REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CHATBOT

15011022 — Harun UZ
15011027 — Arda Taha CANDAN

SENIOR PROJECT

Advisor
Assoc. Prof. Dr. M. Fatih Amasyalı

November, 2018

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

BLEU	Bilingual Evaluation Understudy
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
ROGUE	Recall-Oriented Understudy for Gisting Evaluation
TRY	Turkish Lira

LIST OF FIGURES

1
Introduction

Chatbots are getting more used day-by-day. And soon, we believe there will be more human-machine relationships than human-human relationships while doing daily things. Even now chatbots can do a lot for you, such as, booking your next flight or a table from a restaurant, tracking your finance situation, telling you the weather, customer support etc.

In this project, we design a chatbot with a specific personality which people can have a conversation in Turkish with. We trained a neural network model with a tv-series' subtitles for a chosen character to make the bot generate sentences that the character would normally say. Using a deep learning technique provides more realistic conversation experience than traditional techniques. Because chatbots which use traditional techniques produce pre-prepared sentences as outputs according to given inputs. And this is a very inefficient way to create a chatbot for our daily chat, since we need to keep all possible answers to all possible questions. But if we you use deep learning techniques, we don't have to keep all possible answers because the chatbot will have a model to produce sentences.

Usually, modern chatbots have network models that process the inputs word-by-word to be compatible with root inflected languages such as English. But the model we use in this project processes the input character-by-character to generate better sentences in an agglutinative language, Turkish.

2
Preliminary

The artificial neural network method used for text generation in this project is quite common nowadays. Previously, traditional methods were used in this area. But nowadays, various artificial neural network models are used with RNN method. The fact that the output of the chat robots using traditional methods is very limited and focuses on a certain subject has pushed us to use the RNN method.

2.0.1 Previous Works

The following projects have been studied and tested for this problem by using deep learning algorithms.

In A Neural Chatbot With Personality [1] project, researchers from Stanford University built a generative chatbot with a personality. Unlike previous researchs, they didn't use any hand-crafted rules, instead, they trained a seq-2-seq model with Cornell movie dialogue dataset and 4 different tv series' scripts. They chose a specific character from each tv-series and trained model with his/her dialogues again to make the bot imitate the characters. Here are their 3 training phases:

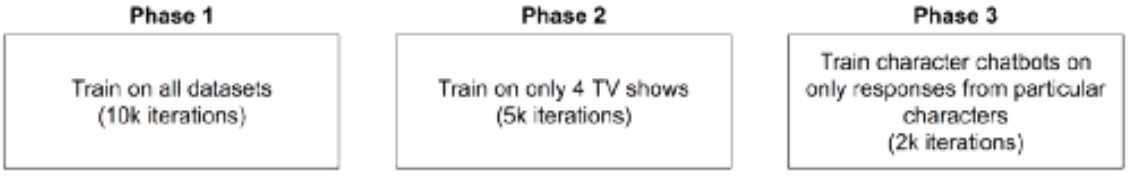

Figure 2.1 A Neural Chatbot With Personality training phases [1]

They used stacked GRU cells of 3 layers as RNNs in their model. To test their model, they used both automatic metrics and human judgment. For automatic metrics, they used BLEU and ROGUE metrics and the results can be seen in Figure-2.2:

The scores are very low, but it was expected since the model is supposed to stick to its character and its answers may seem to be wrong according to automatic evaluation although it says what the character would normally say.

