

Chatbot

What is a Chatbot?

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. We will focus on the written aspect only.

Project: Phase 2 Proposal CS476

Members

1. Fouad Majd Alkadri | 218110075
2. Abdullah Rajoub | 218110141
3. Abdulaziz Alowain | 219110119
4. Ibrahim Khurfan | 218110082
5. Hammad Ismaeel | 219211202

Date	Version	Authors	Notes
3/14/2022	Version 1.0	<ul style="list-style-type: none"> Fouad Majd Alkadri Abdullah Rajoub Abdulaziz Alowain Ibrahim Khurfan Hammad Ismaeel 	<ul style="list-style-type: none"> We might try Arabic (Bilingual chatbots). In (problem section) give more example to make sure everything is clear. The (Dataset section) should be clarified more. In (scope of the project section) clarify give more examples. Advice: Considering the chatbot that we're developing is integrated with WhatsApp. In (Measure & error analysis section) it should be clarified which dataset used. In (Measure section) define who will be the users?
4/23/2022	Version 2.0	<ul style="list-style-type: none"> Fouad Majd Alkadri Abdullah Rajoub Abdulaziz Alowain Ibrahim Khurfan Hammad Ismaeel 	<ul style="list-style-type: none"> Fixing the last version (1.0)

Contents

1. The Problem	4
2. The Dataset available.....	5
3. Proposed Approach and Tools to be Used	5
3.1 Creating The NLU	5
3.2 Processing pipelines:.....	6
4. Scope of the Project	7
4.1 Features that are included in the project.....	7
4.2 Features not included in the project.....	7
4.3 Interface Used.....	7
5. Work Plan.....	8
6. Related Work	9
6.1 Conclusion	10
7. Evaluate the Measures and Error Analysis Methods	11
7.1 Experiment	11
7.2 Measures	12
7.3 Error Analysis Methods	14
8. Group Management.....	16

1. The Problem

Problems we are trying to solve are that customers waste a lot of time waiting in line of a coffee shop to make an order and need to waste money in calling customer service to ask questions or to give feedback, so in order to make it easier and less expensive for the customers to do all of these things, and make them do it with a click of a button we are going to make a virtual agent (chatbot) for the coffee shop where the chatbot can do the following tasks:

❖ 1.1 Taking orders from customer

- Customer will initiate a conversation with the chatbot
- Chatbot detect the order intent.
 - If the customer has provided what they want to order already chatbot will add it to the cart and ask the customer if they want anything else or if they order or not.
 - if they do not provide the items they want to order or the chatbot is unable to understand what the customer wants, it will display the menu and make them order.

❖ 1.2 Take questions from customers

- ❖ The customer will initiate a conversation with the chatbot
- ❖ Chatbot detect the question intent
 - the chatbot will check the question if it is already stored in the database with its answer, and provide the answer associated with it
 - If the question was not stored it will initiate a live chat session between the customer and customer service agent to answer the customer's question

❖ 1.3 Take Complaints from customers

- The customer will initiate a conversation with the chatbot
- Chatbot detect the complaint intent
 - the chatbot will check the complaints if it is already stored in the database with its steps to deal with complaints
 - Some examples of some frequent asked complaints are the following:
 - . dirty place: ask to send a picture + location
 - . late order: ask for order number
 - The chatbot will store the customer complaint and send it to the customer service team to review it and solve the customer problem.

❖ 1.4 What is excluded

- ❖ The chatbot will not provide audio or video calls.
- ❖ The chatbot will not support any language other than **English and Arabic**.
- ❖ The chatbot will not understand any items that are not listed in the coffee shop menu.

2. The Dataset available

For our chatbot, we are going to use external existing coffee shop datasets to shorten the development time and increase the accuracy of our chatbot, most of our datasets will be taken from Kaggle website, and we are going to combine these datasets together and modify some of them to fit the coffee shop menu items. Finally, the datasets will be in English and Arabic.

3. Proposed Approach and Tools to be Used

The tool that will be used is Rasa. Rasa is an open-source machine learning framework for building AI assistants and chatbots. Mostly you don't need any programming language experience to work in Rasa. Rasa can be used to build contextual AI chat bots, meaning that the chat bot will be capable of providing responses that fits the context. For example, if I'm ordering a coffee, and the chatbot wants to confirm my order, the chatbot should response with "do you want to confirm your coffee purchase? "

Rasa consist of the following components:

NLU: it is the part of chatbot used for entity identification and intent classification. It enables your chatbot to understand what is being said. It takes the input in unstructured human language form, and extract entries and intents.

Core: It is also referred to as a dialog management component. It is the part of the chatbot that is concerned with decision making. How should I response to a specific input?

