# AUTOMATED RESPONSE GENERATION PROJECT

## Importance and potential impact

**Problem:** People are overwhelmed with daily routine tend to make similar quick responses in their messengers and lose valuable time throughout the day. They seldom happen to text specific responses so another reply should take no time at all.

**Potential Impact:** We expect people to save time in case of using the automated response system for a private messaging.
Example:
**Case:** Faculty Director at the university has many organizational responsibilities. Students often contact him in case of clarifying schedule of exams, places, etc.

- Are we having an Cybersecurity Exam today in room 303 at 10 a.m.?
- Yes, you are

Such kind of questions could happen to appear multiple times per day. And every time he types letter by letter to respond to all of the questions. With our solution he will save time responding to the typical questions and each response will be customized according to his messaging style in case of open-ended response.

## Datasets and our explorations

The project we are conducting is concerned about Automated Response Proposal generation in messenger programs on your smartphone. Consequently, we wanted to find a dataset of private texting. Sadly major, messenger services, due to security legislation, create too many obstacles to gather such kind of data from storage. After several explorations there were some selected:

| Dataset | Description | Example |
|---|---|---|
| Amazon product data | **Contains:** ['answer', 'answerTime', 'answerType', 'asin', 'question', 'questionType', 'unixTime'], **Amount of answers:** around 142.8 millions (May 1996 - July 2014). | **Question:** can these be used for gas stove burner? **Question Type:** yes/no **Answer:** yes it can.i use it on my grills side burner to slow boil leftover marinades and veggies. **AnswerType:** Y |
| Harry Potter books | **Contains:** all text from 7 Harry Potter novels | **1) Direct Speech:** - Як його звати? Говард, чи що? |

| | | |
|---|---|---|
| | **Amount dialog lines (found):** 24240<br><br>Used to check if the question identification and question type prediction work correctly;<br><br>Doesn’t have any labels. | - Гаррі, якщо тобі так уже кортить. Паскудне й банальне ім'я. |
| **Subtitles** | Gathered from movies and series to check if the question identification and question type prediction work correctly;<br><br>**Amount:** is not bounded;<br><br>Doesn’t have any labels. | **Grey’s Anatomy**<br>63<br>00:02:24,356 --> 00:02:26,925<br>who's doing a similar kind of ultrasound research.<br>64<br>00:02:26,959 --> 00:02:29,260<br>Uh... she hasn't published much yet, |

The “Amazon product data” publication made us take a closer look at the datasets they propose freely.

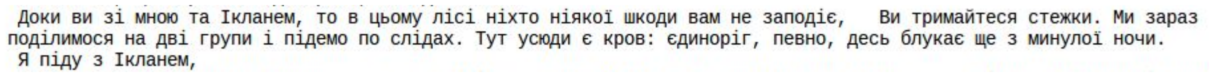
So we took the two of these datasets: “qa_Appliances” and “qa_Tools_and_Home_Improvement”. These are number of conversations about some exact product offers on Amazon. There are real people and, also, Amazon consultants helping each other, giving advice. The data in the datasets is divided by question/answer by type (yes/no question or not). So with no effort, we have a labeled data ready to be investigated. As a first step, we decided to investigate if we could predict whether the message texted belongs to Yes/No or to open-ended. We tried to train the model with 9011 elements and to predict 101088 elements of data. The result mentioned below proved that this dataset could be taken for first-step investigations for message automation process.

| **Method** | **Accuracy** | **Mean squared error** | **Variance score** | **Precision** | **Recall** | **F1 score** |
|---|---|---|---|---|---|---|
| Linear Regression | 0.32 | 0.17 | 0.32 | 0.86 | 0.83 | 0.84 |
| Logistic Regression | 0.50 | 0.12 | 0.50 | 0.88 | 0.88 | 0.88 |

Regardless low accuracy we could state that these datasets let us understand how to build a robust automated response model.

Concerning Harry Potter novels, it should be stated that regardless more than 24 thousand direct speeches discovered, the dataset generated is pretty messy. And It does not correspond that much to the idea of our project: to generate user adopted automated response, because it is unclear which of the direct speeches should be reacted with a reply, and what kind of it should be proposed.

