



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

Our goal in this project is to develop a Turkish Question Generation System using available online sources such as Wikipedia. Question Generation Systems are capable of generate various logical questions from the given text input. QG Systems are prevalent in several computer applications such as chatbots, automated grading systems etc. Development of such a system is particularly important since there are not any examples of online tools and available datasets in Turkish. However, QG research has already reached a notable level in other languages such as English and there are available datasets for this task. For instance, SQuAD by Stanford University is one of the well- known datasets used for this task in English.

In this project, we are aiming to create the first Turkish Question Generation Dataset of moderate size. This step is required for developing learning-based solutions that we're going to apply in this project. We plan to apply different machine learning based techniques for questions of generating varying forms (Who-What-When etc.) and assess their performances on the dataset we'll gather.

Keywords: Automatic speech recognition, Natural Language Processing, Deep Learning, Sentiment Analysis, MOOC¹

¹Massive 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 projedeki amacımız, Wikipedia gibi mevcut çevrimiçi kaynakları kullanarak bir Türkçe Soru Oluşturma Sistemi geliştirmektir. Soru Üretme Sistemleri, verilen metin girişinden çeşitli mantıksal sorular üretebilir. QG Sistemleri, sohbet robotları, otomatik derecelendirme sistemleri vb. Gibi birçok bilgisayar uygulamasında yaygındır. Bu tür bir sistemin geliştirilmesi, Türkçe’de herhangi bir çevrimiçi araç ve veri seti örneği bulunmadığından özellikle önemlidir. Bununla 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

| | |
|---|----|
| ABSTRACT | v |
| ÖZET | vi |
| 1. Introduction | 1 |
| 1.1. Motivation | 2 |
| 1.2. Broad impact | 3 |
| 1.2.1. Global Impact of the solution | 3 |
| 1.2.2. Economic Impact of the solution | 3 |
| 1.2.3. Environmental Impact of the solution | 3 |
| 1.2.4. Societal Impacts of the solution | 3 |
| 1.2.5. Legal Issues related to the project | 3 |
| 2. Project Definition and Planning | 5 |
| 2.1. Project Definition | 5 |
| 2.1.1. Dataset | 5 |
| 2.1.2. Programming Languages and Frameworks | 6 |
| 2.2. Project Planning | 6 |
| 2.2.1. Aim of the project | 7 |
| 2.2.2. Project Coverage | 7 |
| 2.2.3. Use Cases | 7 |
| 2.2.4. Success Criteria | 8 |
| 2.2.5. Project time and resource estimation | 8 |
| 2.2.6. Solution Strategies and Applicable Methods | 9 |
| 2.2.7. Risk Analysis | 9 |
| 2.2.8. Tools Needed | 9 |
| 3. Theoretical Background | 10 |
| 3.1. Literature Survey | 10 |
| 3.1.1. Question Answering | 10 |
| 3.1.2. Question Generation | 11 |
| 3.1.3. BERT Language Model | 11 |
| 3.2. Question Generation Model with BERT Language Model | 14 |
| 4. Analysis and Modeling | 15 |
| 4.1. System Factors | 15 |
| 4.2. How System Works | 15 |
| 4.2.1. Modelling | 15 |
| 4.2.2. System Architecture | 15 |

| | | |
|-----------------|--|----|
| 4.2.3. | UML (Unified Modeling Language) Diagrams | 15 |
| 5. | Design, implementation and testing | 16 |
| 5.1. | Design | 16 |
| 5.2. | Implementation | 16 |
| 5.3. | Testing | 16 |
| 6. | Results | 17 |
| 7. | Conclusion | 18 |
| 7.1. | Life-Long Learning | 18 |
| 7.2. | Professional and ethical responsibilities of engineers . . | 18 |
| 7.3. | Contemporary Issues | 18 |
| 7.4. | Team Work | 18 |
| APPENDIX A | | 19 |
| APPENDIX B | | 20 |
| ACKNOWLEDGMENTS | | 21 |

