Online Courses

ÖZET

TÜRKÇE SORU OLUŞTURMA MODELİ

Alp Gokcek   Erdal Sidal Dogan

MEF ÜNİVERSİTESİ
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Dr. Şeniz Demir

ARALIK, 2020

Bu projenin amacı Wikipedia gibi mevcut çevrimiçi kaynakları kullanarak Türkçe Soru Üretme Sistemi geliştirmektir. Soru Üretme Sistemleri, verilen metin girişinden çeşitli mantıksal sorular üretebilen sistemlerdir. Bu sistemler, sohbet robotları, otomatik değerlendirme sistemleri vb. birçok bilgisayar uygulamasında yaygın olarak kullanılmaktadır. Bu tip uygulamaların Türkçe destekli olarak geliştirilmesinde çok önemli bir yeri bulunan Türkçe veri seti örneği bulunmadığından,. Bunun birlikte, QG araştırması İngilizce gibi diğer dillerde çoktan kayda değer bir seviyeye ulaştı ve bu görev için mevcut veri kümeleri var. Örneğin, Stanford Üniversitesi tarafından SQuAD, bu görev için İngilizce olarak kullanılan iyi bilinen veri kümelerinden biridir.

Bu projede, orta büyüklükte ilk Türkçe Soru Üretme Veri Kümesini oluşturmayı hedefliyoruz. Bu adım, bu projede uygulayacağımız öğrenmeye dayalı çözümler geliştirmek için gereklidir. Çeşitli formlar (Kim-Ne-Ne Zaman vb.) Üretme soruları için farklı makine öğrenimi tabanlı teknikler uygulamayı ve toplayacağımız veri kümesindeki performanslarını değerlendirmeyi planlıyoruz.

Anahtar Kelimeler: Doğal Dil İşleme, Derin Öğrenme, Duygu Analizi

Table of Contents

LIST OF FIGURES

List of Tables

List of Abbreviations

1. INTRODUCTION

Artificial Intelligence is one of the promising research fields of the computer science. Scientists and researchers are working for developing novel or more efficient ways to teach computers how to achieve humanly-kinds of tasks.

Over time, developments in this area yield to whole new class of possibilities and of course problems that require brand new methodologies to be applied for reaching to the solution. With the emergence of Machine Learning, computers became to be able to parse data, learn from the data and apply their findings from it in a way that they are told to.

In the recent years, researchers came up with a more niche way of handling Machine Learning (ML) tasks, creating a structure that imitates the Neurons in our brains. This new subclass of ML is named Deep Learning (DL). Compared to ML, it is capable of handling much more complex task due to its multi-layered structure.

Developments in this area enabled researches from numerous fields to move their research a step further. Today, Deep Learning is ubiquitous in computer applications. It is utilized for Computer Vision, Natural Language Processing, Biotechnology and so on. In this project, methods and techniques that are results of the studies conducted in the research area of Natural Language Processing are used heavily along with various Machine Learning and Deep Learning models. Consequently, Turkish Question Generation is invigorated by the state-of-art Deep Learning and Language Models.

Turkish Question Generation Model aims to generate natural language questions from a given content, such as paragraphs, where the questions can be answered solely by the content. As the authors, we find this task worth tackling since there are not many examples of comprehensive studies on the topic and the outcomes of this project will empower development of more-capable-than-ever Question Answering Systems for Turkish Language.

1

1.1. Motivation

As humans we are constantly learn, adapt to changes or make decisions on everyday basis. Even if we don't realize, we usually are dependent on other humans and their knowledge and expertise even for most trivial things we do in our daily lives.

For instance, a student feels in need for an instructor or others than can address his/her questions about a topic, an individual may be in a hurry to find out whether he/she can perform SWIFT transaction from his bank, or simply a person can ask about the weather forecast to the voice assistant on a smartphone. All of these examples are having a common point that they are all based on Question Answering (QA) Systems.

As we have discussed, QA Systems are being getting more and more space in our lives and presents wide range of opportunities. With these systems, we are able to get answers to our question at no time, without any human interaction at all. This feature enables humans to reach to even the most isolated information in the context a matter of time upon request.

Unfortunately, given that almost every human spoken language has specific set of rules, grammar and vocabulary, QA systems needs to be developed for each language separately. Development of such systems are not easiest thing to do, first of all, even if the developer(s) has all the competency in the technical skills required, there is a need for clear, labeled, reliable and excessively large dataset. The dataset must contain questions and possible answers to that question. Creating such dataset manually is almost unattainable, therefore there is a need for a system that can automate this process. Question Generation systems are perfect for meeting such need. Output of the QG system can be used as the dataset for development.

On top of that, QG systems can be utilized in the education field also. Given a passage, instructor may want to create various questions for the class automatically without putting any effort. Same application can be used by a student for practicing the learnings from a section.