An example from Harry Potter dialog dataset:



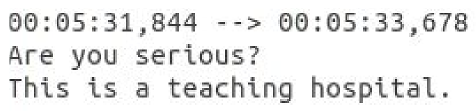
```
 Доки ви зі мною та Ікланем, то в цьому лісі ніхто ніякої шкоди вам не заподіє,   Ви тримайтеся стежки. Ми зараз
поділимося на дві групи і підемо по слідах. Тут усюди є кров: єдиноріг, певно, десь блукає ще з минулої ночи.
 Я піду з Ікланем,
```

- **Incoming message:** “Доки ви зі мною та Ікланем, то в цьому лісі ніхто ніякої шкоди вам не заподіє,   Ви тримайтеся стежки. Ми зараз поділимося на дві групи і підемо по слідах. Тут усюди є кров: єдиноріг, певно, десь блукає ще з минулої ночи.”
- **Response:** Я піду з Ікланем,

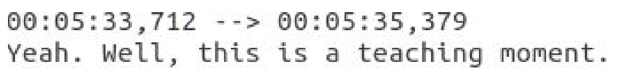
We presume that major amount of this dialogue is out of the scope of topics typically discussed in the real world of messenger apps.

Another attitude of our investigations was to work with movies’ subtitles. It was a better decision in case of consistency with the real world message texting. Its major drawback was that most of the subtitles were complicated to use to find out full direct speeches. There were a lot of subtitles where the text was chopped to correspond to the movie timeline and because of that tears direct speeches in most cases. But the one from Grey’s anatomy appeared pretty nice to work with:

**Incoming message** **Response**



```
00:05:31,844 --> 00:05:33,678
Are you serious?
This is a teaching hospital.
```



```
00:05:33,712 --> 00:05:35,379
Yeah. Well, this is a teaching moment.
```

