REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

DOMAIN-SPECIFIC QUESTION ANSWERING ASSISTANT

13011029 — Umut GÜNERİ

SENIOR PROJECT

Advisor
Prof. Dr. Banu Diri

June, 2018

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Dr. Banu Diri, for the patient guidance, encouragement and advice she has provided throughout my time as her student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

Umut GÜNERİ

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

biLSTM	Bidirectional Long Short Term Memory
CUDA	Compute Unified Device Architecture
DL	Deep Learning
DSTC	Dialogue System Technology Challenge
GPU	Graphics Processing Unit
IEEE	Institute of Electrical and Electronics Engineers
IWSDS	International Workshop on Spoken Dialogue Systems Technology
LSTM	Long Short Term Memory
NIPS	Neural Information Processing Systems
NLTK	Natural Language Toolkit
RNN	Recurrent Neural Networks
SQuAD	Stanford Question Answering Dataset
SIGDIAL	Special Interest Group of the Association for Computational Linguistics
SGD	Stochastic Gradient Descent
TREC-QA	Text Retrieval Conference Question Answering

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Domain-Specific Question Answering Assistant

Umut GÜNERİ

Department of Computer Engineering
Senior Project

Advisor: Prof. Dr. Banu Diri

The goal of this project is to implement a text based question answering system which can be used for different sectors. The system was designed that finds the closest questions among the previously asked questions by user and also the system returns correct answer to the user. In case of the answer could not be found, this system informs the user and finds soon a suitable answer. If the chosen answer is satisfactory for the user, the new question-answer binary is added to the database. After a certain period of time, the data is retrained to increase the success of system.

To find correct answer the closest question among the previously asked questions need to be found. This problem was solved with two different approaches. First approach is using neural network and second approach is using string similarity methods(Cosine similarity, Levenhstein distance, Qgram similarity) as a solution of the problem.

Neural networks models depend on size of data significantly, therefore string similarity methods can be used in some circumstances which insufficient sample exist in dataset. On the other hand, in case of the size of data increases, the accuracy of the neural network model will also increase.

Keywords: chatbot, assistant, question answering, neural networks, string similarity, Cosine similarity, Levenhstein similarity, Qgram similarity

ÖZET

Konuya Özel Soru Cevap Asistanı

Umut GÜNERİ

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Prof. Dr. Banu Diri

Proje farklı sektörler için kullanılabilecek genişletilebilir text tabanlı yanıt verme sistemidir. Bu projede sistem, kullanıcı tarafından yöneltilen soruya daha önceden sorulmuş sorular arasından en yakın soruyu bulup, bu sorunun cevabını kullanıcıya cevap olarak döndürür. Cevabın bulunamaması durumunda, bu sistem kullanıcıyı bilgilendirir ve yakın zamanda uygun bir cevap bulur. Seçilen cevap kullanıcı için tatmin edici ise, yeni soru-cevap ikilisi veritabanına eklenir. Belirli bir süre sonra, sistemin başarısını artırmak için veriler yeniden eğitilir.

Doğru cevabı bulmak için, daha önce sorulan sorular arasından en benzer soru bulunmalıdır. Bu problem iki farklı yaklaşımla çözülmektedir. İlk yaklaşım olarak, yapay sinir ağları, ikinci yaklaşım olarak ise, string benzerlik yöntemleri(Kosinüs benzerliği, Levenhstein uzaklığı, Qgram benzerliği) problemin çözümü olarak kullanılmaktadır.

Sinir ağları modelleri büyük ölçüde veri büyüklüğüne bağlıdır, bu nedenle veri setinde yetersiz örneklem var olduğu durumlarda string benzerlik yöntemleri kullanılabilir. Öte yandan, veri büyüklüğü arttıkça, sinir ağı modelinin doğruluğunun da artması beklenmektedir.

Anahtar Kelimeler: chatbot, soru cevap sistemi, asistan, yapay sinir ağları, string benzerliği, kosinüs benzerliği, Levenhstein uzaklığı, Qram benzerliği

1
Introduction

1.1 Literature Review

On this project, it is aimed to response with appropriate answers to the questions asked by the user. It is expected that the system will only answer the questions in specific sector because this question answering system will be prepared for a specific sector purpose.

1.2 Objective of the Thesis

The project is a question answering system that improves learning with the logic that is set up at the back. It is aimed to create more accurate and meaningful answers over time. Methods to be used;

1. Determining the needs and deficiencies in the first stage and eliminating the data and other needs for the project,
2. Collection of question and answer text data from specific sectors,
3. Cleaning of collected data,
4. Training of different approaches of gathered data deep learning framework, application of optimization, regulation methods,
5. Testing various methods to find similar questions to the question asked,
6. If there are no similar questions, asking the user for new questions and getting the correct answer,
7. If the system fails, re-arrange, train and test the new model.

1.3 Hypothesis

On this project my purpose is to give correct answers to the questions asked by users, although the level of achievement can not be reached in the first stage, it is aimed to reaching the acceptable levels of this level step by step.

2
General Information

The main purpose of this section to review projects that are already developed by others which have similarities with my project.

2.1 Related Works

These studies are about Question Answering and ChatBot systems, which are based on English data. "Deep Learning for Answer Sentence Selection [1]" is take part in this section that are very useful for my project about answer selection. "End-To-End Memory Networks [2]" will contribute significantly for implementing deep learning methods on the project.

2.1.1 Deep Learning for Answer Sentence Selection

Answer sentence selection is the task of identifying sentences that contain the answer to a given question. This is an important problem in its own right as well as in the larger context of open domain question answering. They propose a novel approach to solving this task via means of distributed representations, and learn to match questions with answers by considering their semantic encoding. This contrasts prior work on this task, which typically relies on classifiers with large numbers of hand-crafted syntactic and semantic features and various external resources. Their approach does not require any feature engineering nor does it involve specialist linguistic data, making this model easily applicable to a wide range of domains and languages. Experimental results on a standard benchmark dataset from TREC [3] demonstrate that "despite its simplicity" their model matches state of the art performance on the answer sentence selection task.

2.1.2 End-To-End Memory Networks

In this paper we introduce a neural network with a recurrent attention model over a possibly large external memory. The architecture is a form of Memory Network (Weston et al., 2015) [4] but unlike the model in that work, it is trained end-to-end, and hence requires significantly less supervision during training, making it more generally applicable in realistic settings. It can also be seen as an extension of RNNsearch [5] to the case where multiple computational steps (hops) are performed per output symbol. The flexibility of the model allows us to apply it to tasks as diverse as (synthetic) question answering and to language modelling. For the former approach is competitive with Memory Networks, but with less supervision. For the latter, on the Penn TreeBank [6] and Text8 [7] datasets this approach demonstrates comparable performance to RNNs and LSTMs. In both cases this model show that the key concept of multiple computational hops yields improved results.

2.2 Sample Data Sets

2.2.1 The Stanford Question Answering Dataset

Stanford Question Answering Dataset [8] (SQuAD) is a new reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer at each question is a segment of text, or span, from the corresponding reading passage. With 100,000+ question-answer pairs on 500+ articles, SQuAD is significantly larger than previous reading comprehension datasets.

Several sample questions and answers in this data set:

Question: What causes precipitation to fall?
Answer: gravity

Question: What is another main form of precipitation be-sides drizzle, rain, snow, sleet and hail?
Answer: graupel

Question: Where do water droplets collide with ice crystalsto form precipitation?
Answer: within a cloud

The paragraph in which the answers are obtained:

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of pre-cipitation include drizzle, rain, sleet, snow, grau- pel and hail ... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

2.2.2 Dialogue System Technology Challenge (DSTC)

The Dialog State Tracking Challenge (DSTC) [9] is an on-going series of research community challenge tasks. Each task released dialog data labeled with dialog state information, such as the user's desired restaurant search query given all of the dialog history up to the current turn. The challenge is to create a “tracker” that can predict the dialog state for new dialogs. In each challenge, trackers are evaluated using held-out dialog data.

DSTC1 [10] used human-computer dialogs in the bus timetable domain. Results were presented in a special session at SIGDIAL 2013. DSTC1 was organized by Jason D. Williams, Alan Black, Deepak Ramachandran, Antoine Raux.

DSTC2/3 [11] used human-computer dialogs in the restaurant information domain. Results were presented in special sessions at SIGDIAL 2014 and IEEE SLT 2014. DSTC2 and 3 were organized by Matthew Henderson, Blaise Thomson, and Jason D. Williams.

DSTC4 [12] used human-human dialogs in the tourist information domain. Results were presented at IWSDS [13] 2015. DSTC4 was organized by Seokhwan Kim, Luis F. D'Haro, Rafael E Banchs, Matthew Henderson, and Jason D. Williams.

DSTC5 [14] used human-human dialogs in the tourist information domain, where training dialogs were provided in one language, and test dialogs were in a different language. Results were presented in a special session at IEEE SLT [15] 2016. DSTC5 was organized by Seokhwan Kim, Luis F. D'Haro, Rafael E Banchs, Matthew Henderson, Jason D. Williams, and Koichiro Yoshino.

DSTC6 [16] consisted of 3 parallel tracks: End-to-End Goal Oriented Dialog Learning, End-to-End Conversation Modeling, and Dialogue Breakdown Detection [17]. Results will be presented at a workshop immediately after NIPS [18] 2017. DSTC6 is organized by Chiori Hori, Julien Perez, Koichiro Yoshino, and Seokhwan Kim. Tracks were organized by Y-Lan Boureau, Antoine Bordes, Julien Perez, Ryuichi Higashinaka, Chiori Hori, and Takaaki Hori.

A few sample questions and answers for the restaurant booking system [19]:

Customer: Good morning
Bot: Hello!
Client: Can you make a restaurant reservation in rome in a cheap price range
Bot: I'm on it, Any preference on a type of cuisine
Client: With spanish food
Bot: How many people would you in your party
Customer: We will be six
Bot: Ok let me look into some options for you

3
Feasibility

3.1 Technical Feasibility

As technical feasibility study, the software, hardware, labour force, time, legal, economical needs for the project is defined on the following sections.

3.1.1 Software Feasibility

- **Operating system**
  This project will be programmed on Linux operating system however it can be run on any operating system.