We will take a deeper look at the approach used for each of these two components:

3.1 Creating The NLU

In this step, we must do conversation design. Conversation design includes:

- ❖ Identifying your target users.
- ❖ Understanding what they will use you assistants for.
- ❖ Crafting the most typical conversation they will have with your assistant.

Our target users are coffee shop customers. They will be using our assistant to do the following:

- Place a new order.
- Make complaints.
- Ask for available menu items
- Ask for available offers

To craft a typical conversation, we used our experience within person customer service and how a typical conversation between us (customers) and a coffee shop employee goes on. Somethings to consider while crafting the typical conversations are:

- We will not include all variations of responses in NLU training data.
- Only a sample example 5-15 per intent, for harder cases like complaints we are going to use 20+ examples to make sure that we get the right intent.
- The example given should have a single intent (e.g., want can have purchase as intent and complain as intent).
- The examples given under intent should be diverse in vocabulary and grammatical structure.

3.2 Processing pipelines:

A processing pipeline is a sequence of processing steps that extracts text features that allows the module to learn the underline pattern from the provided example. At the beginning, we are going to use one of the pre-configured pipelines. This pre-configured pipeline is called `pretrained_embedding_spacy`. This library represents each word as a vector of values. These vectors of words are used to compare how closely two words are similar to each other in meaning (semantic) and grammar (syntactic).



Figure 1



As shown in the Figure 1, the word cheeseburger and hamburger are closer to each other (their vectors are closer). The word Ferrari is far away from both.

Some advantages of using `pretrained_embedding_spacy`:

- Faster training and iteration
- Less training data required to achieve good model performance

It is worth noting this model can handle intent classification and entity extraction too.

NOTE: The approach provided is not comprehensive and will probably change throughout the project. This is just the initial approach that we have in mind.

4. Scope of the Project

4.1 Features that are included in the project

- ❖ Chatbot will be able to provide offers available by providing the link to the coffee shop website.
- ❖ The chatbot will be able to provide a list of items on command.
- ❖ The chatbot will be able to take orders from customers.
 - Order: I want to order coffee: Response: what type of coffee do you want?
 - Americano: Ok, nice choice, but what size?
 - Large: Ok, noted sir. Anything else?
- ❖ The chatbot will be able to take complaints.
- ❖ The chatbot will be able to respond to greetings.
- ❖ The chatbot will provide helpful contact information on command (e.g., support team email)
- ❖ The chatbot will be able to provide an exact list of the available offers, and the list will be static.

4.2 Features not included in the project

- ❖ The chatbot will not be able to respond to any non-English and non-Arabic input.
- ❖ No speech recognition feature for voice input.

4.3 Interface Used

We will use **Twilio** which provides all the needed services for Rasa customer services with chatbot capabilities. **Twilio** takes the chatbot model given and connects it with Rasa.

5. Work Plan

Week 1-8

Pick NLP application (Chatbot)

Pick NLP topic (Coffee Shop Chatbot)

Week 9

Create Proposal

Create a simple Rasa Chatbot

Search for datasets

Week 11

Create stories and appropriate actions for our chatbot

Week 10

Implement the datasets to our chatbot

Week 12

Create stories and appropriate actions for our chatbot

Week 13

Create stories and appropriate actions for our chatbot

Week 14

Refine the Machine Learning Components

Week 15

Test the chatbot and add final touches

6. Related Work

In this part of our proposal, we tried Rasa's starter pack Retail chatbot. The inputs we tried in this chatbot are:

1. Hi using unconventional ways.

Input: "Greetings!"

Output: The bot asked to try again because it did not understand.

```

Your input -> Greetings!
Ok, let's start over.
Buttons:
1: Check status of my order (Check status of my order)
2: Start a return (Start a return)
3: Check inventory (Check inventory)
4: Subscribe to product updates (Subscribe to product updates)
Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you?
beep, boop, don't understand
  
```

2. A normal Hi.

Input: "Hi"

Output: The bot greets the user and tells him to pick a service from the list.

```

Your input -> hi
? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? (Use arrow keys)
> 1: Check status of my order (Check status of my order)
  2: Start a return (Start a return)
  3: Check inventory (Check inventory)
  4: Subscribe to product updates (Subscribe to product updates)
Type out your own message...
  
```

3. We picked the 4th service "Subscribe to product updates" and entered an invalid email (no @).

Input: "fjffj-.com"

Output: The bot asks the user to start over.

```

? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? 4: Subscribe to product updates (Subscribe to product updates)
I can keep you up to date with our weekly email about product updates! If you'd like to be added to the list, please add your email address.
Your input -> fjffj-.com
Ok, let's start over.
? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? (Use arrow keys)
> 1: Check status of my order (Check status of my order)
  2: Start a return (Start a return)
  3: Check inventory (Check inventory)
  4: Subscribe to product updates (Subscribe to product updates)
Type out your own message...
  