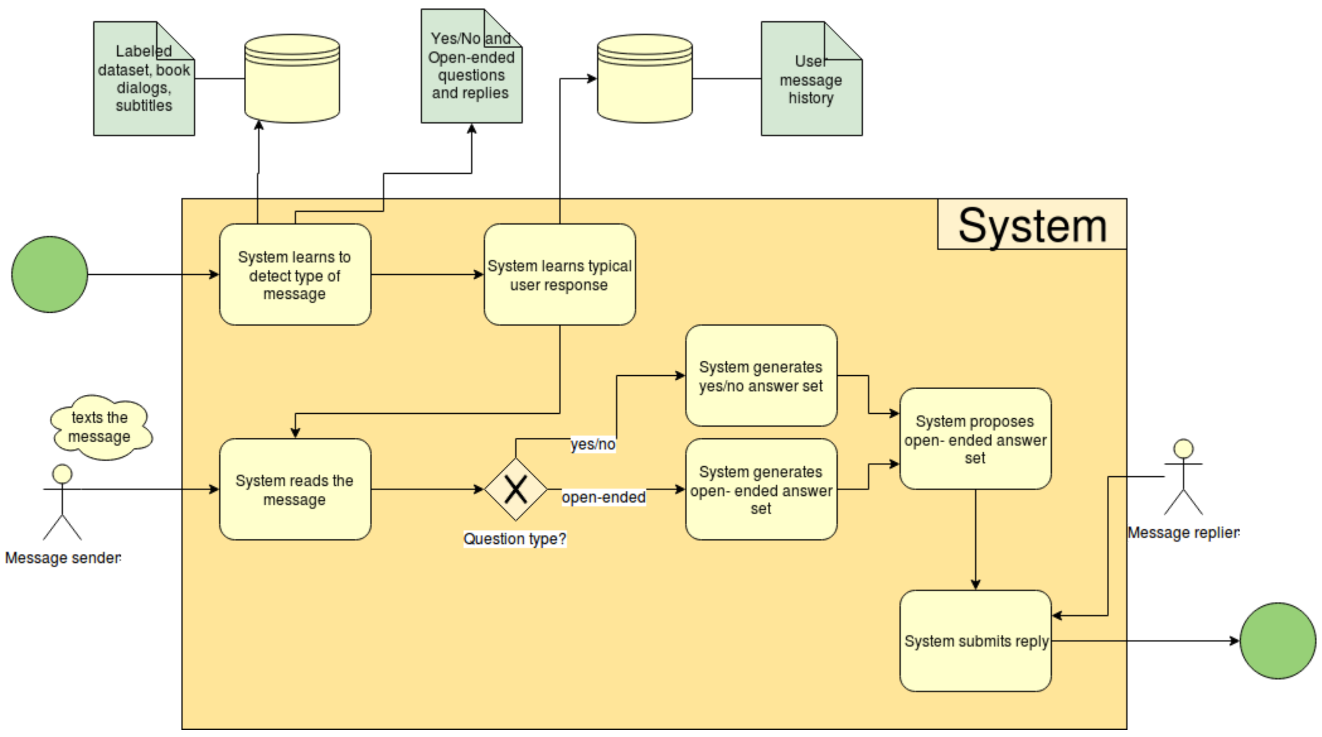
## System Overview

### System (Version 1)

From the very beginning we decided that the right implementation is to develop a system which:

1. could detect the type of message user receives, naimly “yes/no” or “open-ended”
2. learn the history of user replies style and words usage
3. generates the user-customized automated response

## General Architecture


Labeled dataset, book dialogs, subtitles
Yes/No and Open-ended questions and replies
User message history
System
System learns to detect type of message
System learns typical user response
texts the message
Message sender
System reads the message
Question type?
yes/no
open-ended
System generates yes/no answer set
System generates open- ended answer set
System proposes open- ended answer set
Message replier
System submits reply

We decided that the best service to use for our system is Telegram.

Here are the general steps of how the system should have worked:

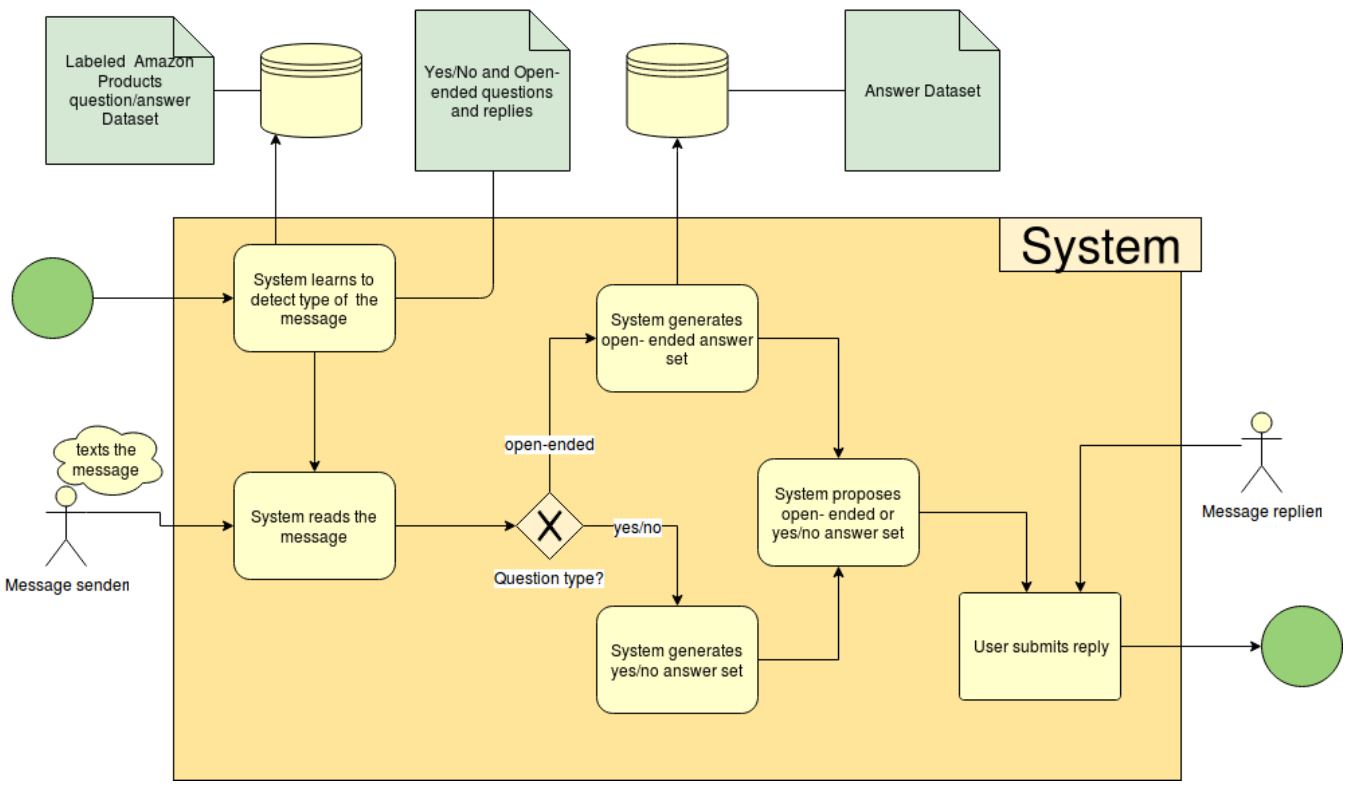
1. System learns on labeled data from Amazon how to define yes/no and open-questioned questions and replies.
2. System learns the variety of questions and common answer types to them from books, subtitles, and other sources.
3. System learns the typical replies of its user by accessing his/her message history.
4. System reads the message from the Telegram.
5. System identifies a type of the question.
6. Relying on top priority on user historical replies sourced by other reply info ( books, subtitles, etc.) produces the set of possible responses.
7. Chosen by the user alternative is sent as a reply to the incoming message.