- **Development Language**
  Python has been chosen as programming language. The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features. Python is a robust programming language and provides an easy usage of the code lines, maintenance can be handled in a great way, and debugging can be done easily too. It has gained importance across the globe as computer giant Google has made it one of its official programming languages.

- **System Requirements**

  The web interface requires the following dependencies:

  - python 3.5
  - tensorflow (tested with v1.0)
  - numpy
  - CUDA (for using GPU)
  - nltk (natural language toolkit for tokenized the sentences)
  - tqdm (for the nice progression bars)

  For The web interface requires these packages:

  - django
  - channels
  - Redis
  - asgi-redis

3.1.2 Hardware Feasibility

For implementing this system, it is required at least one computer which has high performance gpu processor. The user need just any kind of device and internet connection to interact with chatbot. Minimum hardware requirements tables are shown in Table 3.1 and in Table 3.2.

Table 3.1 Minimum hardware requirements for Android Mobile Application

CPU	1.6GHz
RAM	256MB
Operation System	Android 4.3 or later versions

Table 3.2 Minimum hardware requirements for Desktop Application

CPU	2.1 GHz
RAM	1GB
Operation System	Windows, Linux, MacOs

3.2 Labor Force and Time Feasibility

Gannt Diagram is shown in Figure 3.1.

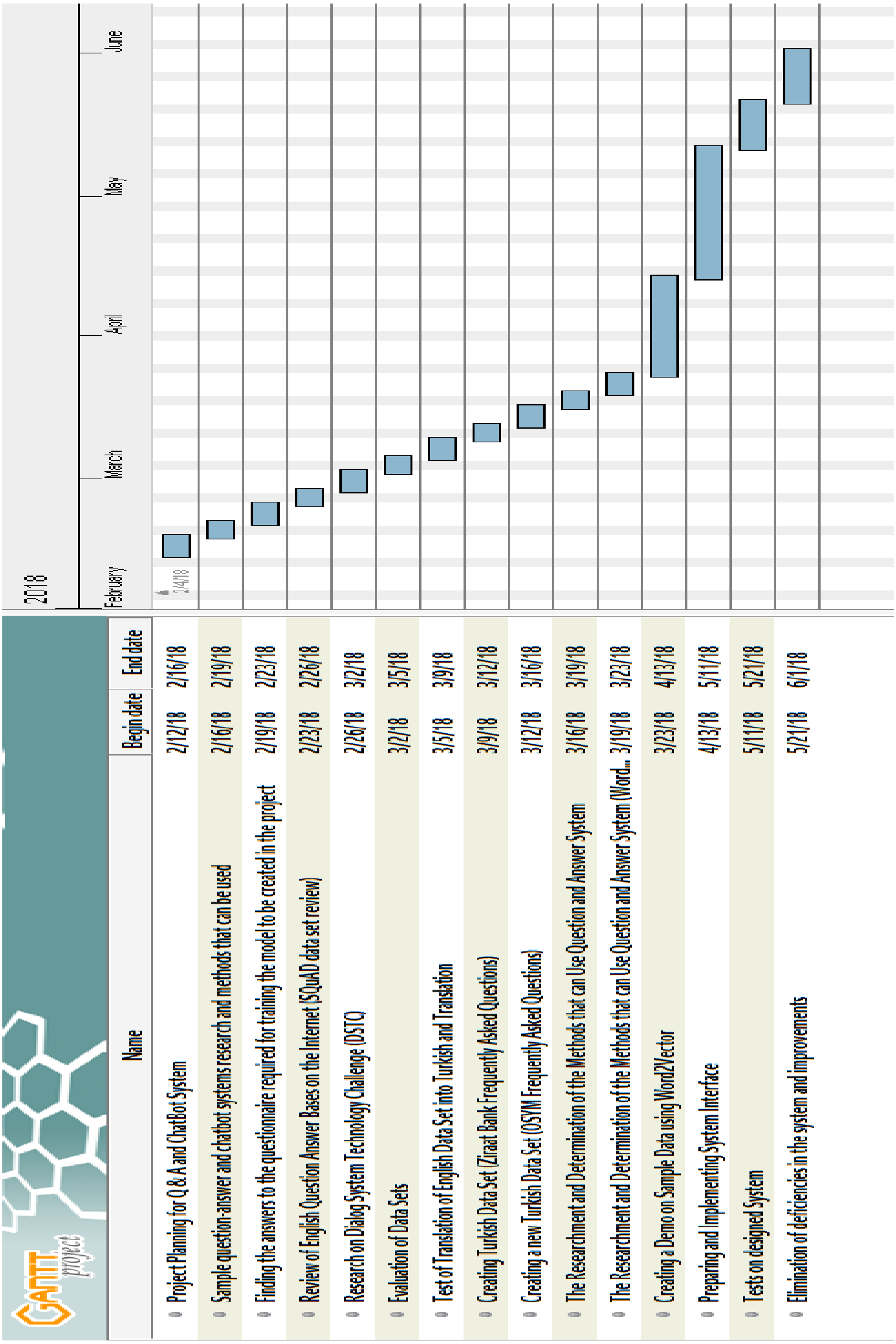

Figure 3.1 Gantt diagram

3.3 Legitimate Feasibility

Software which is used within the project does not face any legal issues. All of the software used in the project contain license requirements.

3.4 Economic Feasibility

The estimated hardware cost required for the project is shown in Table 3.3.

Table 3.3 Hardware Cost Table

Hardware	Estimated Cost
high performance gpu processor(NVIDIA GeForce GTX 850M)[20]	1050 TL

The estimated working fee for this project is shown in Table 3.4.

Table 3.4 Personal Cost Table

Personal	Total Hours	Estimated Cost
Data Analyst	200	4500 TL
Python Developer who has experienced machine learning	200	5500 TL
Web Interface Developer	200	4000 TL

4
System Analysis

When the studies has been reviewed for the question-answer system , it can be found significant hints for the implementation of system. The most important thing about that kind of system is always preparing an appropriate big training data. Because of insufficiency question-answer pairs data in Turkish language various methods has to be tried to create a data set. There are two ways currently for creating data set, first way translation one of data set from any kind of language (probably in English) to Turkish language, a second way is create a new data set in Turkish manually. Currently, it is not certain which method is useful for the project.
For the question-answering system this path will be followed:

1. The user asks a question on the program.
2. The program take apart the questions into words, processes the words.
3. The system finds closest question to obtained question.
4. If the question is found below a certain level of similarity, or a similar question is not found, the user is asked to answer a few decisive questions to find the correct answer.
5. If the answer is found, that will be returned; otherwise, the user is directed to an authority.
6. If the chosen answer is satisfactory for the user, the new question-answer binary is added to the database. Scoring and Evaluation can be used for that.
7. After a certain period of time, the data is retrained to increase the success of system.

Workflow Schema for question answering system is shown in Figure 4.1.

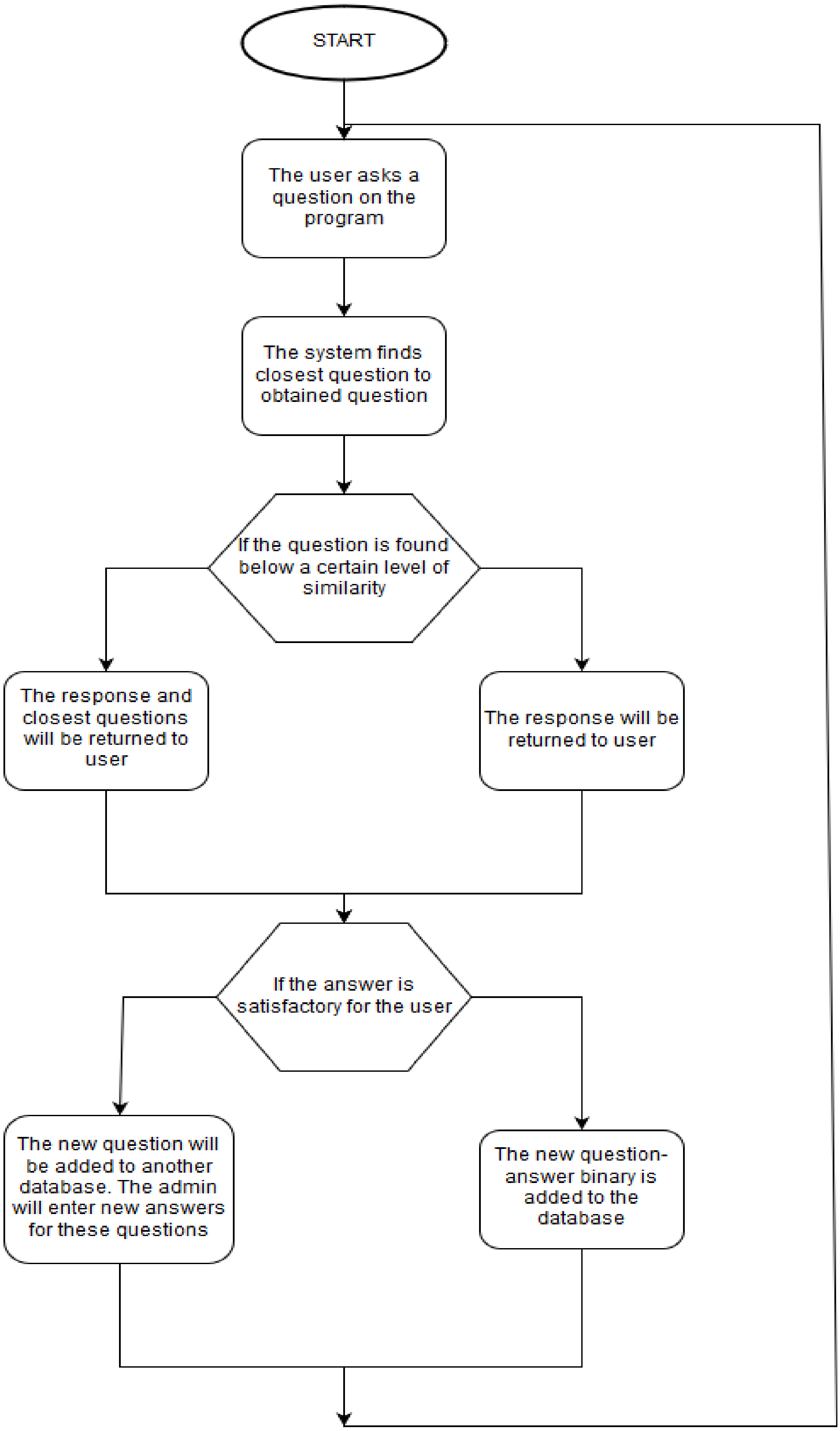

Figure 4.1 Workflow Schema for question answering system

5
System Design

Project details and descriptions also architectural design of project are explained in this chapter.