```

4. We picked the 4th service "Subscribe to product updates" and entered a valid email but an unknown domain (@mail).

Input: "aziz@mail.com"

Output: The bot asks the user to start over.

```

? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? 4: Subscribe to product updates (Subscribe to product updates)
I can keep you up to date with our weekly email about product updates! If you'd like to be added to the list, please add your email address.
Your input -> aziz@mail.com
Ok, let's start over.
Buttons:
1: Check status of my order (Check status of my order)
2: Start a return (Start a return)
3: Check inventory (Check inventory)
4: Subscribe to product updates (Subscribe to product updates)
Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you?
um, what did you just say to me?
Your input ->
  
```

- We picked the 4th service “Subscribe to product updates” and entered a valid email domain.

Input: “aziz@gmail.com”

Output: The bot asks the user to start over. Which means that there is a bug with the bot.

```

? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? 4: Subscribe to product updates (Subscribe to product updates)
I can keep you up to date with our weekly email about product updates! If you'd like to be added to the list, please add your email address.
Your input -> aziz@gmail.com
Ok, let's start over.
? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? (Use arrow keys)
» 1: Check status of my order (Check status of my order)
   2: Start a return (Start a return)
   3: Check inventory (Check inventory)
   4: Subscribe to product updates (Subscribe to product updates)
Type out your own message...
  
```

- We tried double intents with spelling mistakes (produce not product).

Input: “hi I would like to subscribe to produce updates”

Output: The bot subscribes the user to the service and does not greet the user.

```

Ok, let's start over.
Buttons:
1: Check status of my order (Check status of my order)
2: Start a return (Start a return)
3: Check inventory (Check inventory)
4: Subscribe to product updates (Subscribe to product updates)
Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you?
um, what did you just say to me?
Your input -> hi I would like to subscribe to produce updates
Alright, you're subscribed!
  
```

- We tried greetings with spelling mistakes

Input: “helo”

Output: The bot can detect the spelling mistake and greet the user back.

```

Your input -> helo
? Hi there! I'm a demo bot from Rasa to help you with ordering shoes. How can I help you? (Use arrow keys)
» 1: Check status of my order (Check status of my order)
   2: Start a return (Start a return)
   3: Check inventory (Check inventory)
   4: Subscribe to product updates (Subscribe to product updates)
Type out your own message...
  
```

6.1 Conclusion

The bot was able to detect spelling mistakes and deal with double intents correctly. There are however some issues with the services of the bot. For example, even if we provide a correct email for the subscription service the bot still asks the user to try again.

7. Evaluate the Measures and Error Analysis Methods

7.1 Experiment

The experiment that we're going to conduct is about the cafe shop chatbot that will include starter pack, so the steps that we're going through that we implement is the following:

- ❖ Build the chatbot project through the command line interface.
 - Command Line Anaconda.
- ❖ Installing **Rasa** inside the virtual environment that we're going to use.
 - Python 3.7
- ❖ Adding the pipelines or the components that's going to be trained as a models in **config.yml** which are the following:
 - Machine learning Component.
 - Neural Network Component.
 - RexgetFeaturizer Component.
 - LexicalSyntaticFeaturizer Component.
 - CountVectorFeaturizer Component.
 - WhitespaceTokenizer Component.
- ❖ Implement our datasets and put the examples of intents and the actions through **data/nlu.md**.
 - Sample example:

EX 1:

##Intent:greet

-hello

-hi

-yo

-wassapp

-hya

EX 2: [--] search for that pokemon, (--) entity.

##Intent:Find Pokemon

-is [bulbasaur] (pokemon-name) a pokemon

##Lookup:pokemon-name

-data/pokemone.txt

- ❖ Ensure that the **Domain.yml** file covers the domain of **data/nlu.md** file that is responsible for the action of the chatbot if it is either available or not.
- ❖ Implement the stories and the sequence of action examples in the **data/stories.md** file that is responsible for the flow of story that's created by the user when they're experimenting with the chatbot.
- ❖ Training **Rasa** through this dataset by using the pipelines or components that we've provided.
- ❖ Experimenting the chatbot that has been built and trained.

7.2 Measures

The measures that we're going to use throughout the project to evaluate that the chatbot works well are for the following reasons:

- ❖ User feedback.
 - By simply asking real coffee shop customers, if they're satisfied with the chatbot or not.
- ❖ Classification Evaluation Metrics by measuring the.
 - Confusion matrix is useful to see what classes got confused with each other.
 - Binary confusion matrix (positive, negative)

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

- ❖ **True Positives:** positive examples that the classifier predicts, they are positive.
- ❖ **True Negative:** negative examples that the classifier predicts are negative.
- ❖ **False Positives:** negative examples that the classifier predicts, they are positive.
 - Type I Error.
- ❖ **False Negative:** positive examples that the classifier predicts they are negative.
 - Type II error.