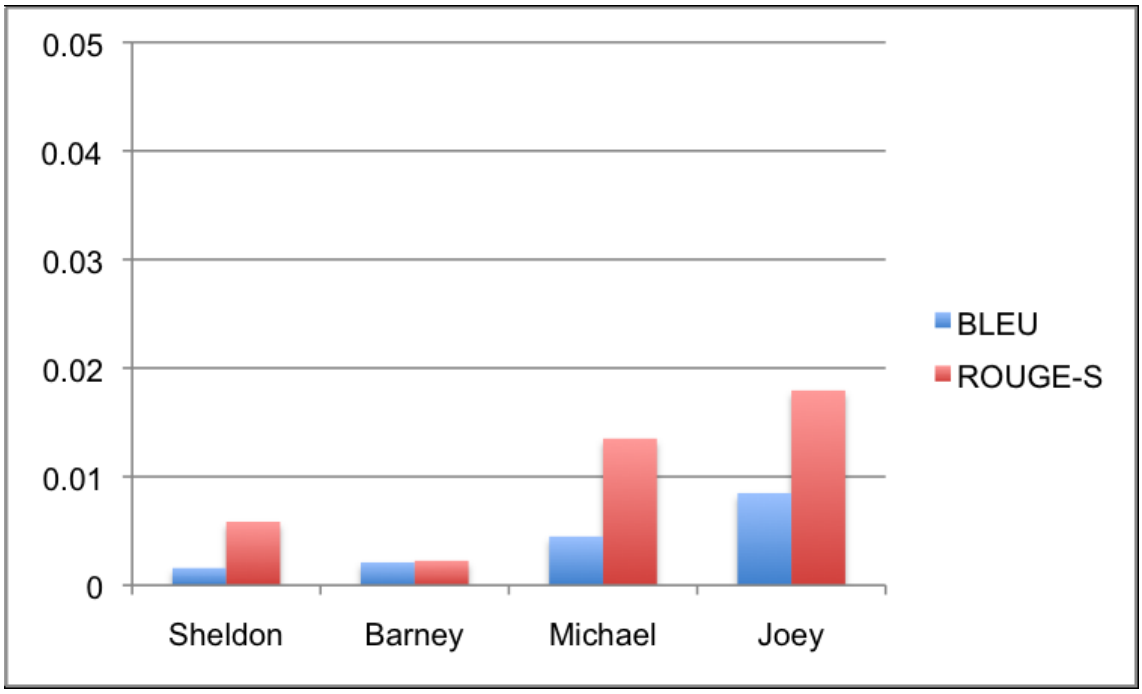

Figure 2.2 A Neural Chatbot With Personality results for automatic evaluation metrics, BLEU and ROGUE [1]

In Dialogue Modeling Using Recurrent Neural Networks [2] project, researchers have developed a chat robot using the sequence to sequence method. They have taken a model with a LSTM structure that has been trained with the words in the dictionary to a certain extent and built their own models on this model. They then trained this model with the OpenSubtitles data set, which includes thousands of movies and subtitles. For the testing they chose 1 real human, 2 chatbots which produced in other projects and the chatbot produced in this project. The participants chatted with someone randomly from this quartet without knowing who the other person was. Here is the example conversation between a participant and chatbot:

user: hi!
chatbot: hello.
user: how are you?
chatbot: you've been so much
user: i've been so much?
chatbot: i'm not going to do that, i mean, you're a fool.
user: i am not a fool!
chatbot: it's a little thing.
user: ok.
chatbot: i don't know.

Figure 2.3 Example conversation between a participant and chatbot [2]

Each participant was given a questionnaire to evaluate his / her responses after the interview was terminated. The results were not as expected. Even though chatbot has established meaningful and meaningful sentences, they were independent of the

content and the speech history. Here is the result of questionnaires:

Statement	Alice	Cleverbot	Human	First model	Second model
The conversation between myself and the Chatbot felt natural	-0.6	0.2	1.3	-1.7	-1.0
The Chatbot's responses were always appropriate	-1.0	-0.8	1.3	-1.4	-1.1
The Chatbot demonstrated emotion at times	0.6	0.6	0.0	-0.7	-0.2
The conversation was coherent	-0.1	-0.2	1.3	-1.3	-1.4
I found the Chatbot entertaining	0.9	1.3	0.8	0.0	-0.3

Figure 2.4 Questionnaire results. The score ranges between -2 and 2. [2]

3
Feasibility

The aim of this project is to create a chatbot using artificial neural networks. But processing artificial neural networks is a work that requires high hardware power. Since these kind of hardwares are expensive in our country our limited resources have led us to use open source and free licensed software.

3.1 Technical Feasibility

We explained which software and hardware tools we used in this project below.

3.1.1 Software Feasibility

We used TensorFlow, an open source machine learning framework developed by Google, in Python since there are plenty of documents and examples both for Python and TensorFlow. We trained and tested our model on a free cloud platform provided by Google due to its compatibility with Python and GitHub.

3.1.2 Hardware Feasibility

Two personal computers were used during both research and design stages.

First computer's specifications:

- 16,0 GB RAM
- Intel Core i7-5700HQ CPU
- CPU Clock Rate: 2.70GHz
- Graphics Card: NVIDIA GTX 960M 2 GB
- Operating System : Windows 10, 64 Bit

Second computer's specifications:

- 8,0 GB RAM
- Intel Core i5-5200U CPU
- CPU Clock Rate: 2.20GHz
- Graphics Card: NVIDIA Geforce 830M 2 GB
- Operating System : Windows 10, 64 Bit

Since we needed high computation power during implementation and test phases, we used 24GB GDDR5 NVIDIA Tesla K80 GPU on a Linux server provided by Google Colaboratory.

3.2 Schedule Feasibility

In this project we chose waterfall development model and tried to distribute the work force equally. The planning of the project from the beginning to the delivery stage is given in the Figure-3.1.

