

A chatbot assistant

implemented using NLP and ML by

Băilă Romina & Cocoş Tudor

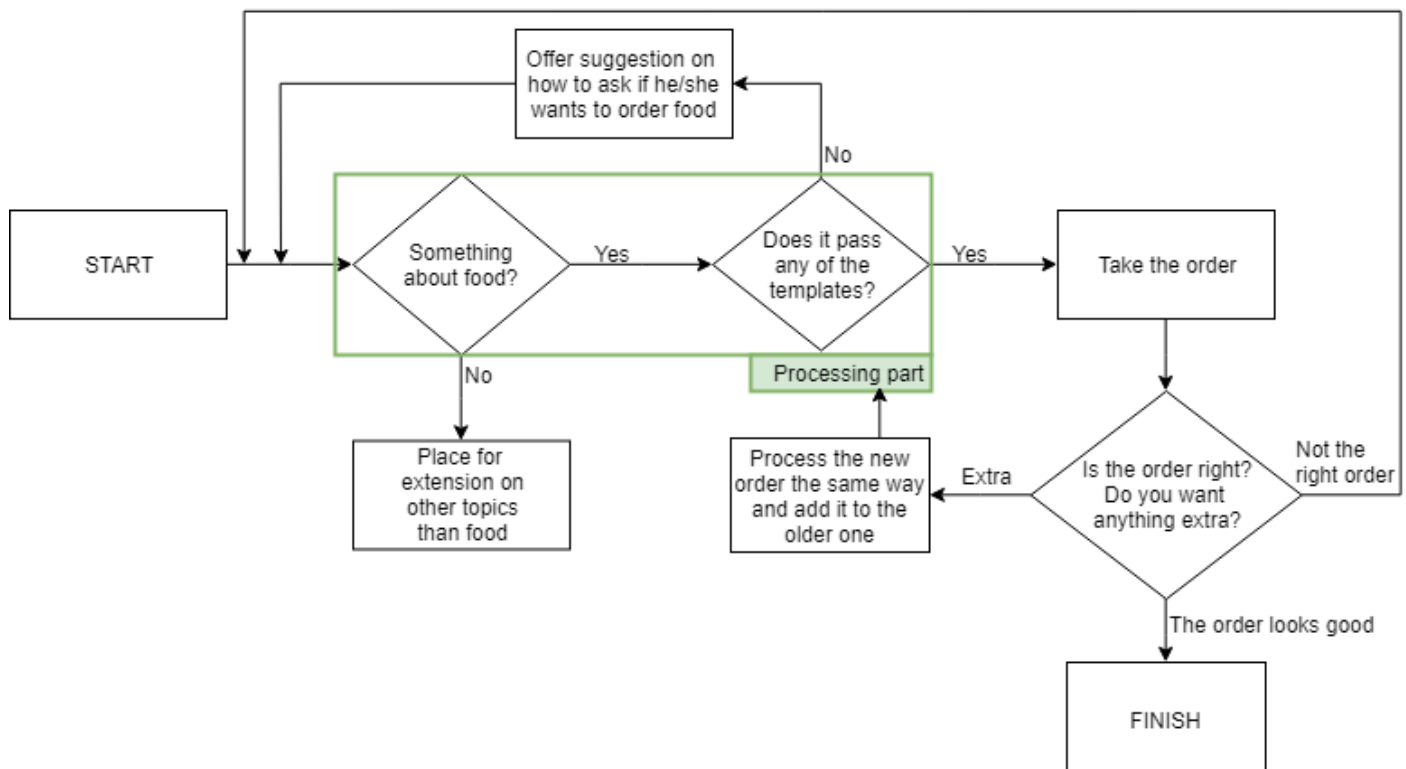
Version	Details	Date
1.0	First report	27.03.2019
1.1	Second report + basic chatbot	09.04.2019
2.0	Third report + use of semantic role labelling	25.04.2019
2.1	Fourth report + use of dependency parsing	24.05.2019

Proposed idea

We would like to develop a chatbot that would act like a personal assistant while it resides in a private chat room with yourself in an application like Slack or MS Teams.

The main objective of the project is to detect the intent of the user by analysing the text input from the chat. Context is also important, and will be analysed first, as to apply the correct intent to the right context.

