
# Rubber Duck Chatbot Report

*CS4811: Artificial Intelligence Term Project*

## Team Johnny 5

*Ethan Visscher*

*Bester Mangisoni*

*Soham Sheth*

*Akshay Kumar Dosapati*

Submitted To:

Prof. Leo Ureel II

April 21, 2023

# Table of Contents

# 1 Introduction

## *1.1 Rubber-ducking*

This report will introduce an AI-powered rubber-ducking chatbot designed to assist developers in debugging code. Rubber duck debugging is the process of explaining code or an issue to a rubber duck to better understand the problem trying to be solved. A rubber duck chatbot takes this to the next level by providing intelligent, context-aware responses to better guide developers toward a solution. The chatbot doesn't necessarily find the exact solution to a problem but instead leads the developer to the desired solution. This approach can help developers recognize small mistakes they may have otherwise overlooked and save some debugging time.

## *1.2 Johnny 5 Rubber Duck Chatbot*

The Johnny 5 Rubber Duck Chatbot leverages natural language processing techniques to understand user input, identify the intent behind the input, and generate an appropriate response. Each response is pre-generated, but the bot can add context from the conversation into the response and it can select different responses for the same input. This gives the bot a more human-like feel.

The general flow of the chatbot is as follows:

- Ask the user about their error
- Categorize said error into a:
    - High-level error:
        - Logical error
    - Low-level error:
        - Compile error
            - Ex: Syntax error
        - Run-time error
            - Ex: Array out-of-bounds
- Ask questions about the specific error
- Continue to get user input and generate responses with the given context

- Loop until the user types “quit”

The separation of high-level errors and low-level errors allows the bot to help on a much more technical level with low-level errors. The bot would be able to detect syntax errors or attribute errors, for example, with a given error message from the console, and would be able to help accordingly. With a high-level error, the bot would follow the more traditional rubber-ducking strategy of having the user walk through the program. A more in-depth diagram can be found in the technical overview section.

The Johnny 5 Chatbot is built using Python and extensively uses the Natural Language Toolkit library. The library is used to match the user input with the pre-generated contexts and responses. This library was also edited to better suit the needs of this specific chatbot. More information about the library and the edits can be found in the technical overview section.