## System (Version 2)

Due to difficult project flow we decided to make an MVP of our system. There were several ideas how to shorten the scope of the work. So it was decided that more effort should be put on the classification of the type of message and create a simple prototype for making predictions for messages.

The system learns a data set and, basing on data, generates predicted world for message. Predictions are made with Markov chain, so the system depends only on the current word. It is obviously, that with different datasets predictions will be different and the bigger dataset is the better it is for the model's predictions.

**General Architecture**


Labeled Amazon Products question/answer Dataset
Yes/No and Open-ended questions and replies
Answer Dataset
System
System learns to detect type of the message
System reads the message
texts the message
Message sender
open-ended
yes/no
Question type?
System generates open- ended answer set
System generates yes/no answer set
System proposes open- ended or yes/no answer set
User submits reply
Message replier

## Modelling

From the very beginning it was decided to move with DocToVec to create a low dimensional representation of each of the questions. We believed that converting to vectors could allow us to reach better results in predictions. So every question will have its words converted to vectors and such sequence of vectors will be assigned to a specific value. Here are the results achieved on 10% of the whole question dataset: (question type in our case).

| **Logistic Regression** | **Accuracy** | **Mean squared error** | **Variance score** |
|---|---|---|---|
| Without Doc2Vec | 0.50 | 0.12 | 0.50 |
| With Doc2Vec | 0.62195 | 0.38 | -0.52 |

So this attitude was justified and we believe should work better with the CNN.

The CNN is implemented but not yet tested due to problems of GPU access. Under such circumstances we were forced to check our system with only linear regression working but with all the dataset converted through DocToVec.

## Evaluation

- Datasets are thoroughly investigated.
- Predictions on yes/no and open-ended questions improved.
- CNN is designed but not trained
- The Demo messenger created

By now the system is not that efficient that was expected. There should be more effort put on designing LSTM network and discoveries how to beat the Google whitepaper on Gmail automated response. But we are pretty clear by now that CNN trained on dataset converted using DocToVec should bring us accuracy of about 65-80% based on whitepapers of the similar projects.

**Authors:**

Ostap Kharysh

Olha Bakay

## Sources Used:

**Dataset:** http://jmcauley.ucsd.edu/data/amazon/

**Academic Sources:**

1. https://medium.com/ymedialabs-innovation/next-word-prediction-using-markov-model-570fc0475f96
2. http://www.aclweb.org/anthology/D13-1160
3. http://jmcauley.ucsd.edu/data/amazon/
4. https://arxiv.org/abs/1408.5882