TestSummary Report

**Cat Assistant Chat Bot**



**Contents**

*1. Purpose* 2

*2. Application Overview* 2

*3. Testing Scope* 2

*4. Metrics* 3

*5. Test Environment & Tools* 5

*6. Types of testing performed* 5

*7. Key insights*  6

***1. Purpose***

This document explains the various activities performed as part of Testing of ‘Cat Assistant Chat Bot’ application.

***2. Application Overview***

‘Cat Assistant Chat Bot’ is a chat bot messenger application. The bot is smart, has a good knowledge about most of the modern things happening around the world and knows some computer science, music and movies. He also is well aware that it is a chat bot designed by humans and has a good weird sense of humor.

There are several modules like Implementation, Building the knowledge base, Text processing and Testing which are integrated to fulfill the purpose.

***3. Testing Scope***

**a) In Scope**

Functional Testing for the following modules are in Scope of Testing

* Module 1 – chatbot link to facebook
* Module 2 – opens and receives requests from module 1
* Module 3 – used by module 1 to error correct text

**b) Out of Scope**

Performance Testing was not done for this application.

**c) Items not tested**

This can be verified during UAT (User Acceptance Testing) where the connectivity is available or can be established.

***4. Metrics***

1. **No. of test cases passed/failed**



|  |  |
| --- | --- |
| **TCs** | **Tcs** |
| **Pass** | **Failed** |
| ~ | ~ |



***5. Test Environment & Tools***

*We used Browser for testing.*

|  |  |
| --- | --- |
| **Application URL** | https://www.messenger.com/t/catassistantrainbow |
| **Apps Server** | <http://ec2-35-157-45-86.eu-central-1.compute.amazonaws.com/> |
| **Database** | aiml |

***6. Types of testing performed***

**a) Smoke Testing**

This testing was done whenever a build was finished *(deployed on AWS, used facebook as a GUI)* for the Testing team to make sure the major functionalities areworking fine.

1. **System Integration Testing** 
   * This is the Testing performed on the Application under test, to verify the entire application works as per the requirements.
   * Critical Business scenarios were tested to make sure important functionalities in the application works as intended without any errors.
2. **Regression Testing** 
   * Regression testing was performed each time a new build is deployed for testing which contains defect fixes and new enhancements, if any.
   * Regression Testing is being done on the entire application and not just the new functionalities and Defect fixes.
   * This testingensuresthat existing functionalities works fine after defect fix and new enhancements are added to the existing application.
   * Test cases for new functionalities are added to the existing test

cases and executed.

1. ***Key insights***

* ***Error correction*** *occurs automatically, without taking into account language detection (hence, results vary based on original input)*

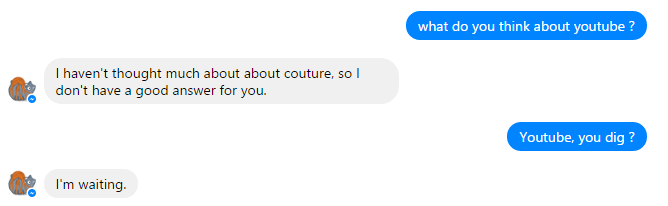
*Observable by user*:



**Proposal**: Add a language detection module, and inform the user that the app won’t serve relevant results in other languages.

* ***Error correction*** *changes words that are not part of the conventional corpus (and doesn’t identify that it might be a title, not a word)*

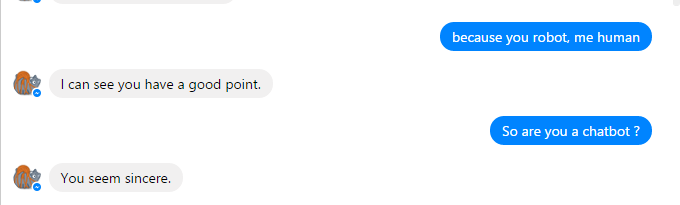
*Observable by user*:



**Proposal**: Extend with a urban/technology corpus, or keep unchanged word unless the distance of the correction is small.