The question answering system can be formulated as follows: Given a question ***q*** and an answer candidate pool a1,a2,⋯,an for this question, I aim to search for the best answer candidate a. An answer is a token sequence with an arbitrary length, and a question can correspond to multiple ground-truth answers. In testing, the candidate answers for a question may not be observed in the training phase. Answer selection is one of the essential components in typical question answering (QA) systems. It is also a stand-alone task with applications in knowledge base construction and information extraction.

The major challenge of this task is that the correct answer might not directly share lexical units with the question. Instead, they may only be semantically related. Moreover, the answers are sometimes noisy and contain a large amount of unrelated information.

5.1 Dataset

The dataset which is required for implementation of my project, has been obtained from section of frequently asked questions in Ziraat Bank website [21]. On this section of website there were approximately 400 pair of question-answer about different subjects in banking sector.

This size of dataset is not enough for implementation my deep learning model. However the dataset can be used for beginning. Using desktop application this dataset can be enlarged for the deep learning model.

Some of sample questions and answers are shown in Figure 5.1.

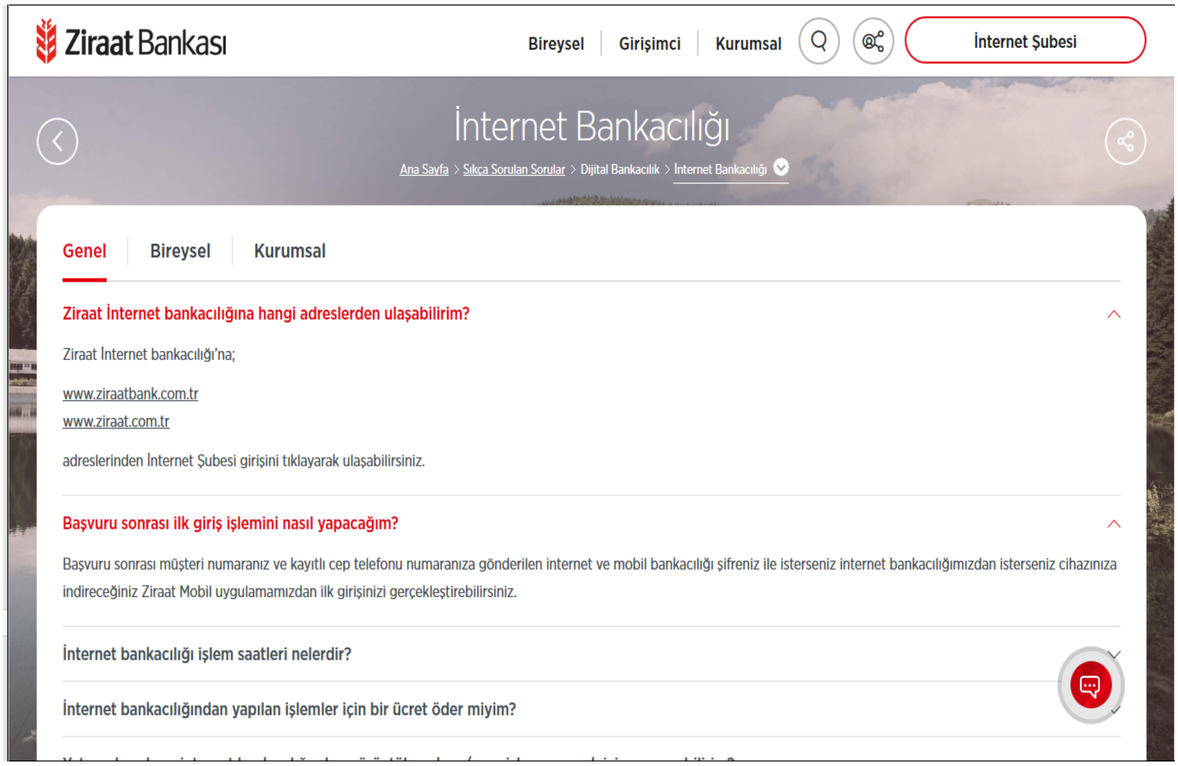

Figure 5.1 Frequently asked questions in Ziraat Bank website

5.2 Software Design

5.2.1 Implementation of Model on Web Interface

The models in this work are implemented with Tensorflow [22], and all experiments are processed in a GPU cluster. I use the accuracy on validation set to locate the best epoch and best hyper-parameter settings for testing.

The word embedding is trained by Word2vec [23], and the word vector size is 300. Word embeddings are also parameters and are optimized as well during the training. Stochastic Gradient Descent [24] (SGD) is the optimization strategy. I tried different margin values, such as 0.05, 0.1 and 0.2, and finally fixed the margin as 0.2.

As an network architecture, multi-layer perceptron model has been used on web interface. This model has 5 layer in total. (1 of them as an Input Layer, 3 of them as an Hidden Layer, 1 of them as an output Layer).

Each word vector, which is obtained of the questions in the data set, is used as input. The output layer has the indexes of the sentences placed in order. During training of the model after every epoch, the sentence index in the output layer is changed to 1 as a binary value. Total number of questions in dataset is currently 488, which are used for training the model.

After 100 epochs the training phase of model is ended. Leaning rate is set to 0.001 and dropout rate is set to 0.5. This approach is very similar to the text classification methods, which use artificial neural networks.

Multi-layer perceptron model for question answering is shown 5.2.

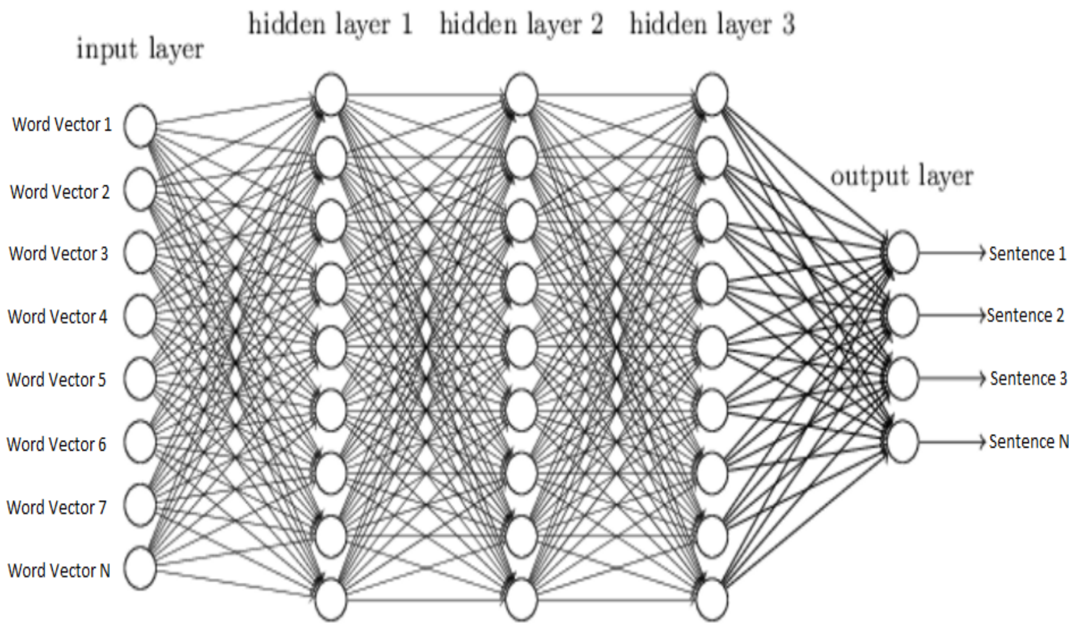

Figure 5.2 Multi-layer perceptron model for question answering

5.2.2 Desktop Application Design

The size of dataset is not enough for implementation my deep learning model. Therefore, in some cases the web application can not reply with correct answer.

The solution for this problem is enlarging dataset for the deep learning model using desktop application. The mentality of desktop application is not complicated. The program finds closest question to obtained question using some similarity methods such as cosine similarity [26], levenshtein distance [27] and Qqram similarity [28]. If the user is satisfied with answer then he has to click Ok button so the new question and answer will be added on database.

If the user is not satisfied with answer, the user can choose one of 5 most similar questions. In this case, a new answer will be shown on text area. If the user is still not satisfied with answer, then he has to click on Not Ok button, the question and answer will be added in another database. The admin will enter new answers after a while for the question.

Using this desktop application, enlarging dataset for the deep learning model can be possible.

UML diagram of desktop application is shown in Figure 5.3.

