Chatting about data

Interim report

Lorenzo Martinico - s1443541

January 26, 2018

The *Chatting about data* project aims to explore the potentials of using chatbot technologies to empower users by increasing their understanding of their own health through data. The project is split between different areas of human health: sleep, mental health, and nutrition. The latter provides an interesting variety of challenges that have been explored in academic and clinical trials, but never for the purpose of a comprehensive, general purpose consumer application. This might be an explanation of why its still not common to discuss your dietary habits with your favourite personal assistant, regardless of how ubiquitous the technology has become in the last decade. But while the trend of consumer voice assistants has been developing products as all-round helpful tools, which do not specialise in any one task, the lack of a clear winner in the space of textual digital assistants has resulted in the proliferation of programs with smaller scope. Known as chatbots, they usually specialize in accomplishing a small set of specific tasks, akin to the app model of the early smartphone era, and are similarly distributed through a centralized platform. Thus, just like food logging, a facet of the growing quantified self movement, has exploded with the smartphone, it also falls well within the scope of existing chatbots. And just as the smartphone has not completely solved food logging, with adoption still relatively low compared to tracking other vitals and abandonment being high, the relatively unexplored area of natural language assistants in the context of *m-health* still presents several unsolved problems.

Components

A nutritional assistant must contain two essential features: the storage of foods consumed by the user, and the retrieval of such information for the purposes of informing dietary choices. There is a large variation in how the two are accomplished among various existing solutions, and the aim of our project is proposing an implementation that, using a Natural Language interface, makes interactions easier and more engaging.

Automated food logging in most commercial solutions is predominantly text based, with the user typing in the name of what they had on the day, selecting the closest match from a database of foods and their nutritional value, and estimating the amount of portions they consumed. It has also become popular to add an option to scan the barcodes that are present on all commercial food packaging, linked to their database entries. Some experiments have been conducted to estimate calorie content through image recognition, but there are no commercial products available to the general public, and most experimental implementations require a marker for dimension, which is a significant detriment to usability.

We propose implementing a hybrid textual and photographic input food assistant, where portion size can be specified by typing in a quantity, or by characterizing consumption in comparison to previous meals. This eliminates the cognitive load of using a complicated interface or trying to remember precise portion sizes, and will initiate a first user reflection on how much they have been eating recently, rather than focusing on number of calories tracked, based on the belief that relative quantities and overall trends could be more useful than rigorous data collection for the purposes of maintaining long term engagement.

The harder problem is information retrieval, or rather how to extract value from the data and display it in a useful manner. The traditional techniques include calorie counting and breakdown of nutritional information based on macro and micro nutrients, usually shown as a table, a graph, or a progress bar. These strategies have been disputed by the nutrition community, and do not seem to be effective on the general public, resulting in poor engagement and little understanding of underlying dietary issues.

Our solution involves the chatbot periodically prompting the user to initiate a discussion on what they have been eating recently, which we hope will help the user reflect on how their food choices are affecting them. There seems to have been little progress on this specific issue, so any naive solution we will come up within the scope of this project is going to be an improvement on the state of the art. The lack of research in this area means that, barring a more comprehensive study involving a large dataset and the assistance of scholars in nutrition and behavioural psychology, we will be required to handcraft a small number of simple heuristics that will detect a dietary pattern we can predict, and initiate a conversation where we can give initial suggestions, and try to predict the user response.

Architecture

As chatbots have become more common, programming your own has also become easier, with many startups and large companies providing Software as a Service (SaaS) platforms to develop integrated solutions. This has been made possible by complementing older pattern matching models of text recognition with modern machine learning techniques (most famously intent recognition). Among the many competing platforms, *api.ai*, now *Google Dialogflow*, has been selected, for its ease of use and large choice of platforms it integrates with. The distribution channel for the chatbot will be Facebook Messenger, because of its large user base. For the backend server, a small *Node.js* app will be hosted on an *Heroku* instance. The choice of these tools was used to maximise productivity in the prototyping phase, but there are several drawbacks that make them unsuitable for a production system. Most significantly, the chatbot platform lacks robustness, requiring the programmer to predict exhaustively where the user may take the conversation. Additionally, the library of entities which Dialogflow already recognises is limited, which makes it challenging for the bot to successfully identify word tokens that correspond to a large vocabulary, and requires finding a comprehensive list of all words that may belong in that category. The use of free (as in beer) tools has also the drawback of adding some usage limitations, which has caused us to have to re-engineer the application several times while adding more components to the prototype.

