

# NET-A-PORTER

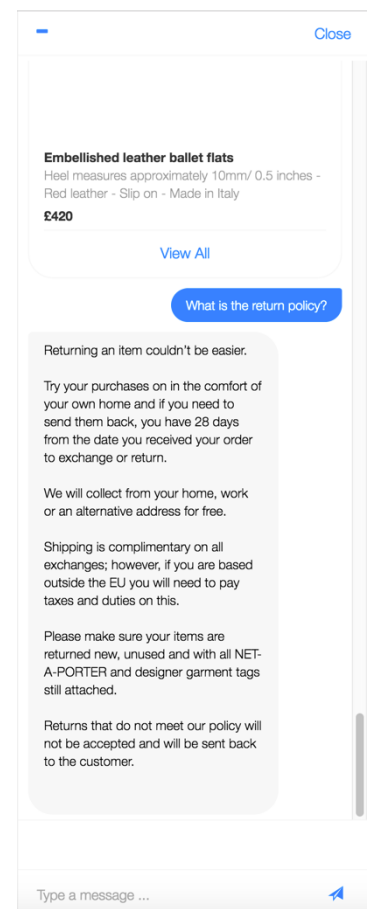
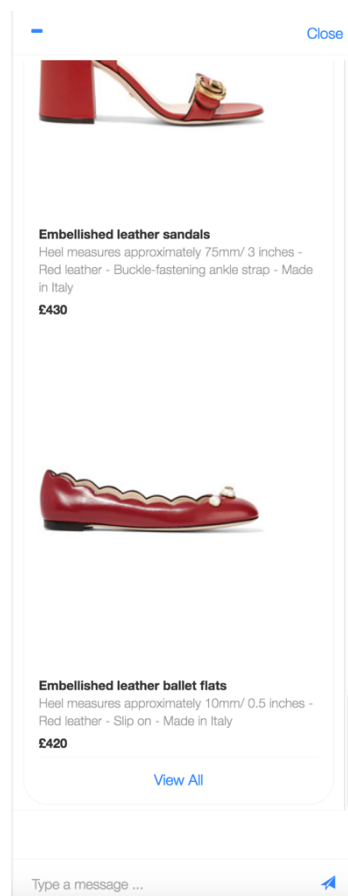
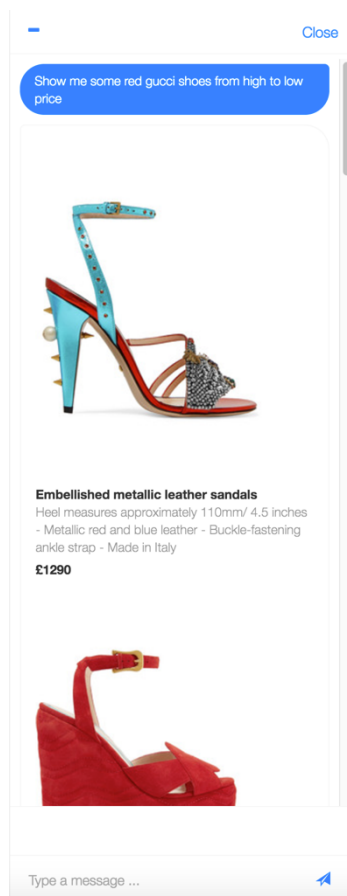
## NAP – UCL Project Team 28 Customer Service Chatbot

Wayne Tsui (Team Leader)  
Aouss Sbai  
Jason In

Bi-Weekly Report #5  
24/3/2017

### Project Overview (10/3/2017 – 24/3/2017)

In the last two weeks of Term 2, we made much progress in developing our chatbot MVP. We have integrated the individual components of front-end, back-end and external services such that we are able to showcase the chatbot to our client. We had completed most of the MoSCoW requirements. In the Spring break that follows, our team will work on consolidating our GitHub repository, writing the documentation and manuals as requested, and also work on our project website.



Screenshot 1. A query with Product Recommendation intent. A user could specify product category, price, brand, sorting order, colour and size.