1.2. Broad Impact

1.2.1. Global Impact of the Solution

Online courses are gaining getting more and more popular each and every year. While their high-quality content available on web, for those people who are not able to understand English or mainstream languages of a specific subject, it is harder to keep up with the recent developments in the area. Up to this point, MOOC platforms offer translation and subtitles for such users. The Question Generation systems, along with Question Answering systems might be used for the purposes of creating an artificial interactive environment between the instructor and the student. Consequently, helping the information to spread around the world.

1.2.2. Economic Impact of the Solution

This project enables us to move faster to the point where we can automate almost every Question-Answer process where the data is assumed to be available to computer system, there is not an ambiguity amongst possible answers and the answers are static and clearly defined or quantitative.

Initial assumption would be that such system would decrease the human dependency and reduce the number of jobs consequently. However, given that the same approach is applicable to almost every new technology/product emerges in the market, it is shown that they also create new opportunities.

1.2.3. Environmental Impact of the Solution

Since the project will be software based, no significant effect on environment will be observed. Of course, one can discuss that keeping the servers on, utilizing their resources to the maximum level continually will increase the total electricity consumption.

1.2.4. Societal Impacts of the Solution

This project has many social impacts, though not directly. With the solution we developed, strengthening the question-answering model carried out in connection with this project, we will aid to the process of knowledge acquisitions for the learners.

1.2.5.   Legal Issues Related to the Project

Such as any system that is designed to present information to the user, Question Answering systems, which our project Question Generation system will yield to development of, requires the necessary information for answering the question. Under this circumstances, source of this information is critically important. Consent of the creator must be obtained before developing such system.

Also, the system must not give inaccurate information to the user, since they may depend on the system on a critical task which fault can't be tolerated at all.

2. Project Definition and Planning

Turkish Question Generation Model is a neural network-based model that is capable of generating questions from a given piece of text in Turkish Language. This project implements such model using *Recurrent Neural Network (RNN)* and/or *Convolutional Neural Network (CNN)*.

2.1. Project Definition

Main feature of the project will be accepting an input, create possible questions from that input while assigning probability scores to each of the generated questions. Questions are delivered in easily parsable file format such as JSON, XML, HTML, CSV, TSV etc.

There is not a GUI is planned for the QG project, however it is going to be presented as CLI and *Python Module*[2] servers. for public use if allowed.
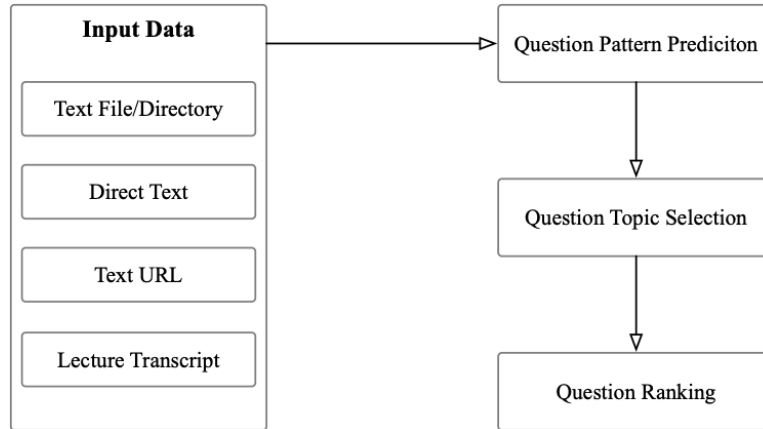

Figure 1 – Question Generation system schema

2.1.1. Dataset

There are lots of publicly available datasets for natural language processing tasks. However, for this project our main constraint was to build our

---

[2]An importable Python file that contains the functions for incorporating Turkish Question Generation for further research.

system on top of a Turkish Dataset. Some other constraints we had was that the data must consist of correctly structured sentences in grammatical aspect, it must be labeled, come from a reliable source with the consent of the owner(s) and must be very large, given that larger dataset yields better models.

2.1.2. Programming Languages and Frameworks

This project is developed with *Python 3*. It has chosen because it is relatively simple and allows us to focus on the project rather than computational concerns, it has a very large community for troubleshooting, which also decreases the chances of having issue about the language.

Furthermore, Python is adapted by the AI researchers and there are numerous resources, frameworks and libraries that is developed for Python in which we utilize heavily. Framework we are going to be using is PyTorch. This framework recently developed and provides state-of-art parallel processing of tenors on GPU. Also, there are other libraries that are developed for scientific computing and has been used in this project such as pandas, NumPy etc.