The "frontend" of the application is a bot account deployed through the Facebook Messenger marketplace, and hooked into the Dialogflow platform. Dialogflow allows the programmer to specify a number of *intents* that the bot should recognise, based on the initial examples provided by the programmer, and supplemented by a machine learning algorithm to improve predictions. Intents can be parametrised with *entities*, which represent categories of meaning, and can be built-in or user created. Intents may have very different structure, and parameters can be marked as optional or required. If not all required parameters are specified with the intent, it's possible to generate follow up questions, or handle missing data on the server side. Once an intent is triggered, its response can be set through the web console for the application, which has facilities for sending multiple messages, customize the response based on the messaging platform used (and thus incorporate native UI element), and providing multiple alternative responses, to be cycled through to give a greater impression of naturalness. To allow more complex logic, the triggering of an intent can also generate an *action* to be sent and parsed by a webhook. The webhook can also invoke *events*, which are associated with intents and will trigger their response. Finally, each intent contains inbound and outbound *context* variables, which can be used for data that is not explicitly declared as a parameter by the user, or for follow-up intents, which build a conversation flow where the bot is "aware" (by storing in memory) of previous steps and can create a reply based on the rest of the exchange.

For our chatbot, the principal intents will be the onboarding welcoming intent (triggered when the user first starts a conversation), a food logging intent with type of food and portion quantity as parameters, a picture logging intent (triggered when a Facebook Messenger image is received, with potential follow ups to clarify if the image has been correctly identified by the chatbot), and potentially (information prompts will be largely served by the bot itself) food retrieval intent ("What did I eat on date?"). There will also be an intent with no user triggerable sentence, but only tied into an *action*, for nudging the user about their habits. The implementation of any eventual additional feature would start with the addition of a new intent.

The backend in our implementation is used to respond to all non-trivial intents, access the database, and call various APIs. As mentioned, we are using a Heroku instance running Node for hosting the webhook. An *Express.js* app responds to HTTP requests from Dialogflow when a new action is started and sends the bot's reply as response. When the user is first on-boarded, a unique user ID associated with their Messenger account and their preferred name are stored in a *MongoDB* instance (hosted on a free webservice). When a new meal is logged, if the input is textual, the food (or foods) is passed through the *Nutritionix API*, which finds the nutritional values of that item, and logs it in a new meal in the database. If instead the food is logged as a picture, the *ClarifAI* API is used to identify what food is in the image. The API provides a certainty percentage; if there are more or less than one prediction over an arbitrary threshold, the user is then asked for clarification, and from there the same process as in a textual representation is followed. After these operations, the food that was added to the database should be used to update a predictive model of what the chatbot should say next. While this has not yet been completely implemented, since the free account limitations of the Nutritionix API make gathering training data a long process, the naive way to classify food would be using a clustering algorithm. Given some training data (nutritional values of a list of common foods, as of *LanguaL.org*), they can be aggregated in a number of clusters based on their nutritional composition. If all goes well, manual inspection of recognisable food added to each cluster will be used to identify the clusters as well known categories of food; there is however a risk that clusters will form in a non-intuitive way, which might be helpful in finding out more interesting nutritional facts that we hadn't predicted in advance, but could also be unexplainable. To each category, a potential conversation will be associated for the chatbot to remark upon, in case of excessive or insufficient consumption. Each of these dialogue prompts will have an associated probability value. Depending on what kind of food the user is eating, and how often the conversation has been brought up, the probability values will be increased or decreased, until the next conversation, or after a specific amount of time if the user has not interacted with the bot in a while; then the conversation with the highest probability will be triggered. To ensure a successful conversation, potential scenarios the user will want to react to have been sketched out to follow potential responses for the user. A tool like *botmock.com* can help to design these conversation flows. Additionally, conversations will be crafted for general case scenarios, such as insufficient tracking, eating too much or too little.

Evaluation

As with all software artefacts that have a user facing component, testing can be lead both on a technical level and on a usability level. Since the chatbot infrastructure contains many different components, it will be necessary to conduct some testing on the robustness of each.

