TicketBot: A Natural Language Interactive Assistant for Ticket Selling

No Author Given
No Institute Given

Abstract. e-Commerce interaction can provide company information to customers. However, most websites can’t consider specific customer needs. Interactive virtual assistants are a way to provide a more personal user experience. This paper explores the development of a ticket selling Chatbot using Microsoft’s Language Understanding Interactive System. The challenges and possibilities of such agents are discussed.

Keywords: Business intelligence; Virtual assistants; Expert system

1 Introduction

In the last years computers and smart-phones became more interactive and personalized interfaces should take into account user preferences. This new trend in user interfaces started with iPhone’s Siri [sir] and is becoming a standard. For example, in its 2017 F8 conference, Facebook previewed new interactive web interfaces for companies increasingly based on the instant message services. This paper focuses on Microsoft’s Language Understanding Intelligent Service (LUIS) [lui]. LUIS is a Natural Language Processing online interactive framework, created by Microsoft, that allows to add some learning capabilities to a Chatbot, namely in understanding requests made by users. It accomplishes that by detecting intents and entities in the sentences. LUIS is used by Chatbots developed with Microsoft Bot Framework in a simple way: in fact, this Chatbots are computer programs written in C# or Node.js and each LUIS intent is mapped to a program’s method [mbf]. To prove this framework’s capacities, suppose we want to create a Chatbot to a ticket selling company: TicketBot.

2 Intentions, Entities and Active Learning in TicketBot

TicketBot shall help the user on the ticket buying process and support event search, to events such as concerts, festivals or plays. Each search can be filtered by event name, type or date. Each one of these functionalities correspond to an intent. Intents are the intentions or desired actions conveyed through the utterances (sentences). They match user requests with the actions that should be taken by the app. Therefore, it’s necessary to add three intents: buy tickets for event, search events by type and search events by date. The following example
sentences can be used: *Buy tickets for event; Music events; Events for tomorrow.* It’s also necessary to create the entities involved in the created intents.

Entities are key data in the application’s domain. An entity represents a class including a collection of similar objects (places, things, people, events or concepts). They describe relevant information to the intent. For the *Buy tickets for event* intent, we create the *Event Name* entity. For the *Search events by type* intent, the entity *Event type* is created. Finally, the entity *Date* is created to the *Search events by date* intent. For each example sentence previously given, the entities are: *Music events* – Entity: *Event type* (*music*); *Events for tomorrow* – Entity: *Date* (*tomorrow*). However, with just this examples, the application will not have a great performance. So, it is necessary to enrich the model by providing properly labeled sentences and phrase lists, that will avoid mistakes on intent/entity identification and phrase interpretation. Among others, phrase lists for dates, names of the most popular artists or sports teams must be added.

LUIS makes use of active learning (its breakthrough feature), to improve itself [WND+15]. In the active learning process, LUIS examines all the utterances that have been sent to it, and calls to the application’s programmer attention the ones that it would like him to label. LUIS identifies the utterances that it is relatively unsure of and asks the programmer to label them.

3 Conclusions

When well used, and adapted to the desired context, LUIS is a relatively simple but very interesting tool to help applications understand users’ requests. If the intention is to create a Chatbot that privileges user experience, then this technology acquires an even bigger importance. However, more than creating a model, it is necessary to gradually provide model improvement by using different LUIS functions: Add/Edit/Remove intents/entities/phrase lists; Provide example utterances and label them; Label suggested utterances (Active learning); Train the application; Test the application; Publish the application. This process - specially Active learning - allows continuous improvement of the application and it is not only a way to better understand the model’s evolution, but also a way to have a better idea of what requests are made by the user and if they are made as expected.

References