2.2. Project Planning

Table 2 – Project plan for 14 weeks

Task	Responsible Person	Weeks													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project Proposal	Erdal Dogan Alp Gokcek	X	X	X											
Literature Review	Erdal Dogan Alp Gokcek			X	X	X	X		X						
Development Environment Preparation	Erdal Dogan Alp Gokcek								X	X	X	X			
System Prototype	Erdal Dogan Alp Gokcek											X	X	X	
Report and Project Presentation	Erdal Dogan Alp Gokcek							X	X						X

2.2.1. Aim of the Project

Turkish Question Generation Model aims to create a Deep Learning model that given a paragraph, passage or an entity in Turkish, it presents the possible questions that can be answered solely by the given content to the system.

$S \rightarrow$ Richard Phillips Feynman (11 Mayıs 1918 – 15 Şubat 1988) 20. yüzyılın en önemli fizikçilerindendir. Kuantum elektrodinamiği üzerindeki çalışmaları nedeniyle 1965'te Julian Schwinger ve Sin-Itiro Tomonaga ile beraber Nobel Fizik Ödülüne layık görülmüştür.

Turkish Question Generation

output

$Q_i = \{$
"Richard Feynman ne zaman doğdu?",
"Feynman ne zaman nobel aldı?",
"Rıchard Feynman çalışma konusu",
"Richard Feynman tam adı nedir?"
$\vdots$
$\}$

Figure 2 – Question Generation Model diagram

2.2.2. Project Coverage

The project consists of sequential stages where the given input is passed over from one to other. These stages can be classified as the main content of the project. Initially, there will be a pre-processing stage for the data, this pre-process might be tweaking the string or simplifying the sentence. Then, BERT language model will be fine-tuned upon retrieving the input string.

Project is based on software; outcome and the deliverables will consist of soft materials. Outcome of the project will be an executable file and/or importable Python module that can be used by other researchers.

2.2.3. Use Cases

Provide UML use case diagrams by specifying systems, actors, use cases and relationships.

2.2.4. Success Criteria

Outcomes of the project will be evaluated by couple of metrics. Given that the outputs will be natural language instances, existing methods and algorithms specifically designed for evaluation of generated natural language examples will be utilized. There are multiple algorithms and methods available, which consider different aspects of the outcome. Some of them are:

- BLEU

- ROUGE

- METEOR

Each of the enumerated methods are taking different aspects into account, while *BLEU* score is calculated with best match length is prominent, *ROGUE* is based on recall score. They all have their tradeoffs and neither of them is merely strong indicator of success.

2.2.5. Project Time and Resource Estimation

The project is estimated to take 16 weeks for 2 undergraduate students that are novices to topic. While estimating, course load and other external factors are taken into account. Detailed timeline can be seen from Table 1 at the previous page.

Assuming that each member spends 8 hours a week working on issues that are related to this project, it makes approximately 128 hours per person, 256 hours spent in total. Also, it has been mentioned that computing resources will be required for the project. If we use cloud computing provider's (AWS for this particular example) rates for this estimation, we observe that hourly price is \$2 for a server that we need to run. Considering the fact that these are very intensive applications, it might be the cases that server will be up and making computations for long hours, 20 hours a week. This estimation sums up to \$600 in server running costs and 256 men hours. Notice that not any access to books, online courses or any other learning material has not been included.

2.2.6. Solution Strategies and Applicable Methods

For the project we decided to use off-the-shelf language models by fine-tuning them according to our data. Of course, we could've tried to achieve it without using such advanced models, however, in this case there would be risk of not being able to complete the project in time or a flawed model could have been produced.

We had an option to use another main-stream language model such as ELECTRA or RASA. Given that BERT was adopted by the community, faster to train and documentation is provided is more comprehensive than other, we decided to use BERT model.

2.2.7. Risk Analysis

One of the major risks that may occur is that the model is giving questions that are not directly related to given inputs or contain grammatical/structural errors. There are not any risks.

2.2.8. Tools Needed

The project requires the development of Machine and Deep Learning Models from scratch. Due to wide selection of libraries for this purpose, community behind it and being the de-facto standard of the AI research and scientific development, it has been decided that *Python Programming Language* will be used for development and implementation during this project.

Specifically, for the model development, an up-to-date, state-of-art library named *PyTorch* will be used extensively. Also, development of such models and processing large amounts of data requires computing resources. Therefore, servers that are configured with GPU optimization in mind will be required for faster development process.

3. Theoretical Background

3.1. Literature Survey

In this section, a survey of the literature from multiple sources could be found.

3.1.1. Question Answering

Question Answering (QA) is problem of automatically presenting the answers of the question that is asked by users without any human interaction. In a QA model, it is expected that the system has access to the necessary information to answer the question.

