NET-A-PORTER

NAP – UCL Project Team 28
Customer Service Chatbot

Wayne Tsui (Team Leader)
Aouss Sbai
Jason In

Bi-Weekly Report #1
27/1/2017

Project Overview (16/1/2017 – 27/1/2017)

For the past 2 weeks, our team have been reading up on IBM Watson Conversation documentation, including the use of Bluemix credentials and tokens and the IBM Node.js SDK. We have also been working with sample applications that incorporated the Conversation service so as to understand the interactions (format of inputs and outputs) between the front-end, the Conversation service, and the back-end.

Our team plan to have the first iteration of our PoC on 20th Feb, which is the end of the reading week. By then, our chatbot will have a simple front-end, working back-end API calls and populated Conversation workspace.

Meetings Summary

In our meetings, we cleared the misunderstandings that our team have on the API documentation. We also worked out parts of the project that can be developed in parallel, so that each team member is responsible for a part of the project. In this way, we can achieve efficiency and accountability. We have also set out key dates that documented the state of progression of our project which must be achieved in order to keep up with deadlines.

Tasks Completed

- 3rd Client Meeting completed
- Requested test NET-A-PORTER.com user account
- Received and walkthrough working API documentation with client
- Drafted development plan for term 2
- Set internal deadlines and milestones with detailed deliverables
- Assigned and agreed on allocated tasks amongst team members

Problems to be Resolved Before Next Meeting

We have 4 problems to be resolved before the next meeting.

Firstly, we need to work with the generation of credentials and tokens for front-end to call IBM Watson Conversation service within our application. Secondly, we need to import useful NAP database data into the Conversation workspace, such as creating a brand entity for NAP's brands and populating it. Thirdly, we need to incorporate NAP's J_SESSIONIDs, which is associated with a user account when they are logged in, into our back-end API calls for them to be valid. Lastly, we would require a test-account from our client that is populated with data such as order history.

Plan for Next Two Weeks

As we have assigned clear tasks to each team member, we will continue to work on our allocated tasks. We would use the lab sessions to keep one another updated on our progress and solve any individual problems that we might have, as a team, during that time. Some of the key tasks would be to create a template application that is able to get credentials and tokens so that we verify that our service calls are working, and also to write scripts that convert JSON objects returned from NAP API calls into CSV (Comma-separated values) files that we can import into the Conversation workspace.

Contributions

Wayne Tsui

I have worked on liaising with our client to set up our 3rd meeting, including getting the API documentations required, and going through the iteration of identifying shortfalls in the documentation and requesting for a more complete documentation. I have identified the need for a test user account and requested for it through our client.

In terms of development, I have summarised the key deliverables this term and worked out a product plan for our customer service chatbot. In the plan, it includes the dates that I have set for the team as internal deadlines and what milestones should have been achieved by then. I am tasked to work in the IBM Conversation workspace, which ensures that the user's plain text query get analysed and converted to useful intents and entities that the back-end can take as parameters to call the relevant APIs.

Aouss Sbai

I have been working on the architecture of the backend of the application. Together with the team, we have designed and organised the global skeleton of the chatbot, and how the different parts should communicate between each other.

My part consists of structuring the different API calls, receiving the queries by IBM Watson and then retrieving information from NAP database. As an outcome, the backend needs to send a JSON file, containing the response to the query so that the

frontend can display the results to the user. To do this, I am learning Node JS, in order to manipulate NAP API.

Jason In

For the past two weeks I have been looking at the IBM Watson APIs and the APIs documentation given by NAP. We had a meeting with our client Irina and Robin at NET-A-PORTER office two weeks ago to discuss about APIs that NAP could provide us. Most of the APIs are RESTFUL API except the user login API. J-session-id cookie (which we get it from login API) is required for user identification. Our team has also come out with a conclusion that Wayne will be working on the IBM Watson platform, Aouss working on the back-end and myself working on the front-end.