# 2 Technical Overview

## *2.1 High Level Design*

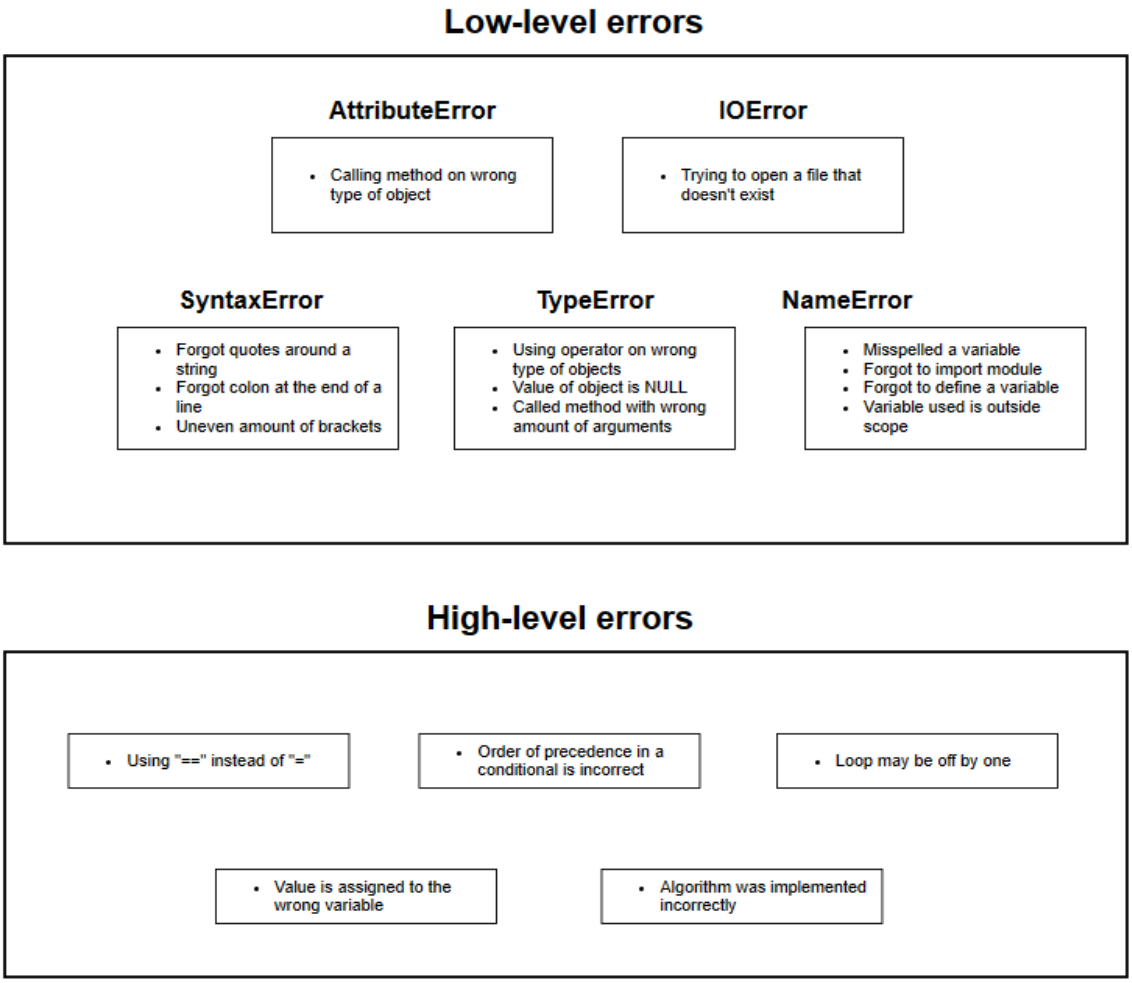


Figure 1: Context Classifications

Figure 1 above has examples of the context classifications of the bot. Each low-level error can generate multiple responses to help the user debug their program. Similarly, the bot can generate numerous responses for high-level errors. The context pairs are then loaded into figure 2 below:

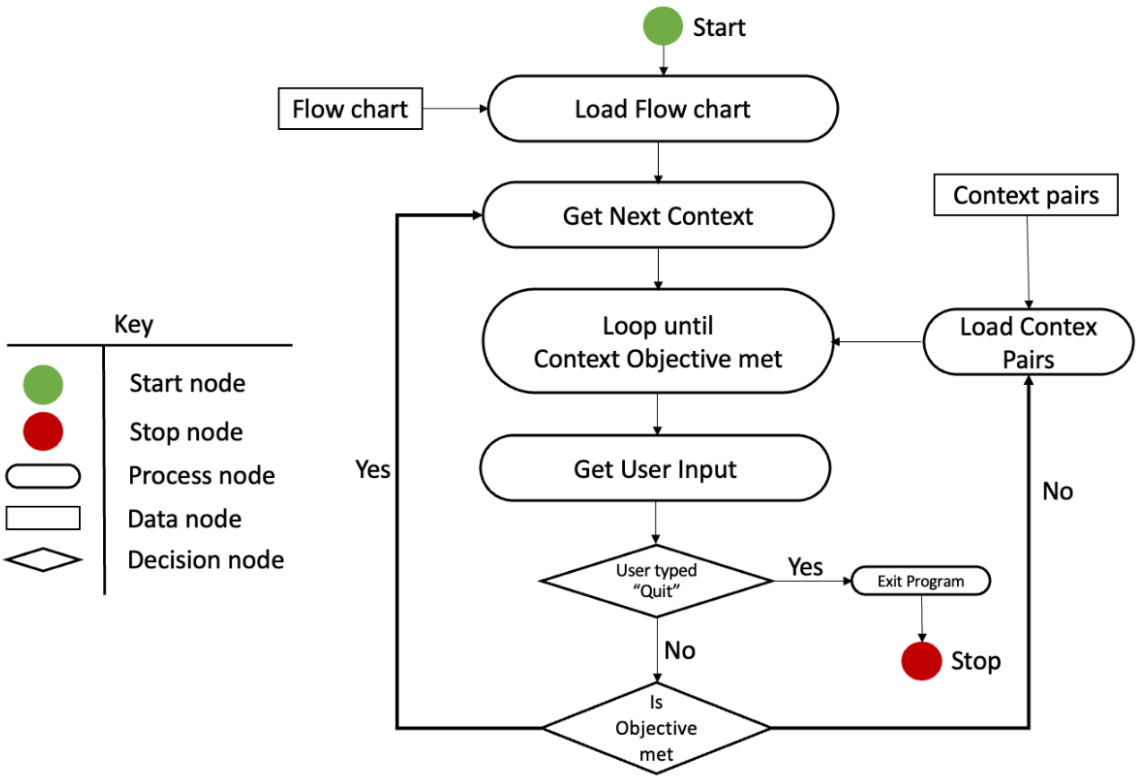


Figure 2: High Level Chatbot Design

Details about the stages in figure 2 follow below:

### *2.1.1 Load Flow Chart*

At the center of the chatbot mechanism is a flow chart diagram that guides the sequence of context the chatbot will be in during the conversation. The flow chart is represented as a list of tuples containing three components: the current stage, the parent stage and the decision value for stages following decision nodes. For testing the chatbot engine we tried to model the flow chart in figure 3 below. This particular flow chart entails that the chat bot will always start by asking the user what type of error they are getting. Whether it is logical or a runtime error. While in this context it is trying to match the user input to confirmation phrases like "yes, I'm getting an error", "or no, no runtime errors". Once it gets a confirmation or not, it then moves on to the next stage which is to do with logical errors or syntax errors. And the cycle continues.

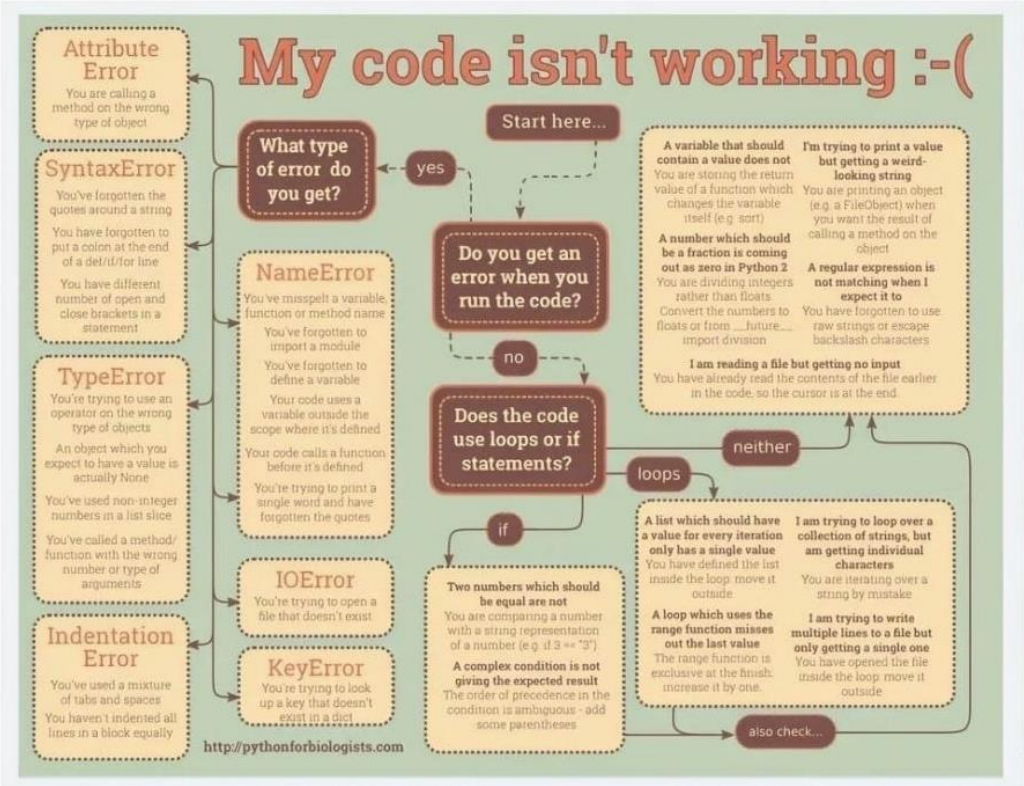


Figure 3: Flow Chart for Chatbot

### *2.1.2 Get Next Context*

This stage gets the next context depending on the previous context. If it's the first execution, it loads the first context in the flow chart. Looking at figure 3, the first context, is the question "Did you get an error when you run the code". The wording is different, however, in the chat program.

### *2.1.3 Loop Until Context Objective Met*

This stage is a loop that keeps executing until the current context objective is met. In the case of the first context in the flow chart in figure 3, this with when a confirmation (yes, yeah, ya e.t.c) or a negation (No, nope, negative e.t.c) is matched. The loops includes the "Load Context Pairs", "Get User Input", "Check for Quit" and "Is Objective met" stages. These are explained in the next sections.