Analysis & design



For a start, we thought of trying to process food orders for lunch breaks. To do so, we need to find the context first: “Does the request have anything to do with food?” – If yes, then we go through a series of filtering steps that involve syntax and parts of speech, to find out what the user wants to order. The current implementation takes into account only the intent of ordering food when it comes to this context. Other types of requests will redirect the user to order some food.

Development

To start, we will try to implement a basic functionality for the chatbot: ordering food for the lunch break. It will take as input a text that contains a name of a restaurant or fast-food joint, what to order and in what quantity and maybe even some details about the order.

Once we manage to achieve a working prototype of this functionality, we can start to diversify what the chatbot can do, especially on the business side of things (reports, appointments, to-do agenda, etc).

Technologies used

The chatbot will be implemented in Python, as far as programming languages go, and we will make use of the AllenNLP and NLTK libraries, which contain vital functions in order to process natural language.

Implementation details

Using the Slack API, we created a simple chatbot that can analyse the messages in a conversation on Slack. The bot, called Starterbot, reads all the messages and reacts only when it's summoned. Then, it takes the command and figures out what the user wants it to do by searching for keywords.

To be able to tell if it is summoned using "@Starter Bot", a regex is used, and if it is the case, the command is processed in the "handle_command" method.

The first step is to analyse the command phrase from sentimental polarity point of view. If it's mostly neutral, then we go ahead to search for keywords.

These keywords are the verbs of the phrase and they can lead us to the intent of the user request. In our case, if the verb used is any shape or form of the verb "to order", we go forward in processing its argument. This argument is given by the Semantic Role Labelling (SRL), a function of AllenNLP and contains the words linked to our verb.

In processing this argument, we make use of another important function of Allen, Dependency Parsing, which analyses the parts of speech of a certain phrase. For example, if the user request was: "I would like to order a pizza and three beautiful cats", the argument of order will be "a pizza and three cats", which are: a determiner, a noun, a conjunction, a numeral and a plural noun. We go through these words and start building entities for food items that contain the quantity, attributes and lastly, but not least, the name of the product.

[1, "a", "pizza"] and [3, "beautiful", "cats"] are our entities, but given our context of "food", we eliminate the second one with cats.

Now, our partial order, as we call it, is "1 pizza". The bot will ask if that is alright or if you want to add something else. If you do, then new food entities will be created and added to the order.

What's next

To finish the "chapter" of ordering food, we need to address some issues with certain food items and also get from the user the restaurant's name and basically that's it for detecting an intent in our context of food.

Example of usage:

```
command="I want to order a pizza and tell me how is the weather"
```

```
| Would you like me to order: 1 pizza? <Yes/No/I also want to order...>
```

```
command="I would like to order 3 cats"
```

```
Would you like me to order: 1 pizza? <Yes/No/I also want to order...>
```

```
command="I would also like to order 4 large portions of french fries, 2 hot coffees and two cheeseburgers"
```

```
| Would you like me to order: 1 pizza, 4 large french portions fries, 2 hot coffees,  
two cheeseburgers? <Yes/No/I also want to order...>
```

As we can see, given the context of food, only food items are taken into account, but there are issues with requests like “portions of french fries” as “french” is seen as the French nationality and not part of the composed noun “french fries”. By doing so, “french” is placed in the attributes of “fries” and “portions” to the name part as it is a noun. The end result for this entity is [4, [“large”, “french”], [“portions”, “fries”]] and we can see how it is displayed in the final order.

To implement any other intent, we should approach the task in a similar manner:

Keywords -> get intent from verbs -> analyse and process their arguments in certain contexts based on the intent.

Bibliography

1. <https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS17/paper/download/15426/14918>
2. <https://medium.com/@phanimarupaka/nlp-design-for-chatbots-4954c2527d88>
3. https://medium.com/@BhashkarKunal/conversational-ai-chatbot-using-rasa-nlu-rasa-core-how-dialogue-handling-with-rasa-core-can-use-331e7024f733?fbclid=IwAR1TJY_iwvqBoLx-2Ske9V85zUjD1sZ3Dh45-bwrORBMSepUXCPX_Gg-TA
4. <https://www.fullstackpython.com/blog/build-first-slack-bot-python.html>
5. <https://medium.com/swlh/deep-learning-for-text-made-easy-with-allennlp-62bc79d41f31>