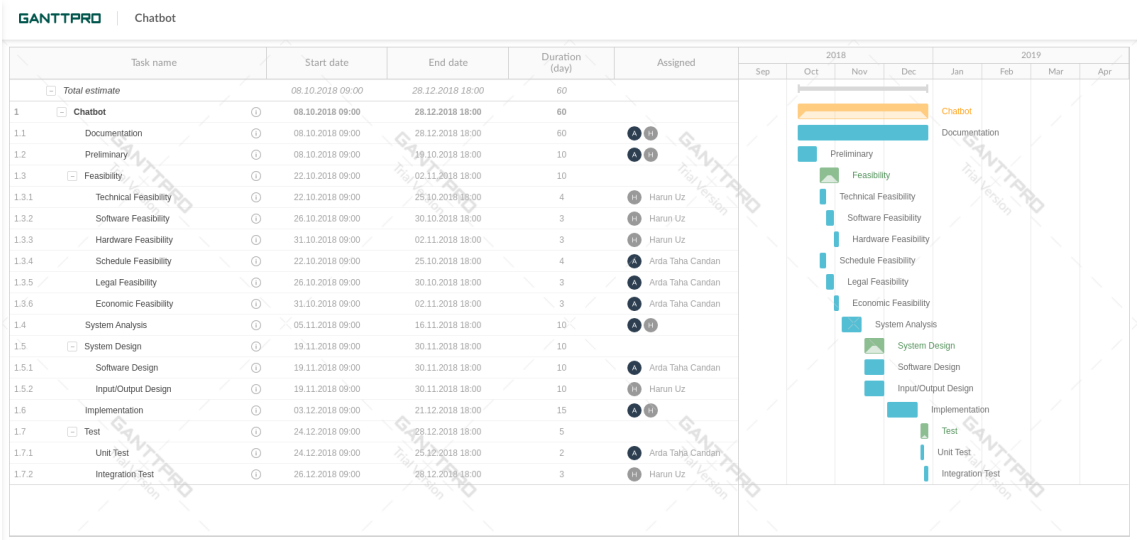

Figure 3.1 Gantt diagram

3.3 Legal Feasibility

In this project we used Tensorflow deep learning library and Google Colab to train the model on cloud which are both open source. The model was trained with free to use subtitles.

3.4 Economic Feasibility

Two computers for each computer engineer were used. All other tools and softwares have free license. It is planned to complete this project in a month.

- 2 * average laptop which has Windows 10 license = 2 * 4.000 TRY
- 2 * computer engineer who has an average monthly salary = 2 * 6.000 TRY

In total, this project approximately costs 20.000 TRY.

4
System Analysis

In recent years, using neural networks to build chatbots have been quite popular. Such deep learning techniques save developers from using hand-crafted rules to design chatbots as they provide better solution to the realism problem. Rule-based chatbots give answers which are stored in an intention-response pool to utterances according to the intention of input utterance. Although this might be useful for particular purpose chatbots like a booking assistant, it doesn't give life-like experience for a casual conversation.

LSTMs-which we used in this project- are special kinds of RNNs. They were developed to learn long-term dependencies, that's why they are really common in language processing. Because, when humans chat with each other, usually the sentences have a connection with previous utterances and a model capable of learning those connections makes it possible to have a realistic conversation with a machine. LSTMs have loops between its layers, and while the current input goes through whole model, LSTM can add or remove some of its information by gates, the connection point between current and previous inputs.

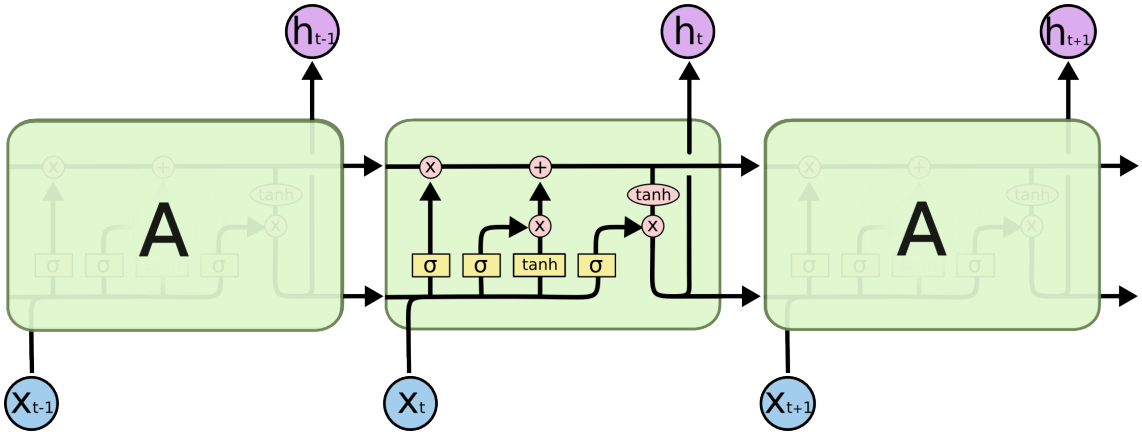

Figure 4.1 The repeating model in an LSTM

In this project, we designed a chatbot that has a personality of a character we chose. We aimed to create a chatbot that can have a random, realistic conversation with the

user just like how the chosen character would talk. The language of the conversation will be Turkish, hence we created the dataset in Turkish.