### *2.1.4 Load Context Pairs*

This stage takes the current context and uses to query a dictionary of contexts or intents that include:

1. Patterns to be matched with user input. Below is an example of a pattern: "*(.*)(yes|ya|yeah|affirmative)(.*)"*. Once a pattern is matched, associated responses are sent to the user.
2. Responses to be given to the user.
3. A target regular expression: Used to determine if an objective is met.
4. A group number to return when the target regular expression is matched.
5. A target translator to convert any matched target expression to a common expression.

#### *2.1.5 Get User Input*

This stage simply Gets user input and submits it for processing.

#### *2.1.6 Check if user typed quit:*

This stage checks if user typed quit for termination purposes.

#### *2.1.7 Is Objective Met:*

This stage checks if the current context objective has been met (whether the target regular expression has matched user input).

### *Natural Language Toolkit*

NLTK is a library for Python that is designed to assist developers in working with human language data using natural language processing techniques. It includes a variety of modules and tools that enable developers to tokenize, tag, and parse text. It is widely used for developing machine learning applications, text classification, and sentiment analysis. For this project, we use a sub library of the library called NLTK.UTILS.CHAT. It is used in the Johnny 5 Chatbot to classify text as one of the loaded context pairs. The libraries were updated to better handle various intents and respond to various interactions. The library was extended for this project by updating the respond method, adding the GeneralResponse method, adding the TranslateTarget method, and the getGroupRegex method.

1. **Respond Method:** When the method is called, it takes in the user input and the intent parameter. It then loads the appropriate rules and regular expressions for that intent from the separate files, and uses them to determine the appropriate response.If the user input matches a specific rule, the method will generate a response based on that

rule. If not, it will call the generateGeneralResponse method to generate a general response for a more open-ended conversation

2. **GeneralResponse Method**: When the user input does not correspond to the specified intent, it generates a response for general chat. The chatbot can continue the conversation even if it cannot interpret the user's input.
3. **getGroupRegex Method**: It uses a regular expression pattern, a group is extracted. An area of the pattern wrapped in parentheses is referred to as a group, and it enables the capture of a particular area of the input string that fits the pattern.
4. **TranslateTarget Method:** This method is used to parse the user's input and extract relevant information that can be used to determine the appropriate response.

*Methods*

The main method for developing this chatbot was Natural Language Processing. Artificial Intelligence, A Modern Approach (8), outlines 3 primary reasons for computers to do natural language processing. These reasons are:

- To communicate with humans. It is more convenient to talk to a computer rather than use other means to interact with the computer.
- To learn, a computer could train on billions of lines of data to learn more about any topic.
- To advance the scientific understanding of languages.

For the Johnny 5 Chatbot, natural language processing is being used to communicate with humans. It makes for a convenient way for humans to debug their code, to talk to a computer. The chatbot does not learn and it does not advance the understanding of languages, natural language processing is used purely as a means to connect a human with the rules defined in the bot for this project. There are many different types of chatbots, such as generative bots and retrieval-based bots, but Johnny 5 Chatbot is a rule-based chatbot. This means that every answer is not a new answer and the bot does not go through any sort of training. This technique was chosen for ease of making and time constraints.

# 3 Conversation Examples

*Simple Greetings*

Although the chatbot isn't designed for "normal" conversation and is instead geared for development/debugging conversations, the bot can handle simple greetings. "Interesting" is one of the generic responses the bot will generate when it doesn't have a clear path to take the conversation. See figure 4 below. Other examples of transcripts can be seen in figures 5 through 10.

```
<------- Rubber Duck Chatbot ------->
Type 'quit' to end the conversation.

Hi there. I am a rubber duck debugging chatbot. So to begin with, are you getting a runtime or console error?
> hello
So are you getting any runtime or console error?
> how are you
I'm fine how are you.
> good
Interesting...
```

Figure 4: Transcript

*General Rubber-ducking For Logical Errors*

```
Hi there. I am a rubber duck debugging chatbot. So to begin with, are you getting a runtime or console error?
> No
So it's logical then i guess. So what's the issue
> I can't seem to have my program print the correct value I am trying to calculate
Okay. So your trying to calculate. What have you done so far?
> I have inputted my equation but it is always yielding the wrong answer
Okay. You have inputted my equation but it is always yielding the wrong answer. can you confirm thats working?
> It is not working
A little bit more please...
> It gives me a negative number when it should be giving me a positive number
A little bit more please...
> I looked and I had a sign flipped
Tell me more...
> it has fixed my issue
```

Figure 5: Transcript 2

Figures 6 through 10 are examples of the chatbot helping a developer debug a specific problem they are running into.