The preferred method for testing software projects is combining unit testing with end to end tests. However, since our Node app contains little internal logic, off-sourcing most of its computation to external webservices, and pulling together different data streams, making it difficult to conduct programmatic testing. We will have to limit ourselves to augment our server-side application with assertions, to ensure that fundamental invariants necessary for its correct functioning aren't broken. Manual testing can be used both during development, to compare the performance of different APIs, and during an evaluation phase, to ensure the system is robust. Dialogflow offers a console within its website to test if utterances are associated with the correct intent, and to correct any misidentifications. Manually providing corrections will supplement the machine learning system with training data to better identify intents; however, this should be done by an external tester before the chatbot is exposed to users, because the developer may not guess exhaustively all possible variations of sentences that should be recognised.

Manual stress testing should also be done on the image recognition system, again for comparison between different APIs and to test how well it performs. Images of both ambiguous food, like cucumbers and courgettes, and non-food items should be tested, as well as pictures where the food is blurry, out of focus or partially occluded.

While most testing is done through the Dialogflow console or a single Facebook Messenger account, some testing should also be conducted using a different account, to test authentication and database management have been correctly implemented.

Usability is in general hard to measure. When there are clear usability issues, simple user testing can quickly expose them, but it's difficult to measure how good an interface is. Since our concern is comparing the usability of a chatbot interface against that of existing food logging systems, we will have to gather data on users of both systems. We will be asking a small sample size (around 5 people each) to use either a commercial food logging system (My Fitness Pal) or the chatbot for a week. To obtain some objective comparative data, we can administer a survey on completion of the week, with questions on their experience recording their meals, and on how the two different modalities of interacting with their data affect dietary choices. The surveys should be complemented with another experiment, for example an interview, to get a fuller picture of how users interact with the chatbot, and what they think its strengths and weaknesses are. Interviewing users of the food logging app shouldn't be necessary, because its effectiveness has already been proven in studies and in the marketplace. Participants in the experiment should come from as many different backgrounds as possible; specifically relevant to diet are socio-economic status and education. Conducting the experiment on such a small sample size will not give a comprehensive overview of the system's quality, nor of its effectiveness over the long term, but only the anecdotal evidence necessary to go from an initial prototype to a fully formed usable tool. Gathering evidence of long term effectiveness and casual intention (willingness to continue using the service) in further studies will be necessary to evaluate the advantages of chatbots over the traditional interface.

Further work

The project has the potential of exploring even more complex solutions, but lack in time and frustrations in the development have caused us to abandon or postpone some of these ideas.

Given the close ties between chatbot usage and social media, an interesting question would have been whether social augmentation could improve user experience and motivate effective positive change in someone's diet.

In recent years the use of social networks like *Instagram* for uploading picture of food, especially when visually appealing or perceived as healthy. To decrease friction in the usage of our chatbot, we started to implement a function to allow our users to log into an Instagram account and automatically log all pictures taken into their food history. After the login functionality had been implemented, however, our program was disabled from using the Instagram API, because scraping the service for purposes other than creating a new frontend to the application are not allowed by the Terms of Service, as we later found out when talking with some Facebook employees at a separate event. Given enough time, it might be possible to explore the possibility of implementing similar functionality with similar photographic hosting platforms like *Flickr*, which seems to have a much more permissive API for non-commercial purposes.

Another interesting potential research avenue would have been exploring if the chatbot interface could be used in the context of a group chat. Mobile apps that force users to exercise or eat better by "shaming" them on social media or taking a small amount of money have enjoyed various degrees of success. A similar approach could be used within a group chat to gameify good nutrition practices, by challenging the members of the group to periodically meet set nutrition goals, or by praising users who have been following healthy eating guidelines, and reprimanding those who have not. Such challenges could also be extended at the individual user level for increase in motivation.

There would also be many benefits in collaborating with nutrition researchers. The main improvement could be in creating a more complete set of nutritional heuristics for conversations to be initiated, and a suite of general nutrition advice. The chatbot could also be customised based on distinct information about the user, such as initial body type and dietary goals.

Finally, greater precision in the quality of prediction could be obtained by training a better food recognition model, which could ideally also perform segmentation to recognise when different meals appear in the same image, and 3D volumetry to correctly guess portion sizes without user input