4.1 Dataset

We used tv-series dialogues to create a dataset which includes the scripts of The Therapist and The Mentalist. We prepared one season from each scripts by labeling and organizing the dialogues. There are about 9.000 utterances in one season of The Therapist and 16.000 in a season of The Mentalist, which makes it about 25.000 utterances,in total. We used 23.000 for training and the rest for test phases.

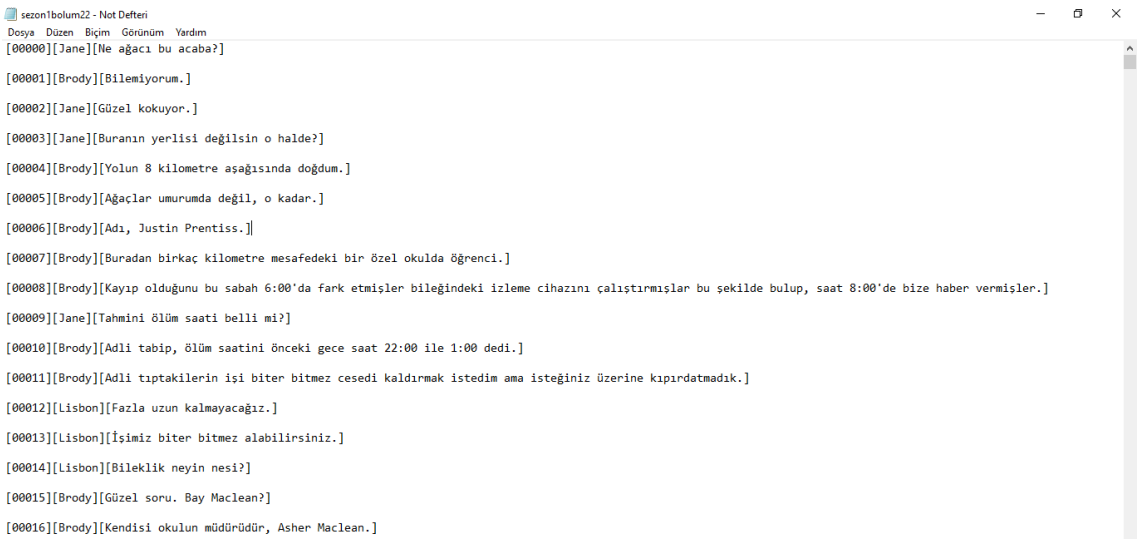

Figure 4.2 An example from dataset

4.2 Training

To train our model in Turkish which is an agglutinative language, we modified the input pipeline of the model to get the string character-by-character. Because, usually, chatbot models are trained to serve in English which is a root inflected language and using the same technique for Turkish causes some problems. The model struggles to produce words. That's why we decided to give the input character-by-character and tested our model by training it with just one episode of The Mentalist's script in 7600 training steps and 100 epoch size. The result which can be seen in Figure-4.3 seems promising due to the words that the model generated are similar to Turkish words.

[00183][Lisbon][Kurbanlarla bağınız nisir sir burdan kondanta sulitdar malırlarızı de karır mem geliran mamır dek oçlaromaz oça dek garlen böy karlın?]

[00235][Jane][Pürnen bar küraya surar bir yolunı alansın.]

[00187][Jane][Sere karın be serin zapdar.]

[00040][Jane][Ke yamın değin isar.]

[00697][Jane][Selin kerlan adün demiyayır.]

[00541][Jane][Buyır.]

[00227][Jane][Seyen ter girin ilik ger daha.]

[00130][Part][Ben garır giri gir ker kuluyar galara sanılında özamar meyin kerseni so yerin barlığurı varlunaz.]

[00226][Jane][Evat gerim.]

[00193][Jane][Benicedir ilk toh ki otme sirmik .]

Figure 4.3 Output example-1

References

[1] H. Nguyen, D. Morales, and T. Chin, "A neural chatbot with personality," [Online]. Available: https://web.stanford.edu/class/cs224n/reports/2761115.pdf.

[2] A. Viktor and O. Jonathan, "Dialogue modeling using recurrent neural networks," 2016. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/237861/237861.pdf.