LIST OF FIGURES

| | | |
|---|--|----|
| 1 | Question Generation system schema | 5 |
| 2 | Question Generation Model diagram | 7 |
| 3 | Sample Question Answering model | 11 |
| 4 | Architecture of a <i>Transformer</i> | 12 |
| 5 | BERT language model architecture | 13 |

LIST OF TABLES

| | | |
|---|---|----|
| 1 | Project plan for 14 weeks | 6 |
| 2 | Performance comparison of Question Generation models on different datasets | 13 |

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. 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

What are the societal impacts of the project?

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.

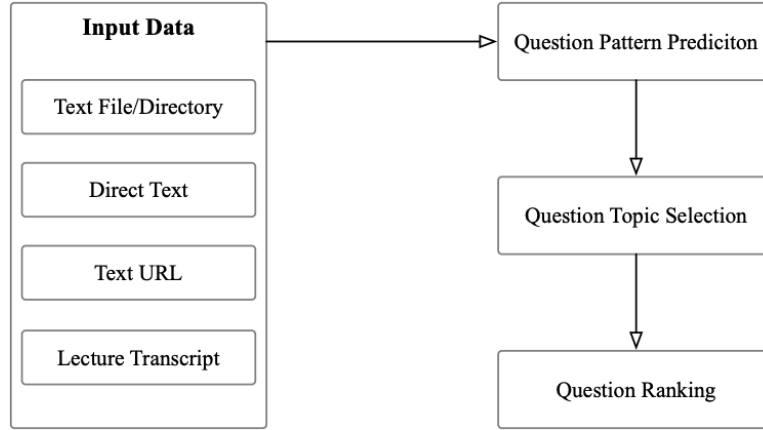


Figure 1 – Question Generation system schema

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

²An 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 tensors 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 1 – 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 | | | | | | | | | | | | | | |
| Literature Review | Erdal Dogan Alp Gokcek | | | | | | | | | | | | | | |
| Development Environment Preparation | Erdal Dogan Alp Gokcek | | | | | | | | | | | | | | |
| System Prototype | Erdal Dogan Alp Gokcek | | | | | | | | | | | | | | |
| Report and Project Presentation | Erdal Dogan Alp Gokcek | | | | | | | | | | | | | | |

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.

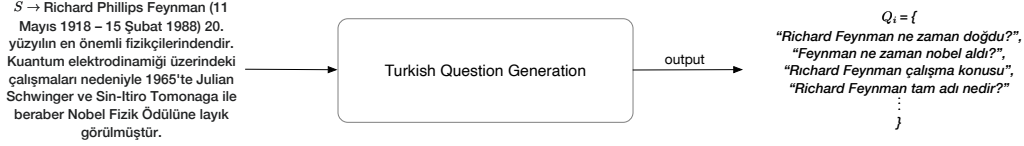


Figure 2 – Question Generation Model diagram

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, *ROUGE* 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

You will define your problem in this section and add subheadings as needed. If you give your problem description and solution methodology in this section, you may need to change the title of this section. We usually write a couple of sentence in this part that explains what we will discuss in this section. For example: We will first provide the problem definition of our study (write you topic here), and then give the mathematical model that we propose as a solution method in this section.

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 $\langle Q, A \rangle$. 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.