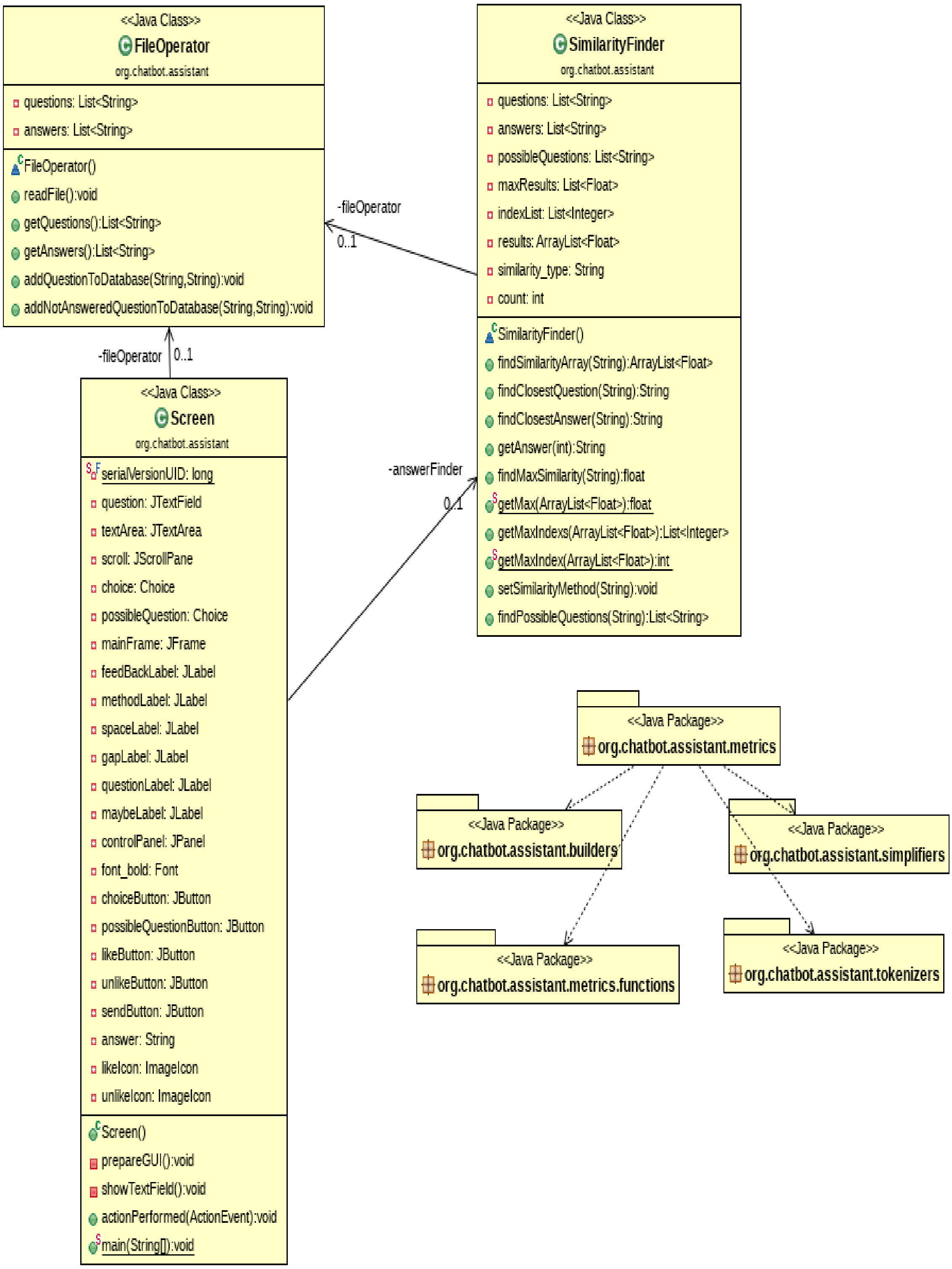

Figure 5.3 UML diagram for desktop application

6
Application

In this section screenshots of the system will be shown. Each screenschot represents a different module of system.

6.1 Web Application

The main page of Web application is shown in Figure 6.1. For now it is only for a demo homepage. Users can write their own question in text label as an input. After a couple of milliseconds the system will print the answer to the screen.

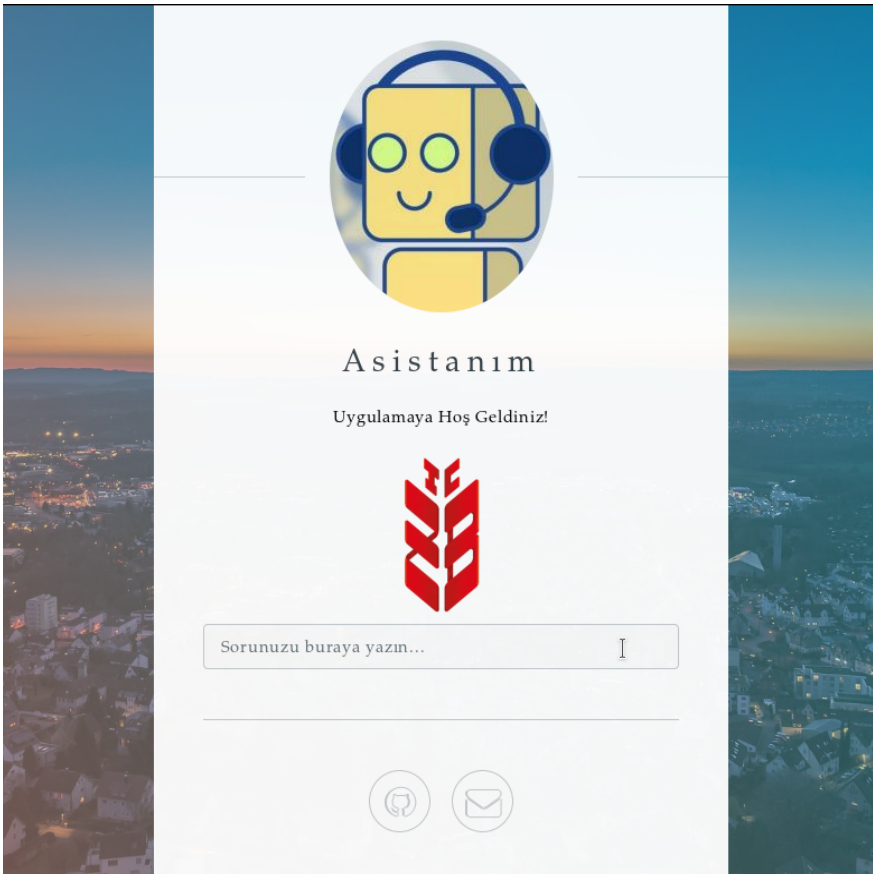

Figure 6.1 Web Application Interface

User can ask a question using web application. It is shown in Figure 6.2.

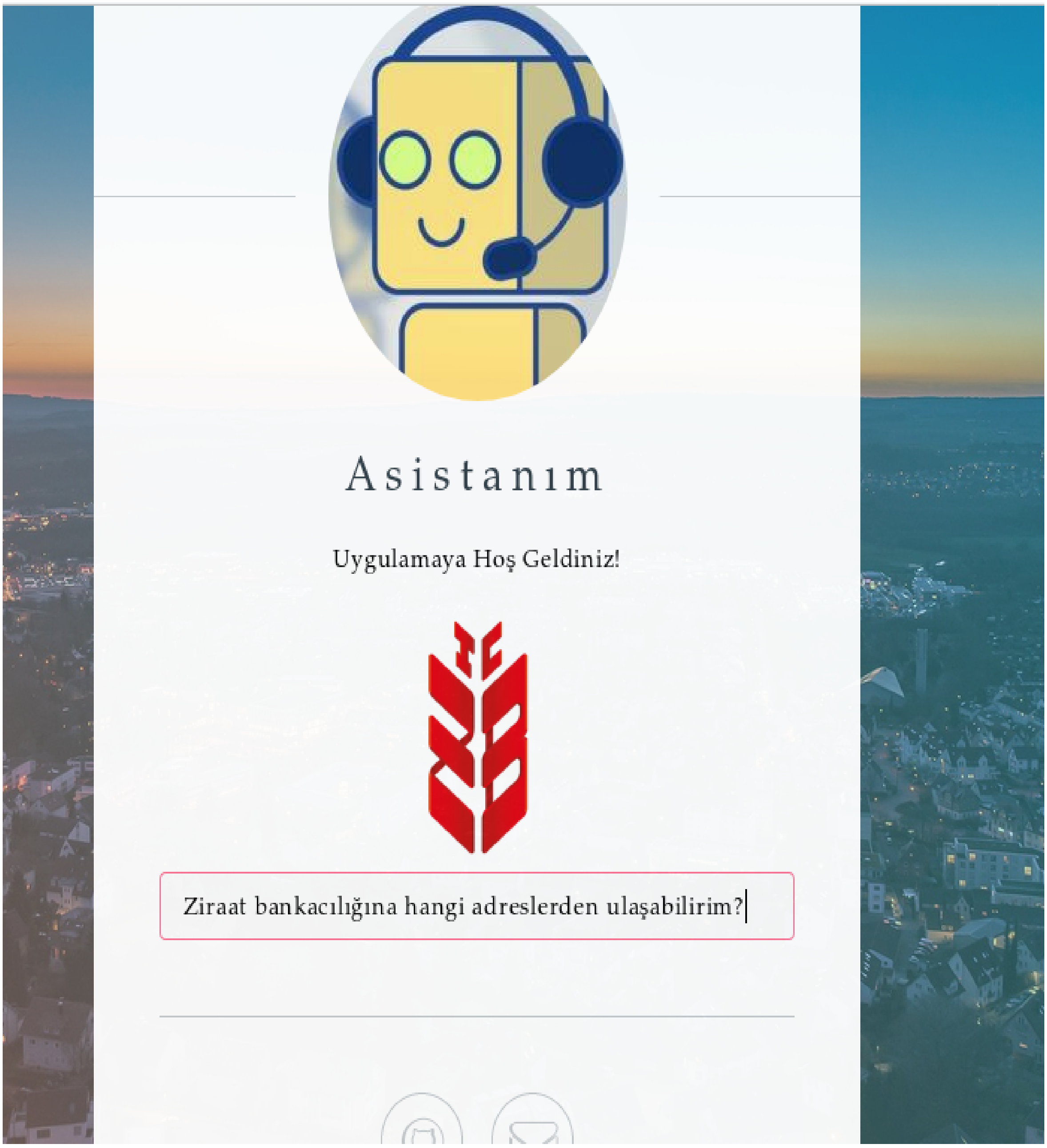

Figure 6.2 User asks own sample question using web application

The answer and question will be shown on text area. It is shown in Figure 6.3.