*Syntax Error*

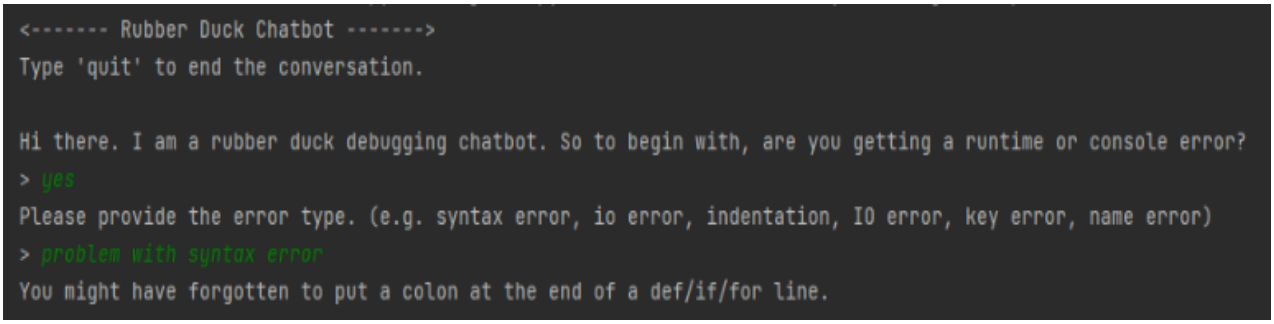


Figure 6: Transcript 3

*Conditional Error*

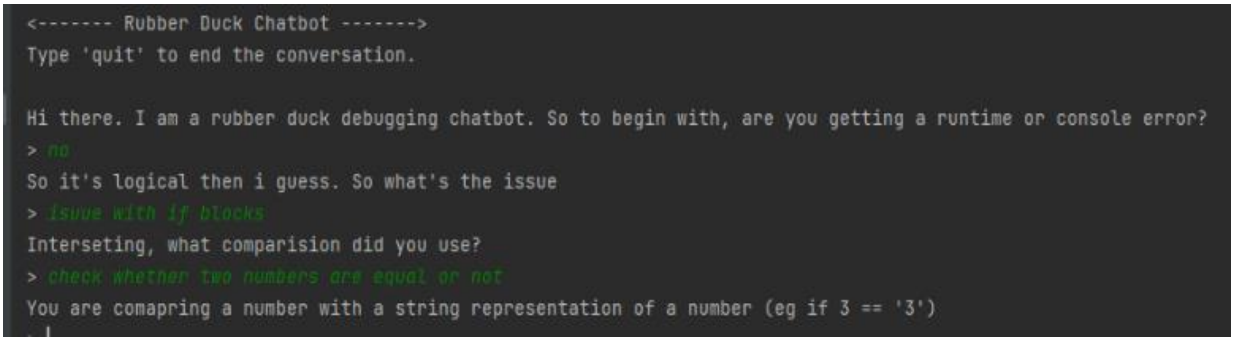


Figure 7: Transcript 4

*I/O Error*

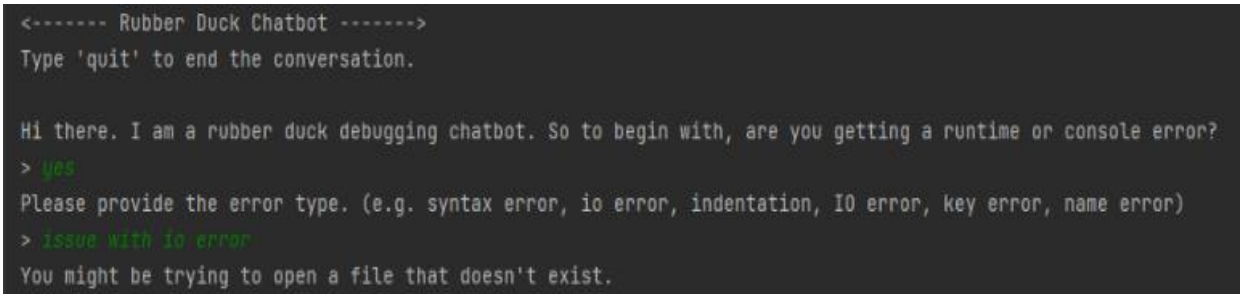


Figure 8: Transcript 5

*Attribute Error*

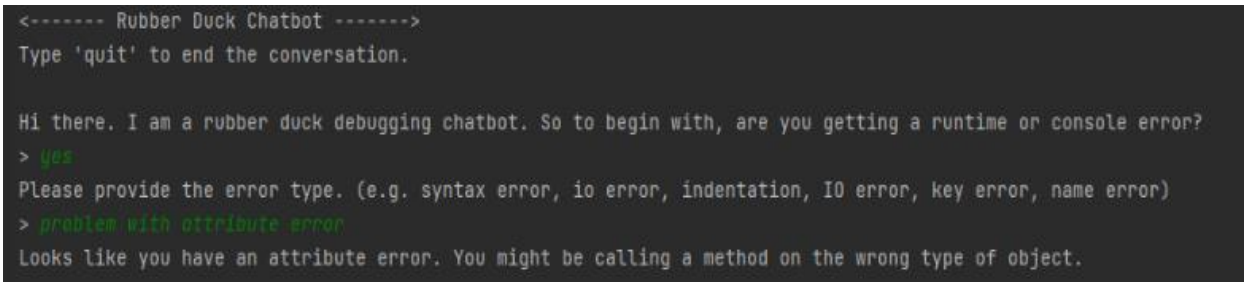


Figure 9: Transcript 6

*Type Error*


Figure 10: Transcript 7

# 4 Discussion

### *Strengths, Weaknesses, and Limitations*

Due to the fact that the Johnny 5 Chatbot uses pre-generated/pre-defined responses that are hardcoded by the developer, the bot has some weaknesses and limitations. The main cons of this approach are:

- There will always exist inputs from the user that the bot won't "understand" (won't have a response for).
- The bot is difficult to scale at a large level due to each response needing to be hardcoded by somebody.
- Some responses might not match up completely with the user input, causing confusion and breaking the illusion that you might be talking to a real person.

But, with this comes also some pros:

- The bot is easy to implement and easy to add new responses that the bot can generate.
- The developer knows all the responses that the bot can generate, there will be no surprise responses that could potentially be harmful or misleading.

### *Fitting Into Modern AI Research*

Even though the chatbot might not be the most advanced chatbot in the field, the Johnny 5 Chatbot can still bring valuable information to the AI research world. Mainly, the chatbot can be used for researching how users interact with chatbots. It is easy to prototype different responses and behaviors with this type of chatbot, so it could be useful to use when studying how humans interact with it. Just like ELIZA, this bot can be used to study the natural language communication between man and machine, but more specifically between software developer and machine. This particular chatbot could also be used to research