In order to develop such model, we need excessively large data that consists of Questions and their Answers, a duple that we will denote by $<Q, A>$. Unfortunately, labeling this kind of data manually or creating answers from scratch for an Artificial Intelligence based solution is not the most efficient way. They address these problems by;

1. Large scale high-quality dataset from Community-QA websites such as Yahoo, Quora etc. is obtained, since they provide large scale QA pairs generated by real users.

2. Two ways of accomplishing such task is implemented and compared. One is a retrieval based method using Convolutional Neural Networks(CNN) and other is generation-based method using Recurrent Neural Network (RNN).

3. Outcomes of the QG model is integrated with end-to-end QA task. It is evaluated on three state-of-art datasets, SQuAD, MS MARCO, and WikiQA. Results show that generated questions can improve QA quality on all these three datasets.

3.1.2. Question Generation

In their paper, research engineers from Microsoft Research proposed a way of extracting question from a given piece of text. Duan et al. (2017) While they used NLP methodologies and Neural Networks for their paper, they didn't include the semantics of words in their paper.

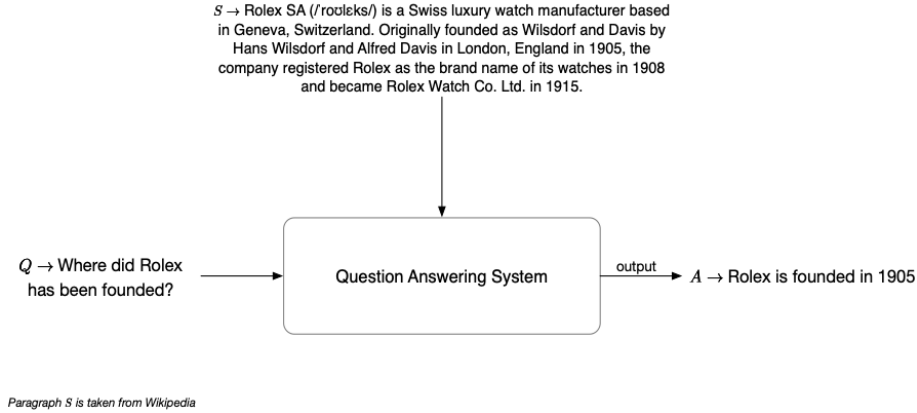QG Engine is consist of four components:

$S \rightarrow$ Rolex SA (/ˈroʊlɛks/) is a Swiss luxury watch manufacturer based in Geneva, Switzerland. Originally founded as Wilsdorf and Davis by Hans Wilsdorf and Alfred Davis in London, England in 1905, the company registered Rolex as the brand name of its watches in 1908 and became Rolex Watch Co. Ltd. in 1915.

$Q \rightarrow$ Where did Rolex has been founded?

Question Answering System

output

$A \rightarrow$ Rolex is founded in 1905

Paragraph S is taken from Wikipedia

Figure 3 – Sample Question Answering model

1. Question Pattern Mining

2. Question Pattern Prediction

3. Question Topic Selection

4. Question Ranking

3.1.3. BERT Language Model

We have encountered a language model called BERT which is also known as Bidirectional Encoder Representation from Transformers developed by Google (Attention is all you need). It uses a machine learning model called Transformers. We have researched other language models and for the language translation problem, we have found two techniques which are LSTM and Transformers.

First of all, we have found out that LSTMs are slow to train, words are passed sequentially and words are getting generated sequentially which can take significant time for the neural network to learn the language. And LSTMs are not truly bidirectional, they learn left to right and right to left separately and simply concatenate afterwards. This is where Transformers come. Transformers are faster because they can process words simultaneously and deeply bidirectional since they can learn in both directions.

On the Figure 4, you can see the architecture of a transformer. It is formed by two components which are encoder and decoder.
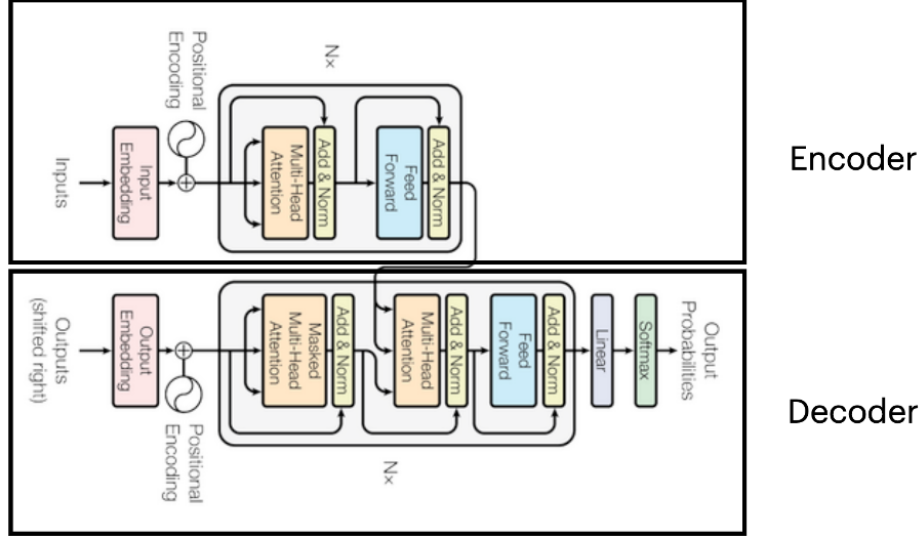
Figure 4 – Architecture of a *Transformer*