- By using the confusion matrix, we can find the following
 - Accuracy:** the percentage of correct answers.

$$Acc = \frac{TP + TF}{TP + TN + FP + FN}$$

- Recall:** ratio of retrieved elements to the relevant ones or we can say that how many positive samples the classifier were able to predict.

$$\text{Recall: } TP/(TP+FN)$$

- Precision:** ratio of relevant elements in the retrieved ones or we can say that how many positive samples are really positive when classifier predicts it is positive.

$$\text{Precision: } TP/(TP+FP)$$

- F-measure:** score those balances recall and precision.
- We use β as a balancer for the weight for precision and recall
 - The more weight or β the more weight for precision and less for recall.
 - The less β or weight the less weight for precision and more for recall.

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (0 \leq \beta \leq +\infty)$$

- F1-measure:** mostly used in confusion matrix.

• F_1 means that $\beta=1$ so we weight recall and precision equally

$$F1 \text{ Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

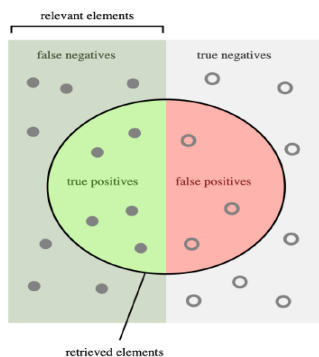
- Micro/Weighted average:** The average considers the number of samples in each class.

┆ Micro: sum of support*f-1 / total sum

○ **Support:** the number of classes

- Macro average:** compute these scores for each class, take average for each score.

macro: sum of f-1 / # classes



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Sample example:

	precision	recall	f1-score	support
Aeroplane ✈️	0.67	0.67	0.67	3
Boat ⚓	0.25	1.00	0.40	1
Car 🚗	1.00	0.50	0.67	6
accuracy			0.60	10
macro avg	0.64	0.72	0.58	10
weighted avg	0.82	0.60	0.64	10

7.3 Error Analysis Methods

In this section, we'll provide scenarios and some examples of what the possible problem are could occur through this. And the possible errors that might disrupt the analysis methods are the following:

❖ Dataset

- Similar classes.
- Not enough features.
- Noisy dataset (High difference in inputs).
- Empty data (missing data → Normalization).

❖ **Note: most of the dataset that we're going to use are preprocessed.**

The solution for that problem is we might be facing are the following:

❖ Dataset

- Normalization the dataset.
- Use Bias and variance analysis to deal with underfitting and overfitting.

❖ Chatbot

- Missing to cover some cases that might the user asks about it, for example the user says (**Hey**) and in the chatbot don't cover this domain and it'll reply to weird stuff.

Sample example:

EX 1:

##Intent:greet

-hello

-hi

-yo

-wassapp

-hya

Compiler:

User: Hey .

Chatbot: Bye.

- ❖ This example shows how the chatbot got confused and didn't know what he could do if there's not possible output for responses.

Solution:

EX 1:

##Intent:greet

-hello

-hi

-yo

-wassapp

-hya

-Hey

Compiler:

User: Hey.

Chatbot: Hi.

- ❖ See the confidence of the model if it works well or not.

```
{
  "intent": {
    "name": "greet",
    "confidence": 0.44583513568867775
  },
  "entities": [
    {
      "value": "neg",
      "confidence": 0.993782948854111,
      "entity": "sentiment",
      "extractor": "sentiment_extractor"
    }
  ],
  "intent_ranking": [
    {
      "name": "greet",
      "confidence": 0.44583513568867775
    },
    {
      "name": "chitchat",
      "confidence": 0.29129539551188588
    },
    {
      "name": "inform",
      "confidence": 0.09576488290307896
    },
    {
      "name": "goodbye",
      "confidence": 0.88987117551991658
    },
    {
      "name": "decline",
      "confidence": 0.88848882616988385
    },
    {
      "name": "affirm",
      "confidence": 0.04842863587816189
    },
    {
      "name": "restaurant",
      "confidence": 0.03121354833799584
    }
  ],
  "text": "Hello stupid bot"
}
```

- ❖ By using the command (**-rasa shell nlu**).
 - That shows how the chatbot are choose or predicting the output with the confidence measure of this model.

8. Group Management

We mainly use discord and WhatsApp for our communications and virtual meetings. Discord is an application like google meet. We do not have specific roles or responsibilities for each member. We instead work on discord where one member would present his screen and code and we would give him ideas. When we feel the work should be distributed as tasks, we try to give each member the tasks he wants. We use google drive to store our documents and code. The frequency of our meetings depends on the amount of work to do. For example, when we made the proposal, we met twice and distributed the work and then met again to finalize it.