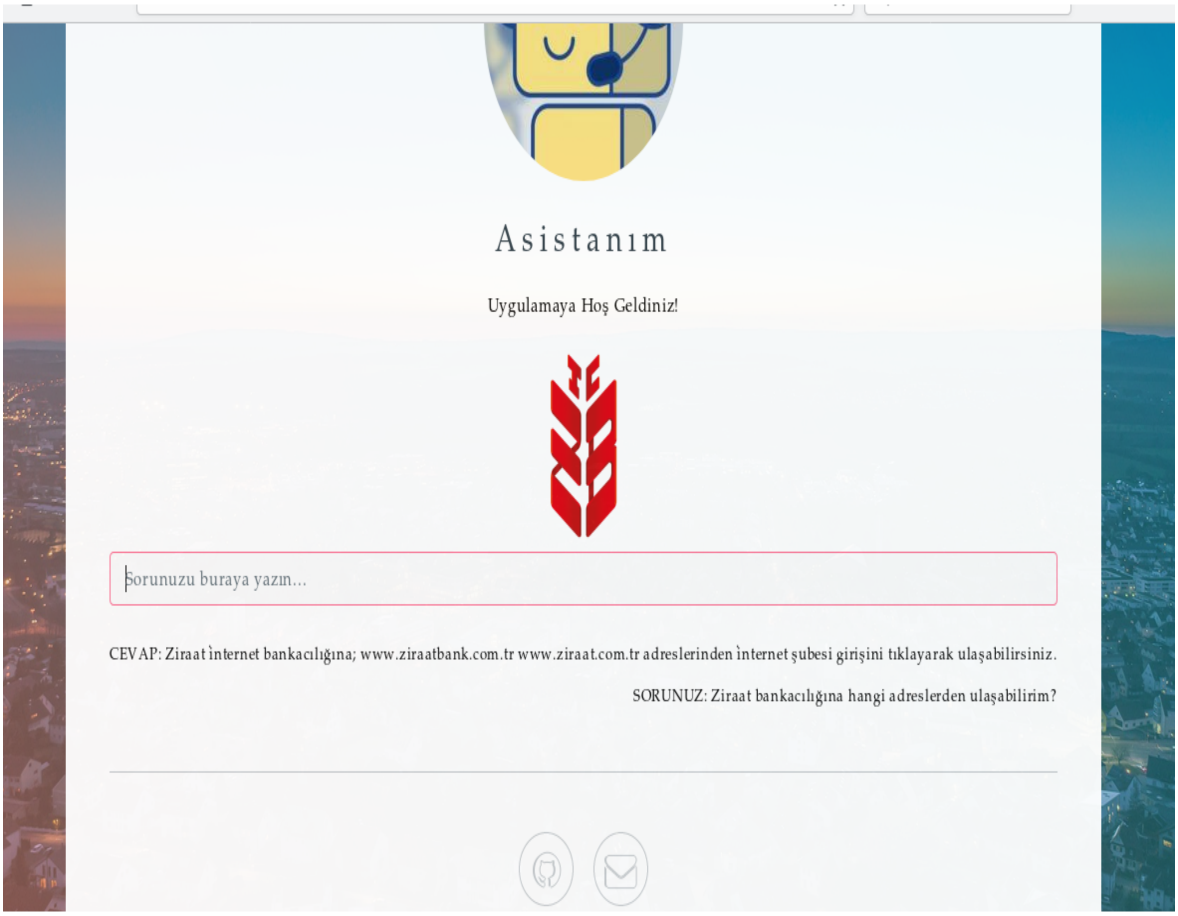

Figure 6.3 The answer and question are shown on web application

If the user change previously asked question and asks a new question, the new answer will be shown on text area. It is shown in Figure 6.4.

Figure 6.4 The new answer and question are shown on web application

6.2 Desktop Application

The main page of desktop application is shown in Figure 6.5. Users can write their own question in text label as an input. After a couple of milliseconds the system will print the answer to the screen.

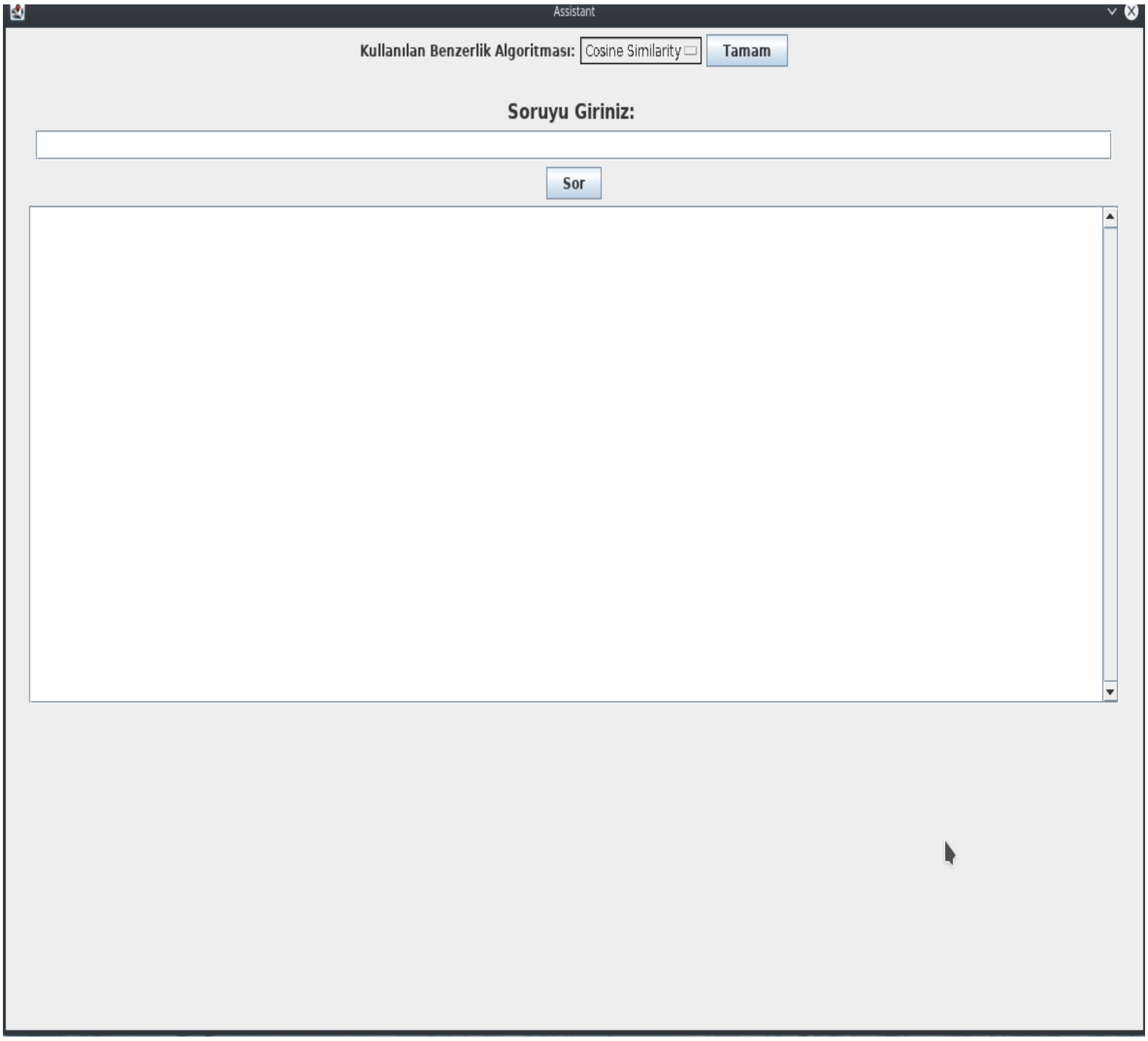

Figure 6.5 Desktop Application Interface

User can ask a question using desktop application. For this sample question :"Ziraat internet bankacılığına hangi adreslerden ulaşabilirim?" the answer, most similar question and similarity rate are shown on screen. Due the question is already in database, similarity rate is 1.0 and correct answer is seen in text area. It is shown in Figure 6.6.

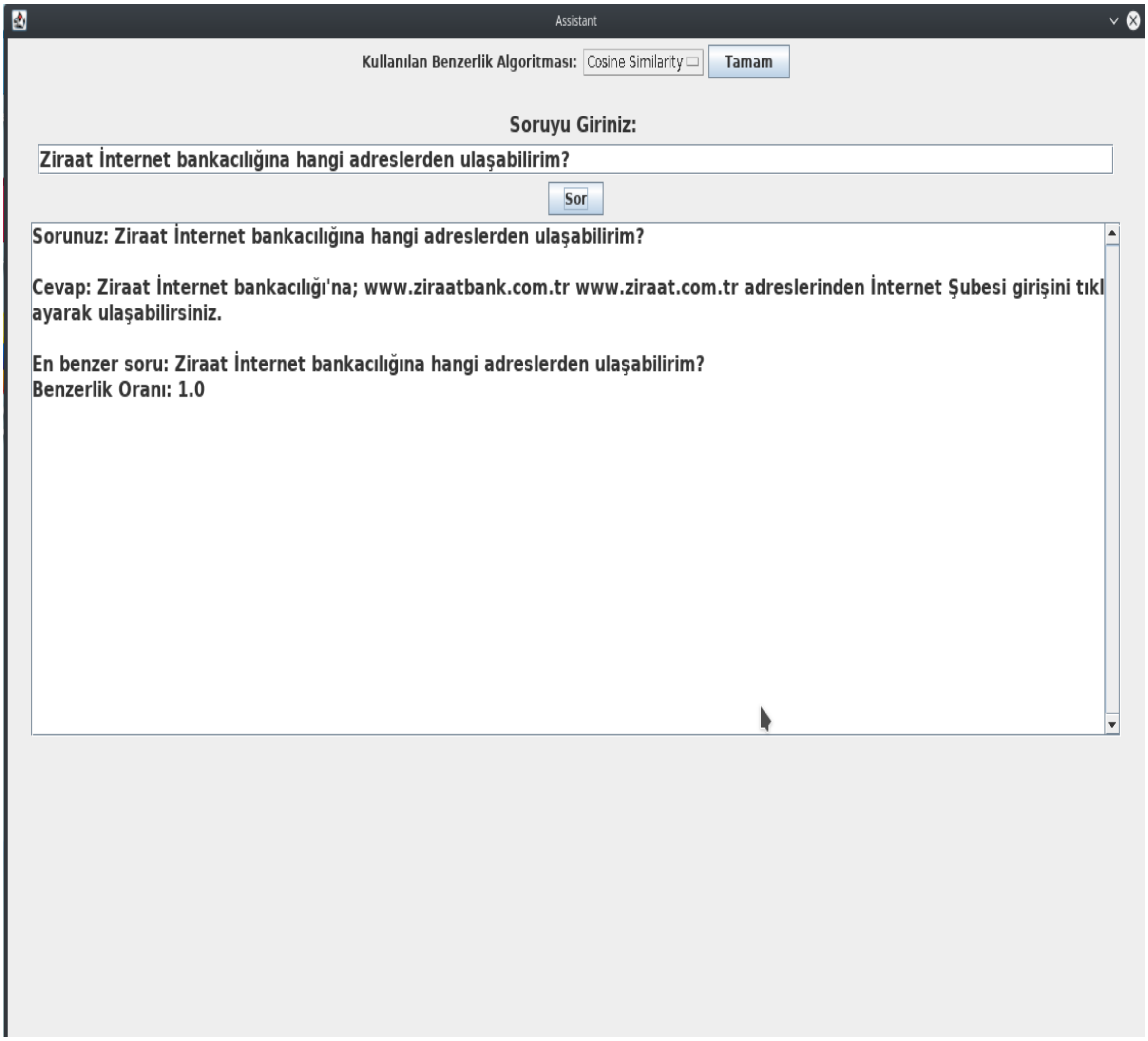

Figure 6.6 User asks own sample question using desktop application

If the user change just one word on previously asked question and asks this question: "Ziraat internet bankacılığına hangi adreslerden ulaşabilirim?", similarity rate will be 0.9 and same answer will be seen on text area. It is shown in Figure 6.7.