the pros and cons of using a rule-based chatbot vs a generative chatbot (like ChatGPT). When using a chatbot like ChatGPT, there is the risk the bot could produce undesired and harmful outputs to the user. With a rule-based chatbot, like the Johnny 5 Chatbot, researchers could study if developers could get enough useful information from the chatbot without creating a more complex, and potentially harmful, generative chatb

# 5 References

1. Athreya, R.G., Ngomo, A.N., Usbeck, R.: Enhancing community interactions with data-driven chatbots-the pedia chatbot. In: Companion of The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon, France, April 23-27, 2018. pp. 143–146 (2018)

2. Heil, E., Neumaier, S.: reboting.com: Towards geo-search and visualization of Austrian open data. In: The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018. pp. 105–110 (2018)

3. Kacprzak, E., Koesten, L., Tennison, J., Simperl, E.: Characterising dataset search queries. In: Companion of the The Web Conference 2018, WWW 2018, Lyon, France, April 23-27, 2018. pp. 1485–1488 (2018)

4. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001. pp. 282–289 (2001)

5. Neumaier, S., Savenkov, V., Polleres, A.: Geo-semantic labeling of open data. In: Proceedings of the 14th International Conference on Semantic Systems, SEMANTICS 2018, Vienna, Austria, September 10-13, 2018. pp. 9–20 (2018)

6. Neumaier, S., Savenkov, V., Vakulenko, S.: Talking open data. In: The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portoroˇz, Slovenia, May 28 - June 1, 2017. pp. 132–136 (2017)

7. Porreca, S., Leotta, F., Mecella, M., Vassos, S., Catarci, T.: Accessing government open data through chatbots. In: Current Trends in Web Engineering - ICWE 2017 International Workshops, Rome, Italy, June 5-8, 2017. pp. 156–165 (2017)

8. Russell, Stuart J. (Stuart Jonathan), 1962-. Artificial Intelligence : a Modern Approach. Upper Saddle River, N.J.: Prentice Hall, 2010.