As the BERT's name suggests, if we put encoders on after another, we will obtain the BERT language model. The architecture of BERT language model could be found on figure 5. It is the current state-of-the-art language model for NLP tasks. Chan and Fan (2019)
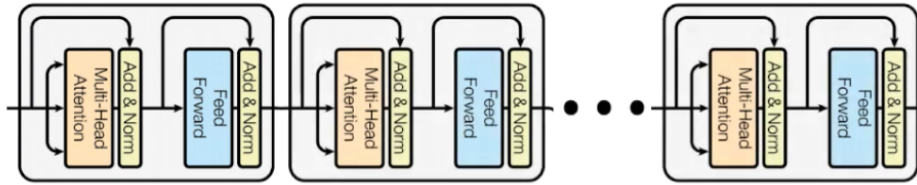
Figure 5 – BERT language model architecture

We have found a paper that uses this language model to solve our problem and they have made sequentially improved 3 models which are BERT-QG (BERT Question Generation), BERT-SQG (BERT Sequential Question Generation) and BERT-HLSQG (BERT Highlight Sequential Question Generation) models. QG is the simplest model that this paper introduces. It

is the initial attempt to create a powerful Question Generation Model using BERT. As proposed by researchers, considering the previous decoded results significantly improve the quality of the model. However, in BERT-QG, token generation is performed without considering the previous states or decoded results. Due to this consideration, BERT-SQG is developed. BERT-SQG addressed the problem of ignoring the previous decodes in the BERT-QG model. However, researchers researched and concluded that BERT-SQG is not capable of producing quality questions in lengthy situations and if an answer phase appears multiple times it struggles to decide which one to decide. As a result, BERT-SQG is restructured and BERT-HLSQG, a model outperforming BERG-SQG is obtained.

They have compared these models with the known to be best question generation models which were NQG-RC and PLQG. NQG-RC is a seq2seq question model based on bidirectional LSTMs. PLQG is a seq2seq network which is capable of handling long text input. This model is known to be the state-of-the-art models for QG tasks. As shown in table 3, their models have outperformed the state-of-the-art models on every metric.

Table 3 – Performance comparison of Question Generation models on different datasets

	Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
SQuAD 73K	NQG-RC	43.09	25.96	17.50	12.28	16.62	39.75
	PLQG	43.47	28.23	20.40	15.32	19.29	43.91
	BERT-QG	34.17	15.52	8.36	4.47	14.78	37.60
	BERT-SQG	48.38	**33.15**	24.75	19.08	22.43	46.94
	BERT-HLSQG	**48.29**	33.12	**24.78**	**19.14**	**22.89**	**47.07**
SQuAD 81K	PLQG	44.51	29.07	21.06	15.82	19.67	44.24
	BERT-QG	34.18	15.51	8.57	4.97	14.57	37.65
	BERT-SQG	50.18	35.03	26.60	20.88	23.84	48.37
	BERT-HLSQG	**50.71**	**35.44**	**26.95**	**21.20**	**24.02**	**48.68**

3.2. Question Generation Model With BERT Language Model

As a result of this literature survey, we have decided to go with the BERT language model. BERT language model shows significant performance over various NLP tasks, such as classification, summarization, translation etc. It has a powerful architecture behind it which is transformers. We have searched a pre-trained Turkish model of BERT and we found a repository on

GitHub for it. Starting from next week, we will start using this model as well.

As the dataset, we have decided to go with Turkish Wikipedia dataset. There are several reasons behind it. Since the data of Wikipedia is close to textbook data, it should be beneficial to train our model with this dataset. And our advisor, Seniz Demir, had a Wikipedia Parser project. Thus, we will use Turkish Wikipedia dataset.

4. ANALYSIS AND MODELING

Aim of this project is to develop a Turkish Question Generation Model. Given a paragraph or a piece of text, our model creates the possible questions that can be answered with the information given in that paragraph.

However, development of such model requires an extensively large and labeled dataset that consist of paragraphs, corresponding question and answers to those questions. Unfortunately, such dataset is not available in Turkish Language. There are examples of it, but they neither contain large amounts of information nor we find the data reliable.