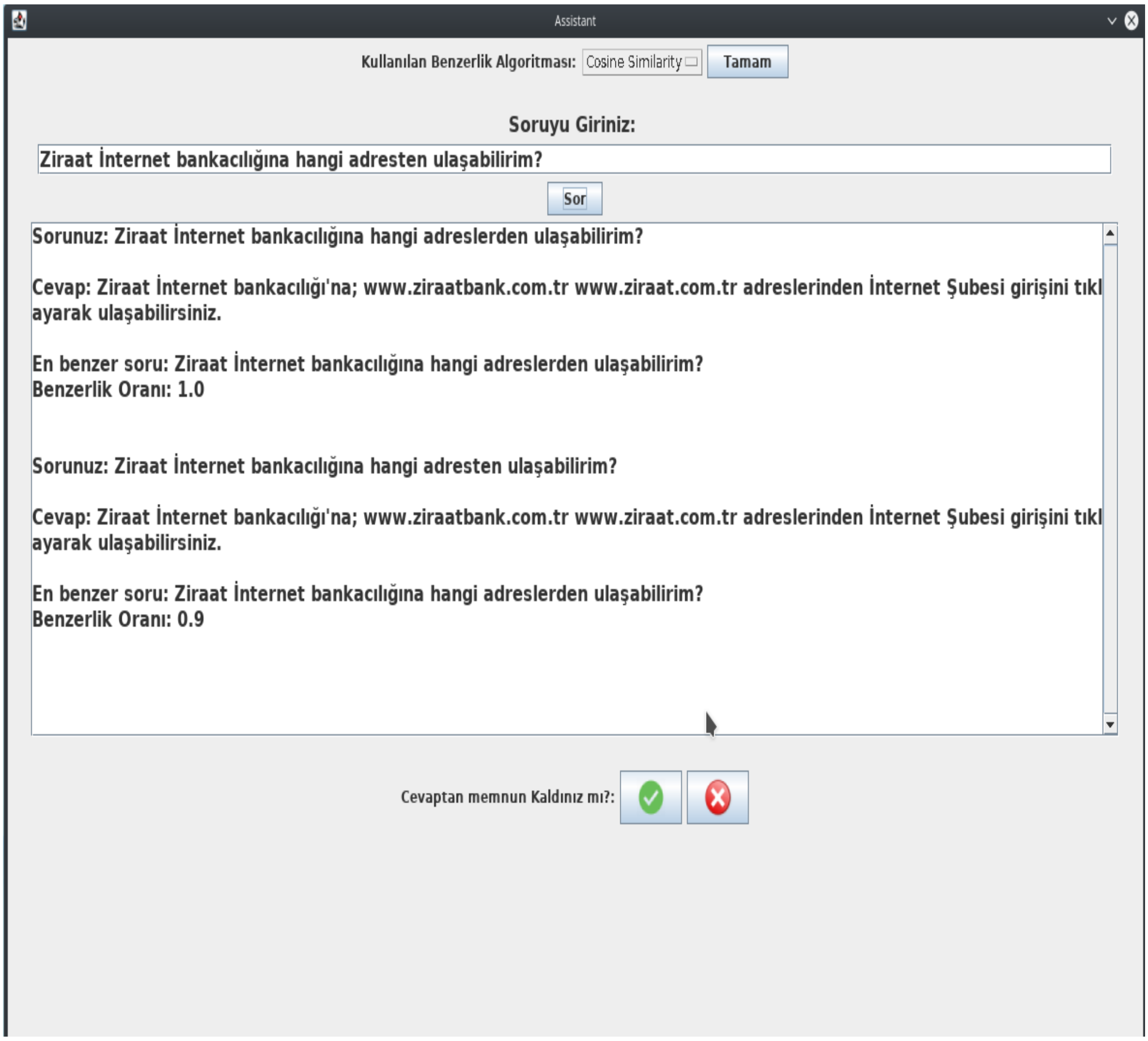

Figure 6.7 User asks own sample question using desktop application-2

At the bottom of text area an important question is asked to user. It means "Are you satisfied with answer?". Also, Ok and Not ok button take part in right side of this question. It is shown in Figure 6.8.

Figure 6.8 "Are you satisfied with answer?" is shown screen in desktop application

If the user click on Ok button, the question and answer will be added in database. The message "The question is added in database" is shown in Figure 6.9.

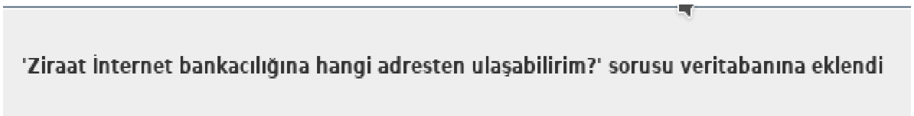

Figure 6.9 The Message is shown in Desktop Application

If the user click on Not Ok button, the question and answer will be added in another database. The admin will enter new answers after a while for the question. The message "For your question will be found better answer in soon" is shown in Figure 6.10.

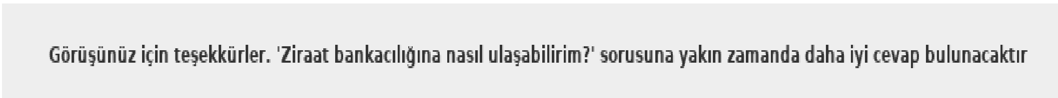

Figure 6.10 If the user click on Not Ok button, the message will be shown in Desktop Application

If the user change the question more deeply, the similarity rate between most similar question will drop. For this example question :"Ziraat bankacılığına nasıl ulaşabilirim?" similarity rate will be 0.7 and correct answer will be able to found by application. It is shown in Figure 6.11.

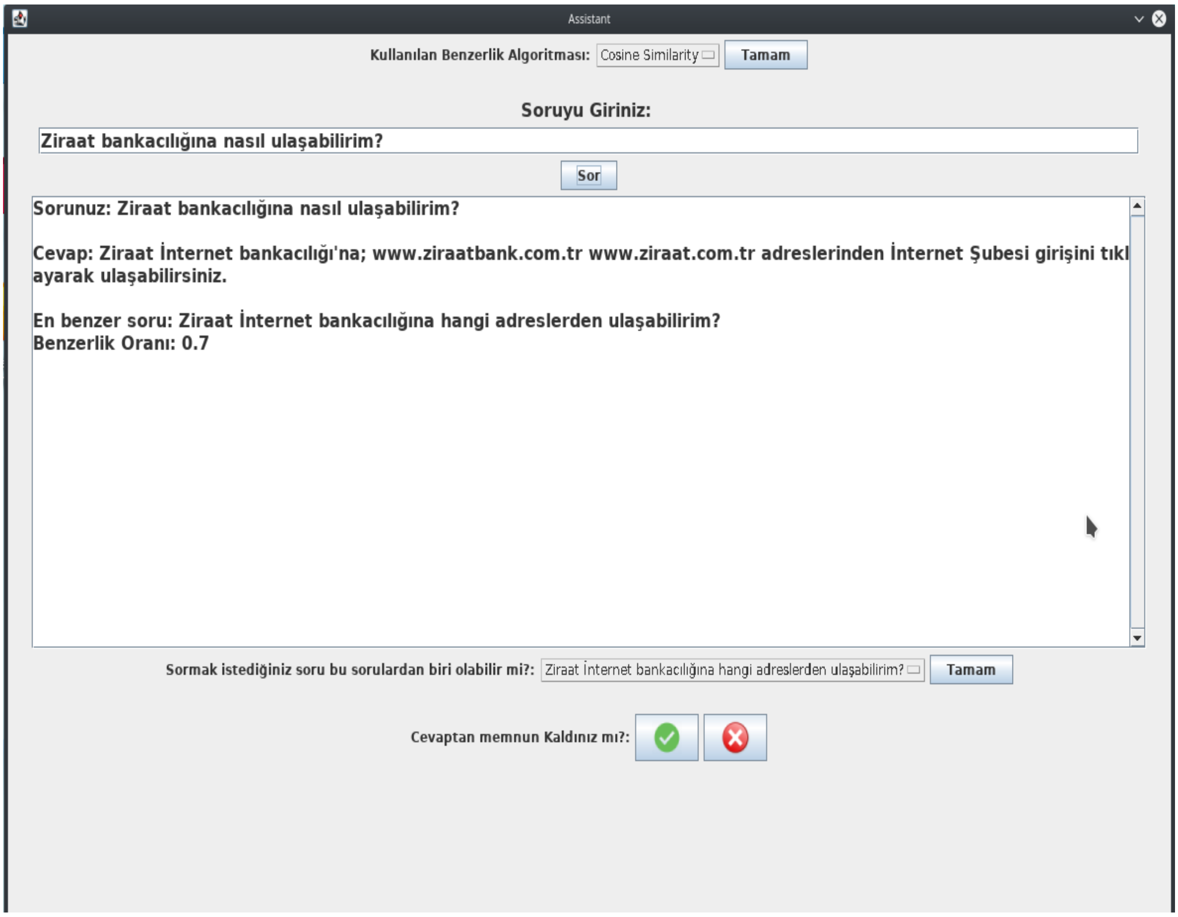

Figure 6.11 The users asks a new question using desktop application

In this application, the user can choose the method which is used to find most similar question and its answer. For a now possible methods are cosine similarity, levenshtein and qram. It is shown in Figure 6.12.