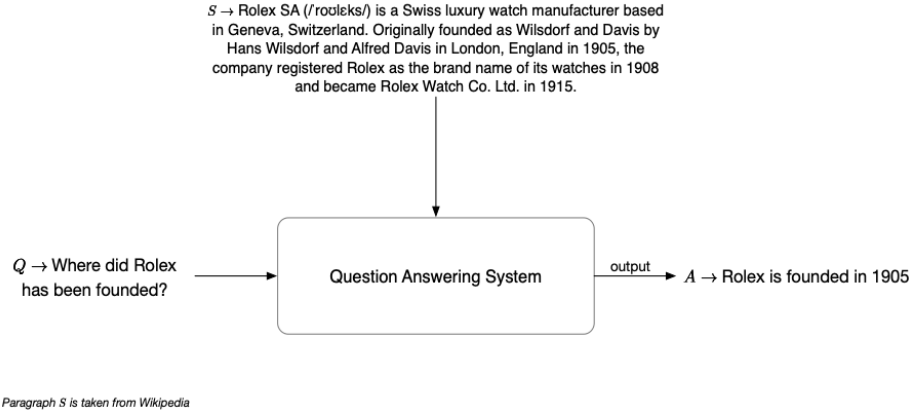


Figure 3 – Sample Question Answering model

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. (1) 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:

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.

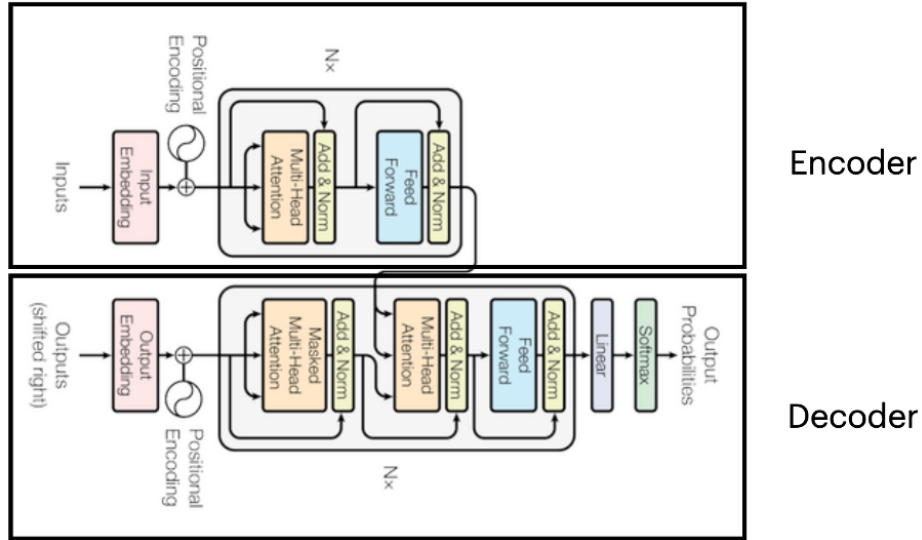


Figure 4 – Architecture of a *Transformer*

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. (2)

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

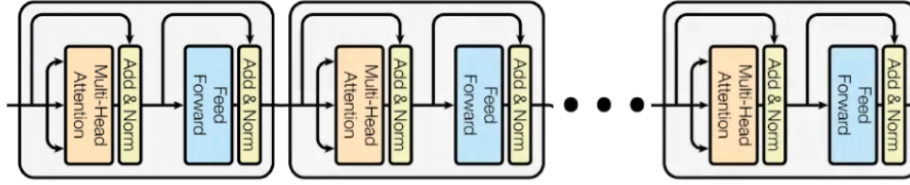


Figure 5 – BERT language model architecture

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 2, their models have outperformed the state-of-the-art models on every metric.

Table 2 – 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

We first give details about the project that we selected. We define the scope of the project, functional and non-functional requirements. Then, we give details of the problem that the company was experiencing. Finally, we provide detailed analysis that we conducted by using real-life data obtained from the company.

4.1. System Factors

Explain factors that can affect the system.

4.2. How System Works

Explain how your system will work

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

Explain design, implementation and testing of your system in detail.

5.1. Design

Explanation of the design of your system.

5.2. Implementation

Explanation of the implementation of your system.

5.3. Testing

Explanation of the testing of your system.

6. RESULTS

Explain your 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 your 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.

APPENDIX A

Your Appendix goes here

APPENDIX B

Your Appendix goes here

ACKNOWLEDGMENTS

Your acknowledgments goes here

REFERENCES

- [1] 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>
- [2] 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>