Screenshot 2. This example search shows red Gucci shoes in descending price order. Clicking on a product links to its product page and a "View All" URL links to all products matching the user query.

Screenshot 3. User can also ask questions that are related to Frequently Asked Questions (FAQ), which are handled by Watson's Retrieve and Rank service.

## **Meetings Summary**

We met in our final lab on Tuesday, 21<sup>st</sup> March to show Ola, our TA, the end-to-end integration of our chatbot as promised. We also had our client meeting to showcase our chatbot MVP and get their feedback. During our meetings, we discussed on the deliverables that we need to produce by the end of April and how we could split the work efficiently and complete them remotely.

## **Tasks Completed**

- Client Meeting #3 – Presented and demonstrated Chatbot MVP and system infrastructure to our client and NET-A-PORTER's Head of R&D
- Received client feedback on Chatbot demo
- Built front-end to display API responses
- Implemented end-to-end integration of front-end, back-end and external services
- Implemented foreign language detection (Should Have requirement) using Watson's Language Translator service
- Implemented client feedback to generate typos for Watson Conversation entity values to handle a larger proportion of user queries

## **Problems to be Resolved Before Next Meeting**

During our 3<sup>rd</sup> client meeting on Wednesday, 22<sup>nd</sup> March, our team made notes on the feedback and points for improvement/future development that was mentioned. We then requested for our client's opinions on the priority of these pointers. We have since received the pointers of improvement/future development as prioritised by them. However, since time for development is limited and these pointers are beyond the scope of our requirements, our team will need to discuss and decide on which pointers to implement, should we decide to do so.

## **Plan for Next Two Weeks**

We will finish up the development of our chatbot, which includes documenting our code and implementing any additional features required. Also, we will plan the structure of the project website deliverable for this term and start to write up the content needed. Lastly, we will also document a system manual and user manual to allow future work on this project to be smooth.

# Contributions

## Wayne Tsui

I **coordinated the client meeting** which we presented our chatbot services and front-end user interface. I also sought to get feedback from them which we will include in our project website as I felt that it was important for UCL to gauge our client's satisfaction.

In terms of development, I made three significant contributions. Firstly, I **completed a "Should Have" functional requirement**. It was to identify if the user query was in a foreign language and direct users to that website's language version if it is supported. The supported non-English languages are German, Chinese, and French. I used Watson's Language Translator service to detect the languages in user query and generated JSON responses with the link of the foreign language NET-A-PORTER site if it matches any of the 3 languages mentioned above. Otherwise, the user query will proceed normally to the Conversation service.

Secondly, I worked on one of the feedback points of our client. This is to allow typo mistakes in their chatbot query so as to satisfy a larger proportion of user requests. To handle this, I **wrote scripts to scrap the results of an online typo generator** as there was no publicly available API to generate word misspellings. These word typos were generated to add-on as synonyms to the Watson Conversation entities so that the typos would be linked to the entity value. The typo generator handles cases for double letters, skipped spaces, missed keys and inserted keys. For example, *Browwn*, *Brownn*, *Beown*, *B4own*, *B5own*, *Btown*, *Bgown* would refer to "Brown", which is an entity value under *product\_colour*.

Lastly, I **organised our GitHub repository to include the Watson services workspace exports and the scripts that I have written to structure data**. This helps for a smoother handover of source code as many of the Watson's services are linked to my personal account. Having the service workspace exports helps our client to quickly replicate the services using their own credentials.

## Aouss Sbai

I have been working on completing the parser that returns the products following the request of the client. To do so, I had to think of how to make the code overall faster and optimise it in order to reduce the response time.

This mainly encapsulates the product-recommendation intent. Then, most of the focus has been put on debugging minor features and make sure that any user input would result in a correct output.

## Jason In

I have developed a front-end User Interface (UI) that is able to showcase the API that our team has developed. The UI is able to display *product\_recommendation* and *faq* intents, and will soon include *anything\_else* and *foreign\_langauge* intents.