For the reasons described above, we decided to compose our own dataset. Our dataset consist of paragraphs from Turkish Wikipedia. By using the Person entities from the Turkish Wikipedia, we can create the question patterns for each attributes of a person. Later, these questions patterns are completed by substituting the person's name with the placeholders. At this step, we have our questions and the description paragraph of the person available to us.

However, we must also confirm that the answer to the question is also found in the paragraph. For this reason, we use a fuzzy search algorithm in order to determine if the answer is in the paragraph or not. If it is, we add the questions to our dataset, if not we skip to next question.

Apart from the dataset, main goal of the project is to create the Question Generation (QG) model. Given that this model QG model will be developed based on the dataset, first stage of this project has been dedicated to production of the dataset.

4.1. System Factors

Since our work is based on the data acquired from Wikipedia and its manipulation, source data is highly effective on the outcome. While creating the dataset we always aim to create grammatically flawless questions while covering the variety of inputs that may come from the end-user

4.2. How System Works

Input data consists of 53,000 person entities from Turkish Wikipedia. Occupations of each person and their attributes are also presented. For instance, for a 'Soldier', given attributes are: 'Rank', 'Wars', 'Nationality', 'Commanders' etc. Given that most of these attributes are also mentioned in the description paragraph of the corresponding person, when we create the questions that asks about these attributes, we will have the description paragraph and the questions together.

Initially, each question patterns for attributes are created manually containing a placeholder for the person's name. Later, by iterating over each person and substituting names for the their attributes we create an example of question.

When we iterate over every person, we accumulate a dataset which consist of paragraphs that describe the person, and questions about the attributes of the person which also is contained in the paragraph as well.

4.2.1. Modelling

Explanation of system model.

4.2.2. System Architecture

Explanation of system architecture.

4.2.3. UML (Unified Modeling Language) Diagrams

5. Design, implementation and testing

During the design phase of our project, we had two different components that have to be completed. These are:

- A Turkish question answering dataset that will be used to train the QG model.

- A QG model.

In this section, we will explain that what kind of design process we have followed during the design, implementation and testing of these two components separately.

5.1. Design

5.1.1. Dataset

Before creating our dataset, we have researched if any of the datasets are suitable and sufficient. We have found one dataset that is suitable for our model. That dataset has around 2.2k questions, and the descriptive texts were about Turkish & Islamic Science History, which is not completely suitable and sufficient. Hence, we have decided to create our dataset. During the design phase of the dataset, we have used 2 Wikipedia datasets. The first data was semi-cleaned scrapped Turkish Wikipedia data. Data we had has the short descriptions and the table contents. An example of the table content could be found in Figure 6.

Figure 6 – Sample Wikipedia table content

However, this data was not sufficient for our dataset since most of the table contents' data does not appear on the short description. Thus, we had

to use long descriptions for each of them. After that, we also matched the whole Wikipedia Turkish dump we found from Kaggle with our first data. This data had long descriptions of every document that is currently available on Wikipedia. During the matching phase, we have decided to match each document by their id's.

After finishing the data part, we have started to analyze our data. We have grouped each person according to their occupation. After grouping, we have found out that we have twenty different occupations. Each occupation has its own table attributes. Our main goal is to generate question patterns according to their occupations and the most used 10 attributes for that specific occupation. Our analysis on the Wikipedia persons data could be found on Appendix A.

Finally, we have started to generate the question patterns according to the occupation and the attributes. During this process, we have decided to the question pattern styling.

5.1.2. Question Generation Model

In the design process of Question Generation Model, we have followed the architecture of the Chan and Fan (2019). According to their architecture, we will use an sequential-to-sequential model. For the given paragraph $C = [c_1, c_2, ..., c_n]$, the answer phase $A = [a_1, a_2, ..., a_n]$ and the question $\hat{Q} = [\hat{q}_1, ..., \hat{q}_i]$, the input sequence X_i will be shown as follows.

$$X_i = ([CLS], C, [SEP], A, [SEP], \hat{q}_1, ..., \hat{q}_i, [MASK]) \tag{1}$$

There are 3 different tokens in that sequence. These can be summarized as follows:

- $[CLS]$: Token shows the beginning of a sequence.

- $[SEP]$: Token separates the answer from the context and the generated question.

- $[MASK]$: Token shows the ending of a sequence.

After that, we take the final hidden state for the last token $[MASK]$ in the input sequence. We denote the final hidden vector of $[MASK]$ as

$\mathbf{h}[MASK] \in R^h$. After that BERT model will be adapted by adding an affine layer $\mathbf{W}_{SQG} \in R^{h|V|}$ to the output of the $[MASK]$ token. The label probabilities $Pr(w|X_i) \in R^{|V|}$ are computed by a soft-max function as follows.

$$Pr(w|X_i) = softmax(\mathbf{h}[MASK] * \mathbf{W}_{SQG} + \mathbf{b}_{SQG}) \qquad (2)$$

$$\hat{q}_i = argmax_w Pr(w|X_i) \qquad (3)$$

Later on, on each iteration, the newly generated token $\hat{q}_i$ will be appended into X and the generation process will be continued until $[SEP]$ is predicted. Thus, in the end, we will have our question.