* *Some* ***responses*** *are too often and* ***feel unnatural***

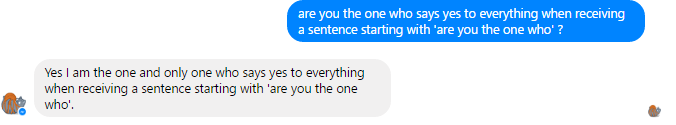
*Observable by user*:



**Proposal**: Increase the complexity of aiml / reduce the prevalence of this category of responses.

* ***Some rules*** *might allow for* ***strange interaction***

*Observable by user*:



**Proposal**: Some aiml refactoring might be needed

* *Linearity issues in conversation*

*Observable by user*:

Luca: hi

Cat Assistant Bot: Hi nice to see you!

Luca: hello

Cat Assistant Bot: Hows it going?

Luca: hi

Cat Assistant Bot: Hi how are you?

Luca: hello

Cat Assistant Bot: Hows it going?

Luca: hello hello

Cat Assistant Bot: Check your human privilege before you wreck your privilege

Luca: salut

Cat Assistant Bot: Hey, I'm over here!

**Proposal**: Check that the user repeats a certain idea in the subject, have a linearity in the conversation (not repetition without need – eg: You already said hi)

* *No support/response for emoticons*

*Observable by user*:

Luca: \*spams likes and emojis\*

Cat Assistant Bot: Hello?

Cat Assistant Bot: Is anyone there?

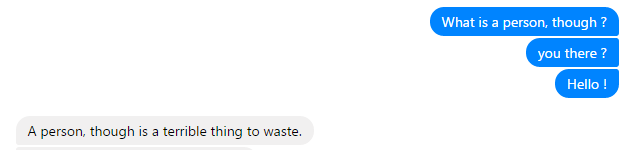
Cat Assistant Bot: Hello?

Cat Assistant Bot: I'm waiting.

**Proposal**: Support usage and responses for emojis/emoticons/images.

* ***Scalability*** *issues – if ~10 users are simultaneously connected, the application will either have medium to severe lag or will stack answers.*

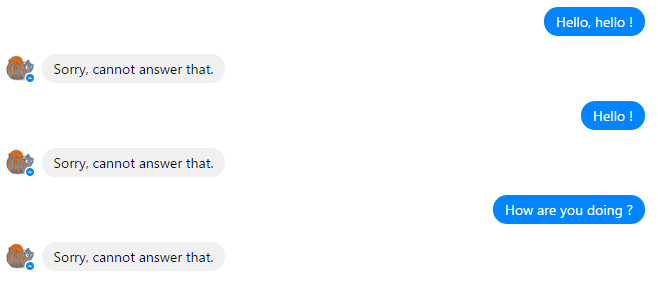
*Observable by user*:



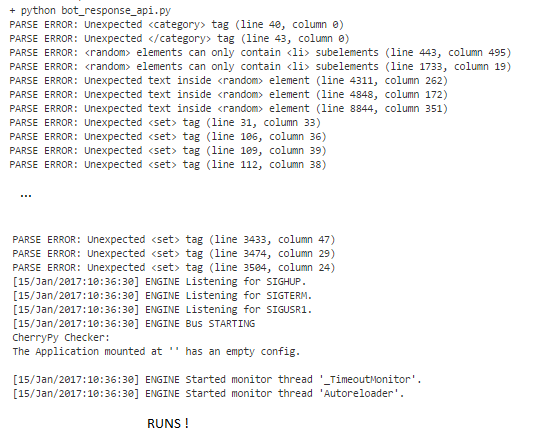
**Proposal**: Use of better machine resources and of threads (to solve async/queying latency, …).

* *The application sends messages as if it’s ok, even though the* ***database is down***

*Observable by user*:



*Observable by developer* (@Jenkins - Module 2, job 12 - /job/module%202/12/console):



**Proposal**: Set different responses in case module 2 database/server is down, so that the user knows that the bot may know the response, but isn’t available right now.

**Cat Assistant Chat Bot hosted here:**

[**https://github.com/2B5/ia-3B5/**](https://github.com/2B5/ia-3B5/)