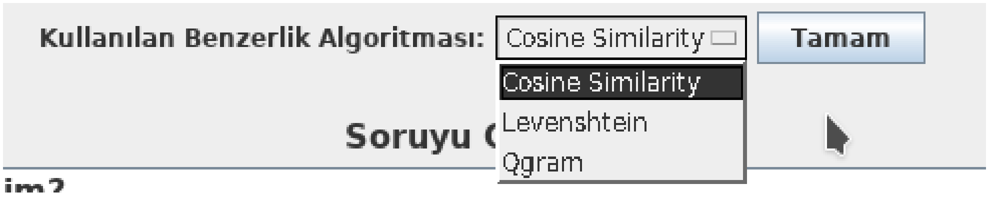

Figure 6.12 Similarity Methods

If the user is not satisfied with question, the user can choose one of 5 most similar questions. In this case, a new answer will be shown on text area. Possible questions section is shown in Figure 6.13.

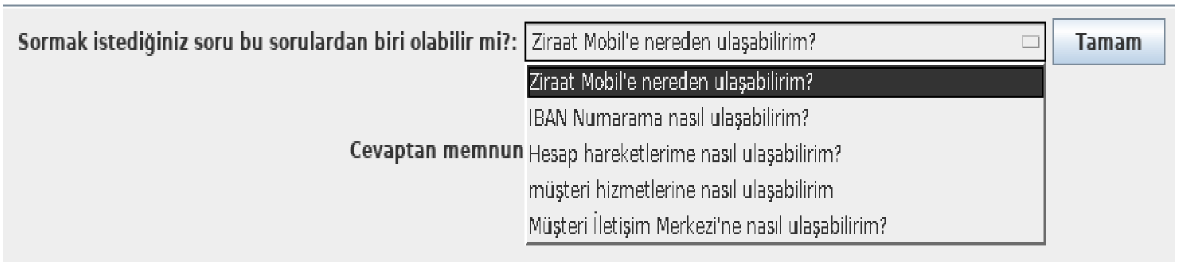

Figure 6.13 Possible questions section

6.3 Mobile Application

Before the main page of mobile application is shown, the splash screen is shown during a couple of seconds on android application. It is shown in Figure 6.14.

Figure 6.14 Splash screen on android mobile application

The main page of mobile application is shown in Figure 6.15. Users can write their own question in text label as an input. After a couple of milliseconds the system will print the answer to the screen.

Figure 6.15 Main Screen of Mobile Application

User can ask a question using mobile application. For this sample question :"Kartım kayboldu, ne yapabilirim?" the answer, most similar question and similarity rate are shown on screen. It is shown in Figure 6.16.

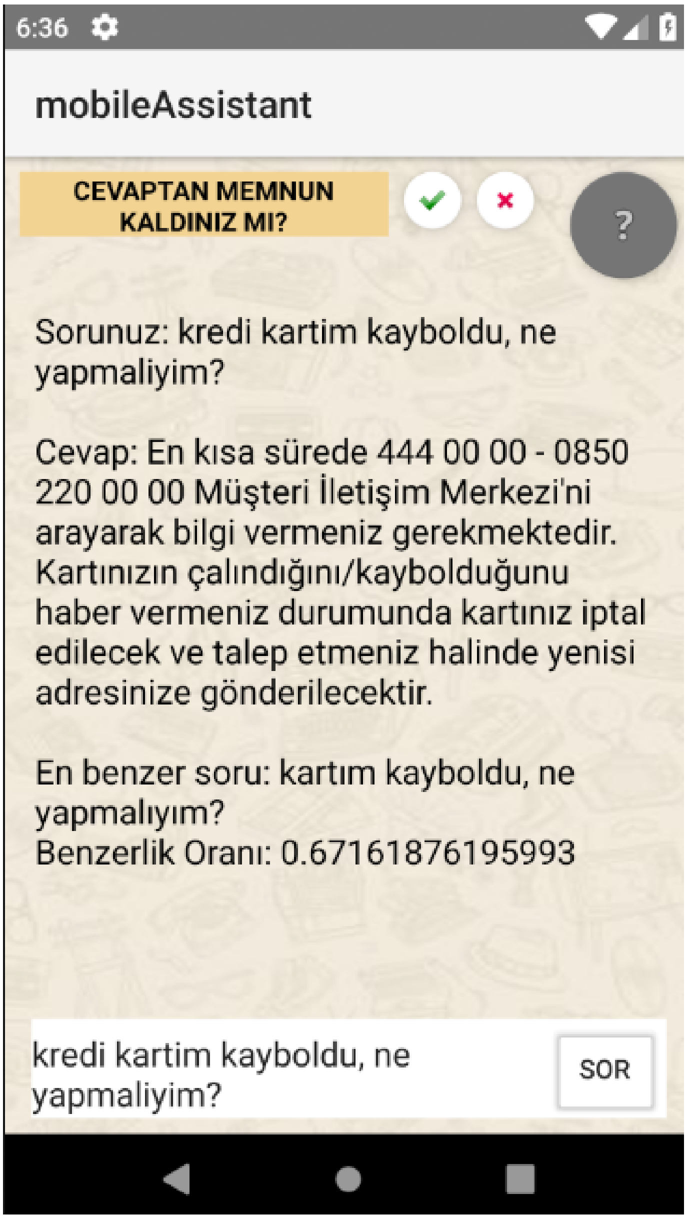

Figure 6.16 User asks own sample question using android application

At the above of text area an important question is asked to user. It means "Are you satisfied with answer?". Also, Ok and Not ok button take part in right side of this question. It is shown in Figure 6.17.

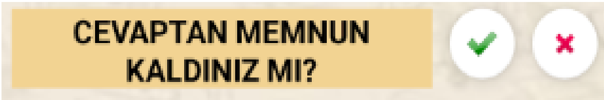

Figure 6.17 "Are you satisfied with answer?" is shown screen in mobile application

If the user click on Ok button, the question and answer will be added in database. The message "The question is added in database" is shown in Figure 6.18.

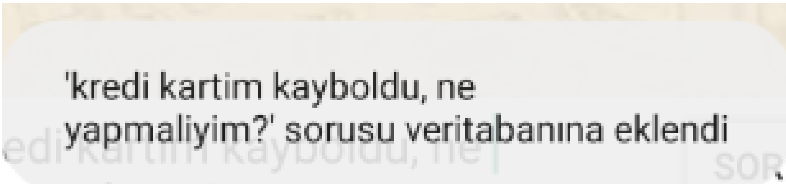

Figure 6.18 The Message is shown in Mobile Application

If the user click on Not Ok button, the question and answer will be added in another database. The admin will enter new answers after a while for the question. The message "For your question will be found better answer in soon" is shown in Figure 6.19.

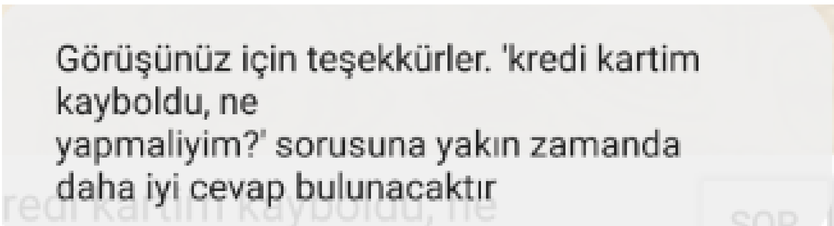

Figure 6.19 If the user click on Not Ok button, the message will be shown in Desktop Application

If the user is not satisfied with question, the user can choose one of 5 most similar questions. In this case, a new answer will be shown on text area. Possible questions section is shown in Figure 6.20.

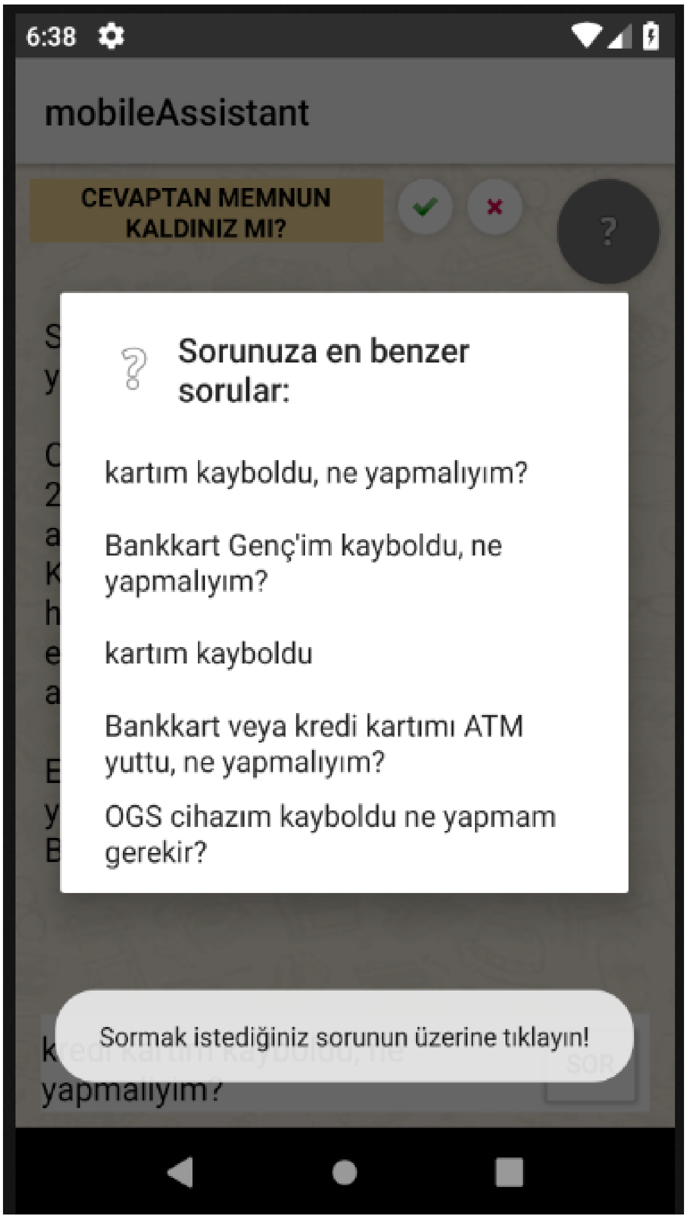

Figure 6.20 Possible questions section

7
Experimental Results and Performance Analysis

7.1 Experimental Results

An evaluation model was used to measure user satisfaction with the question answering assistants. 10 different participants were given hard copy form of 50 question samples in the dataset to use the question-answer assistant. 20 sample of these questions are as follows:

1. Ziraat İnternet bankacılığına hangi adreslerden ulaşabilirim?
2. İnternet bankacılığı işlem saatleri nelerdir?
3. İnternet bankacılığından yapılan işlemler için bir ücret öder miyim?
4. Yatırım hesabımı internet bankacılığından görüntülemek ve/veya işlem yapmak için ne yapabilirim?
5. İnternet bankacılığından açtığım hesabımı kapatmak istiyorum. Ne yapmalıyım?
6. İnternet bankacılığı şifremi unuttum/bloke oldu. Ne yapabilirim?
7. Cep telefonu bilgilerimi nasıl güncelleyebilirim?
8. Harçlık avans işlem bilgilerini Mobil ve İnternet Bankacılığı'ndan görebiliyor muyum?
9. Yurt dışından getirttiğim telefon için İnternet Bankacılığı ve Mobil'den ödeme yapılıyor mu?
10. Ziraat Bankası güvenliğim için neler yapıyor?
11. Güvenliğim için nelere dikkat etmeliyim?
12. Bireysel internet bankacılığına nasıl başvurabilirim?
13. İnternet bankacılığından hangi işlemleri yapabilirim?

14. İşlem limitlerimi nasıl değiştirebilirim?
15. Kurumsal İnternet Bankacılığı şifremi unuttum/bloke oldu. Ne yapabilirim?
16. Ziraat Mobil'e nereden ulaşabilirim?
17. Dekont almak istiyorum. Ne yapmam gerekir?
18. QR Kod ile hangi işlemleri yapabilirim?
19. Ziraat ATM'lerinden hangi işlemleri yapabilirim?
20. ATM'lerden günlük en fazla ne kadar nakit çekebilirim?

Participants were requested to ask similar or related questions to these questions using web application and desktop application. After each response of the application, the users evaluated the system with a score of 1-5. This evaluating was based on the following 4 criteria:

1. Were the answers given by Question Answering Assistant correct? (**Quality**)
2. Are you satisfied with answers? (**Quantity**)
3. Were the answers given by Question Answering Assistant relevant to subject? (**Relation**)
4. Were the answers given by Question Answering Assistant clear? (**Manner**)

The result of the rating is shown in Table 7.2, which have done by the participants:

Table 7.1 Evalution scores

Criterias	**1**	**2**	**3**	**4**	**5**
Quality	1	0	1	2	6
Quantity	0	1	0	2	7
Manner	0	0	0	1	9
Relation	0	0	0	2	8

According to this frequency values accuracy can be calculated using sample mean formula.

Table 7.2 Mean Value of Rating Scores based on criterias

Criterias	Mean Value of Rating Scores
Quality	4.2
Quantity	4.5
Manner	4.9
Relation	4.8

Accordingly, the overall success of the system was estimated to be approximately 92%.

7.2 Performance Analysis

Question Answering Assistant has been tested on two different platforms (Windows and Android). This test was examined according to criteria such as time, speed, success.

First, when the application created on the mobile platform is examined, the opening time of the program takes 1-2 ms. No stuck or waiting period occurs during usage.

Similarly, when we look at the desktop and web application, the opening time of the program lasts in ms. There is no hanging or waiting during usage.

8
Conclusion

In this project, a text based question answering assistant was implemented which can be used for different sectors. The system was designed that finds the closest question among the previously asked questions by user and also the system returns correct answer to the user. In case of the answer could not be found, this system informs the user and finds soon a suitable answer. If the chosen answer is satisfactory for the user, the new question-answer binary is added to the database. After a certain period of time, the data is retrained to increase the success of system.

Two different methods has been used to find best answer. First method is to use neural network model (LSTM) and second method is to use string similarity methods(Cosine similarity, Levenhstein distance, Qgram similarity) to find appropriate answer. Although both methods are considered successful enough, the neural network model needs more data sets.

When the experimental results of the project were reviewed, it was found that the participants evaluated the system very successfully. In 4 different criteria (Quality, Quantity, Manner, Relation) it was scored over 4 out of 5. Overall score is 4.6 points, which is about 92%.

On this project my purpose was to give correct answers to the questions asked by users, although the level of achievement could not be reached in the first stage, it was reached the acceptable levels of this level step by step. There are not many studies done in Turkish about question and answer systems. I hope this work will be useful for other studies.

References

[1] Deep Learning for Answer Sentence Selection *https://arxiv.org/abs/1412.1632*, Lei Yu, Karl Moritz Hermann, Phil Blunsom, Stephen Pulman (Submitted on 4 Dec 2014).

[2] End-To-End Memory Networks *https://arxiv.org/abs/1503.08895*, Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus (Submitted on 31 Mar 2015 (v1), last revised 24 Nov 2015)

[3] Text REtrieval Conference (TREC) Question Answering Collections *https://trec.nist.gov/data/qa.html*, Voorhees, E. Tice, D. "Building a Question Answering Test Collection", Proceedings of SIGIR-2000, July, 2000, pp. 200-207

[4] Memory Networks *https://arxiv.org/abs/1410.3916*, Jason Weston, Sumit Chopra, Antoine Bordes Submitted on 15 Oct 2014.

[5] Neural Machine Translation by Jointly Learning to Align and Translate *https://arxiv.org/abs/1409.0473*, Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio Submitted on 1 Sep 2014 , last revised 19 May 2016

[6] A Sample of the Penn Treebank Corpus *https://www.kaggle.com/nltkdata/penn-tree-bank*, University of Pennsylvania 2017.

[7] Embedding made from the text8 Wikipedia dump. *https://data.world/jaredfern/text-8-w-2-v*

[8] SQuAD: 100,000+ Questions for Machine Comprehension of Text *https://rajpurkar.github.io/SQuAD-explorer/*, Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang (Submitted on 16 Jun 2016, last revised 11 Oct 2016)

[9] The Dialog State Tracking Challenge (DSTC) is an on-going series of research community challenge tasks. *https://www.microsoft.com/en-us/research/event/dialog-state-tracking-challenge/*,

[10] The Dialog State Tracking Challenge *https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/dstc2013.pdf*, Jason D. Williams, Antoine Raux, Deepak Ramachadran, and Alan Black. Proceedings of the SIGDIAL 2013 Conference, Metz, France, August 2013.

[11] The Second Dialog State Tracking Challenge *https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/summaryWriteup.pdf*, Matthew Henderson, Blaise Thomson, and Jason D. Williams. Proceedings of SIGDIAL 2014 Conference, Philadelphia, USA, June 2014.

[12] The Third Dialog State Tracking Challenge *https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/writeup.pdf*, Matthew Henderson, Blaise Thomson, and Jason D. Williams. Proceedings IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoe, USA, December 2014.

[13] 2015 International Workshop Series on Spoken Dialogue Systems Technology *http://wikicfp.com/cfp/servlet/event.showcfp=eventid=38683=copyownerid=66338*,

[14] The Fourth Dialog State Tracking Challenge *https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/dstc4final-1.pdf*, Seokhwan Kim, Luis F. D'Haro, Rafael E Banchs, Matthew Henderson, and Jason D. Williams. Proceedings IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoe, USA, December 2014.

[15] 2016 IEEE Workshop on Spoken Language Technology *https://www2.securecms.com/SLT2016//Default.asp*, 13–16 December 2016 • San Diego, California

[16] The Fifth Dialog State Tracking Challenge. Seokhwan Kim, Luis F. D'Haro, Rafael E Banchs, Matthew Henderson, Jason D. Williams, and Koichiro Yoshino. Proceedings IEEE Spoken Language Technology Workshop (SLT), San Diego, USA, December 2016.

[17] The Dialogue Breakdown Detection Challenge *https://pdfs.semanticscholar.org/c1c1/34f2a411d70687142cb63da14e9d55ffbfb6.pdf*, Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, Michimasa Inaba, NTT Media Intelligence Laboratories , Honda Research Institute Japan Co., Ltd., Toshiba Corporation, Hiroshima City University

[18] 2017 Conference on Neural Information Processing Systems *https://nips.cc/*, Long Beach Convention and Entertainment Center, Long Beach, California, USA

[19] Restaurant information train and development set *http://camdial.org/ mh521/dstc/*, Paul Crook - Microsoft Research, Maxine Eskenazi - Carnegie Mellon University, Milica Gasic - University of Cambridge (3rd February 2014)

[20] Asus N550jk Notebook *http://www.vatanbilgisayar.com/asus-n550jk-core-i7-4700hq-24ghz-16-gb-1tb-156-4gb-w8-notebook-bilgisayar.html*

[21] Ziraat Bank frequently asked questions *https://www.ziraatbank.com.tr/tr/sss*

[22] Tensorflow - An open source machine learning framework for everyone *https://www.tensorflow.org/*

[23] Vector Representations of Words *https://www.tensorflow.org/tutorials/word2vec*

[24] Optimization: Stochastic Gradient Descent *http://ufldl.stanford.edu/tutorial/*

[25] Understanding LSTM Networks *http://colah.github.io/posts/2015-08-Understanding-LSTMs/*

[26] Cosine Similarity *http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/*

[27] Levensthein Distance *http://stackabuse.com/levenshtein-distance-and-text-similarity-in-python/*

[28] Qgram Similarity *http://cs.uef.fi/ zhao/Link/Similarity-strings.html*

Curriculum Vitae

FIRST MEMBER

Name-Surname: Umut GÜNERİ
Birthdate and Place of Birth: 18.10.1995, Çorlu
E-mail: umutguneri.ce@gmail.com
Phone: +90 554 998 04 62
Practical Training: Kafein Technology Solutions

Project System Informations

System and Software: Windows, MAC or Linux Operation System, Android Operation System
Required RAM: 2GB
Required Disk: 512MB