5.2. Implementation

Similar to the design part, we have developed each component separately.

5.2.1. Dataset

created We have generated around 1700 question patterns by hand.

5.2.2. Question Generation Model

5.3. Testing

Explanation of the testing of your system.

6. Results

7. Conclusion

Briefly explain the problem you studied, the solution method you proposed and your experience in the implementation. Then, please proceed to subsections and explain the requirements in details.

7.1. Life-Long Learning

Identify additional knowledge, skills, and attitudes that you needed to complete the project. In the context of the project, which tools and methods did you learn by yourself? From which sources did you learn them? How difficult was for you to learn these topics by yourself? How did you manage to collect the relevant information? Answer all the questions.

7.2. Professional and Ethical Responsibilities of Engineers

Define professional and ethical responsibilities you followed during your design process. What are the professional and ethical standards you used in the project? Is there an ethics code or code of conduct? Answer all the questions.

7.3. Contemporary Issues

Evaluate you project experience in terms of contemporary issues related to the problem that you studied. What kind of contemporary tools you have used throughout your project? Analyze the contemporary issues related to the future of the field that you work in this project. What will change in the products, services, or processes in the next ten, twenty, or fifty years, with emerging technologies such as 3D printers, big data analytics, nanotechnology, internet of things, quantum computing, biotechnology, artificial intelligence, cognitive science, and robotics?

7.4. Team Work

During your design project experience, describe the team work that you have participated. Give positions and majors of the people in the company that you worked with. Evaluate the composition, organization, and performance of your team. Describe how you shared the load in terms of the project tasks.

──────────── attributes.txt ────────────

```
Total persons:   53115

Top Occupations and their attributes:
{'Asker': ['rütbesi', 1135,
            'doğumyeri', 1097,
            'doğumtarihi', 1092,
            'hizmetyılları', 1038,
            'bağlılığı', 940,
            'ölümyeri', 919,
            'savaşları', 915,
            'ölümtarihi', 887,
            'komutaettikleri', 689,
            'madalya', 578],
 'Basketbolcu': ['pozisyon', 1375,
                'doğumtarihi', 1360,
                'doğumyeri', 1356,
                'takım1', 1303,
                'takımyıl1', 1213,
                'takım2', 1154,
                'takımyıl2', 1103,
                'takım', 1098,
                'lig', 1031,
                'takım3', 1013],
 'Bilim adamı': ['doğumyeri', 1411,
                'dalı', 1388,
                'doğumtarihi', 1258,
                'milliyeti', 1011,
                'çalıştığıyerler', 926,
                'ölümyeri', 849,
                'ölümtarihi', 838,
                'ödüller', 775,
                'önemlibaşarıları', 665,
                'vatandaşlığı', 577],
 'Buz patencisi': ['ülke', 217,
                    'doğumtarihi', 214,
                    'koç', 207,
                    'combinedtotal', 192,
                    'combineddate', 191,
                    'fsscore', 185,
                    'fsdate', 185,
```

```
                         'koreograf', 145,
                         'spscore', 131,
                         'spdate', 130],
'Filozof': ['doğumtarihi', 281,
            'ölümtarihi', 241,
            'doğumyeri', 225,
            'çağ', 221,
            'bölge', 212,
            'ölümyeri', 192,
            'etkilendikleri', 192,
            'etkiledikleri', 178,
            'ilgialanları', 167,
            'okulgelenek', 150],
'Futbolcu': ['doğumyeri', 17819,
             'pozisyon', 17649,
             'doğumtarihi', 17559,
             'tamadı', 15754,
             'kulüp1', 14197,
             'kulüpyıl1', 14039,
             'kulüp2', 12783,
             'kulüpyıl2', 12725,
             'kulüp3', 11288,
             'kulüpyıl3', 11250],
'Güreşçi': ['doğumyeri', 152,
            'doğumtarihi', 150,
            'debut', 139,
            'başlık', 135,
            'ringadları', 131,
            'eğiten', 114,
            'doğumadı', 103,
            'eğitildiğiyer', 86,
            'yaşadığıyer', 44,
            'emekliliği', 34],
'Hakem': ['turnuva', 127,
          'görevi', 126,
          'doğumtarihi', 124,
          'yıl', 123,
          'doğumyeri', 114,
          'meslek', 73,
          'tamismi', 43,
          'ölümtarihi', 17,
          'ölümyeri', 7,
          'görevi2', 2],
'Kişi': ['doğumtarihi', 8861,
         'meslek', 8267,
```

```
              'doğumyeri', 7433,
              'ölümtarihi', 4109,
              'ölümyeri', 3675,
              'doğumadı', 2953,
              'aktifyılları', 2139,
              'başlık', 2051,
              'yer', 1775,
              'etkinyılları', 1711],
'Kraliyet': ['hükümsüresi', 1545,
             'sonragelen', 1425,
             'veraset', 1417,
             'ölümtarihi', 1412,
             'öncegelen', 1404,
             'babası', 1379,
             'hanedan', 1310,
             'doğumtarihi', 1285,
             'ölümyeri', 1075,
             'annesi', 1005],
'Makam sahibi': ['makam', 3694,
                 'doğumyeri', 3685,
                 'doğumtarihi', 3508,
                 'dönembaşı', 3486,
                 'öncegelen', 3160,
                 'dönemsonu', 2974,
                 'sonragelen', 2817,
                 'partisi', 2451,
                 'ölümtarihi', 1824,
                 'ölümyeri', 1802],
'Manken': ['yer', 734,
           'doğumtarihi', 733,
           'saçrengi', 715,
           'gözrengi', 713,
           'ulus', 645,
           'doğumadı', 234,
           'bgcolour', 209,
           'imagesize', 186,
           'birthname', 89,
           'boy', 69],
'Müzik sanatçısı': ['artalan', 5441,
                    'tarz', 5402,
                    'etkinyıllar', 4894,
                    'meslek', 3814,
                    'plakşirketi', 3442,
                    'doğumadı', 2874,
                    'köken', 2777,
```

```
                         'yer', 2134,
                         'doğum', 2069,
                         'çalgı', 1983],
'Oyuncu': ['yer', 2785,
           'doğumtarihi', 2584,
           'meslek', 2164,
           'etkinyılları', 2052,
           'doğumadı', 1389,
           'altyazı', 1133,
           'evlilik', 960,
           'ulus', 644,
           'ölümtarihi', 511,
           'ölümyeri', 494],
'Profesyonel güreşçi': ['doğumyeri', 96,
                        'başlık', 93,
                        'debut', 90,
                        'doğumadı', 90,
                        'doğumtarihi', 89,
                        'ringadları', 80,
                        'eğiten', 73,
                        'eş', 55,
                        'eğitildiğiyer', 53,
                        'çocuklar', 35],
'Sanatçı': ['doğumtarihi', 702,
            'alanı', 695,
            'ölümtarihi', 508,
            'ölümyeri', 475,
            'doğumyeri', 424,
            'milliyeti', 330,
            'doğumadı', 314,
            'yer', 294,
            'resimaltı', 278,
            'ünlüyapıtları', 212],
'Sporcu': ['doğumtarihi', 936,
           'doğumyeri', 814,
           'ülke', 800,
           'spor', 795,
           'yarışma', 513,
           'uyruk', 196,
           'kei', 179,
           'resimaltı', 178,
           'ağırlık', 176,
           'ölümtarihi', 170],
'Tenis sporcu': ['doğumyeri', 239,
                 'vatandaşlık', 235,
```

```
                    'enyükseksıralama', 229,
                    'oyunstili', 228,
                    'wimbledonsonuçları', 228,
                    'amerikaaçıksonuçları', 225,
                    'fransaaçıksonuçları', 224,
                    'avustralyaaçıksonuçları', 223,
                    'toplamkupa', 222,
                    'yaşadığıyer', 216],
 'Voleybolcu': ['doğumtarihi', 112,
                'doğumyeri', 110,
                'pozisyon', 107,
                'milliyeti', 104,
                'kulüpyıl', 104,
                'kulüptakım', 103,
                'bulunduğukulüp', 101,
                'numarası', 99,
                'millitakım', 98,
                'milliyıl', 75],
 'Yazar': ['doğumyeri', 1911,
           'meslek', 1897,
           'doğumtarihi', 1776,
           'milliyet', 1311,
           'ölümtarihi', 1079,
           'ölümyeri', 1043,
           'başlık', 751,
           'tür', 620,
           'dönem', 578,
           'ilkeser', 513]}
```

Appendix B

Your Appendix goes here

Acknowledgments

Your acknowledgments goes here

REFERENCES

N. Duan, D. Tang, P. Chen, and M. Zhou, "Question generation for question answering," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 866–874. [Online]. Available: https://www.aclweb.org/anthology/D17-1090

Y.-H. Chan and Y.-C. Fan, "A recurrent BERT-based model for question generation," in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering.* Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 154–162. [Online]. Available: https://www.aclweb.org/anthology/D19-5821