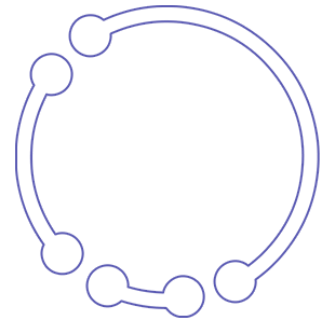


TERRANOHA



How to do NLP Episode 3: Setting up a conversation flow

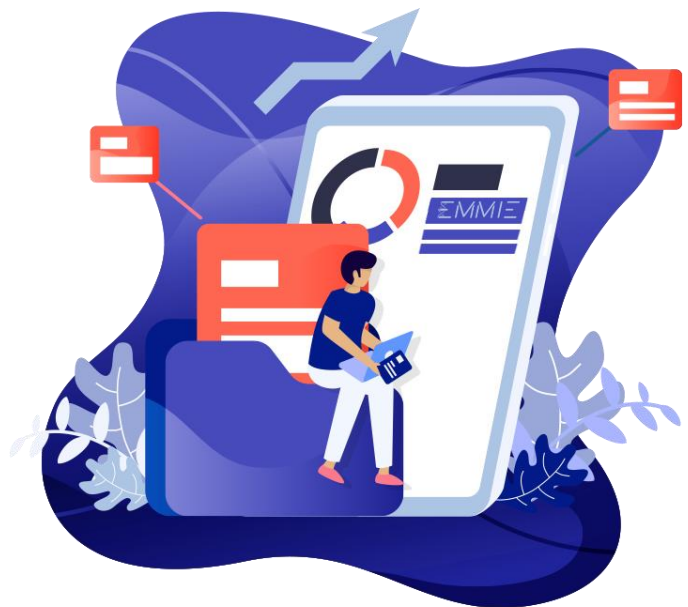
07/11/2022

Previously

In the first two episodes, we saw how to set up a structured and easily accessible database from a huge amount of data in the form of text files.

This database allowed us, in the second episode, to build and perfect a first NLP model able to handle several domains related to RFQs (Request for Quotes).

Today, we will look at how to set up a conversation flow between the requester and our artificial intelligence.



Objective

We now have a solid model, which can effectively interpret several different intentions. This model is based on real-life examples, so we are confident that it is not biased and will meet our client's needs.

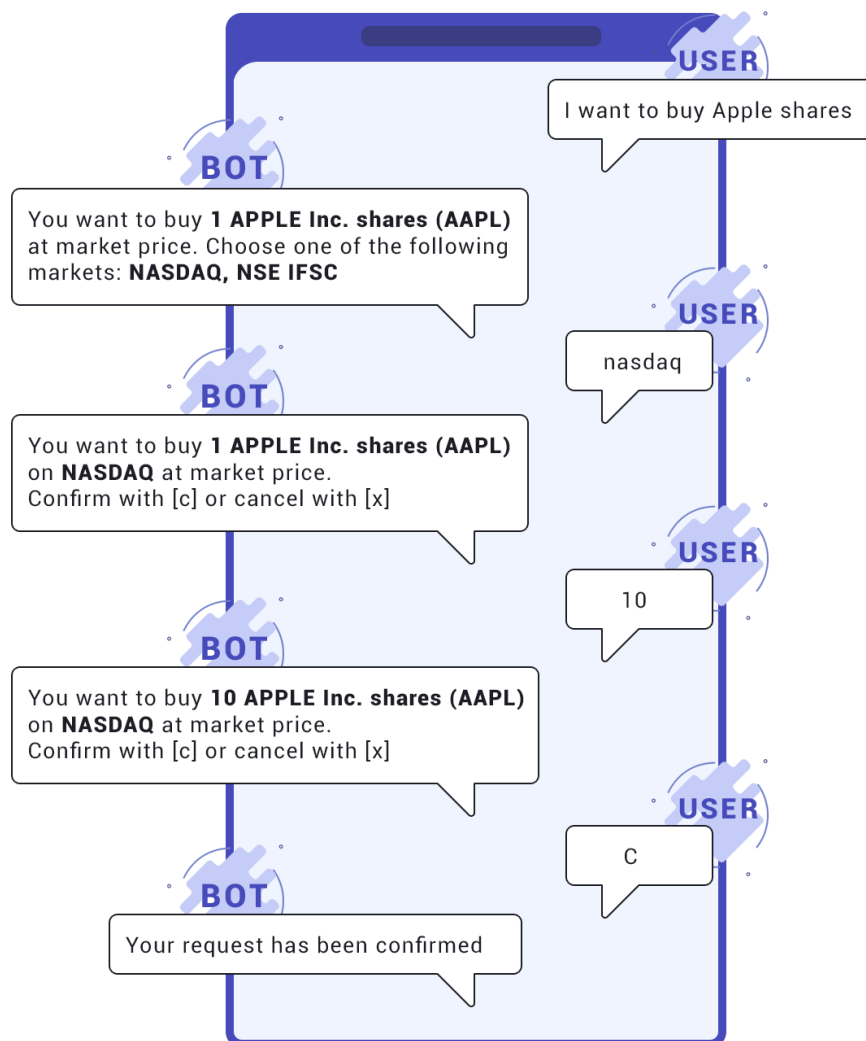
The next step is to be able to set up fluid conversations between our customers and AI. We will see that there are two philosophies in this area and focus on the choice we have made.

The problem of conversation management

Conversation is a complex interaction, which is why creating an AI that can simulate a conversation is so complex. If you are further interested in the characteristics of human conversation, you can turn to this academic paper from Stanford University: "[Chatbots & Dialogue Systems](#)". We will summarize here the general thinking, and the essential points to our problem.

The aim is to create a fluid exchange between the different interlocutors. It is therefore necessary to go beyond simply understanding a request: it is now necessary to put this request into perspective in relation to previous exchanges, and even to anticipate future exchanges.

We also must take grounding into account. Grounding is an essential concept. It ensures that the chatbot gradually understands the elements of the discussion as they arrive.



We see here that the messages taken separately do not make sense. Yet it is natural for any human being to deduce the overall meaning of a conversation created by a sequence of lines. We use our immediate memory to do this, as well as our ability to understand contexts, particularly by knowing the speaker and putting his or her words into perspective in relation to what we know about him or her and his or her previous lines. This is what we need to give our AI: an immediate memory and knowledge of the context.

There are two distinct approaches to solving this problem. The first is to rely solely on the NLP tool to manage the conversation flow. Indeed, a chatbot would not be complete without a conversation management system.

The second is to use an "in-house" solution, which can be adapted to our needs. To do this, it will be necessary to keep in mind the crucial elements of the previous requests, and to process the following ones in the light of these.

Motivated by the specific domain of finance, and by the desire for specific user management, we chose the second solution. Such a solution will indeed allow us to adapt the behavior of our AI to the specificity of the users.

We will present the architecture we have chosen, as well as its advantages.

The choice of home-made architecture

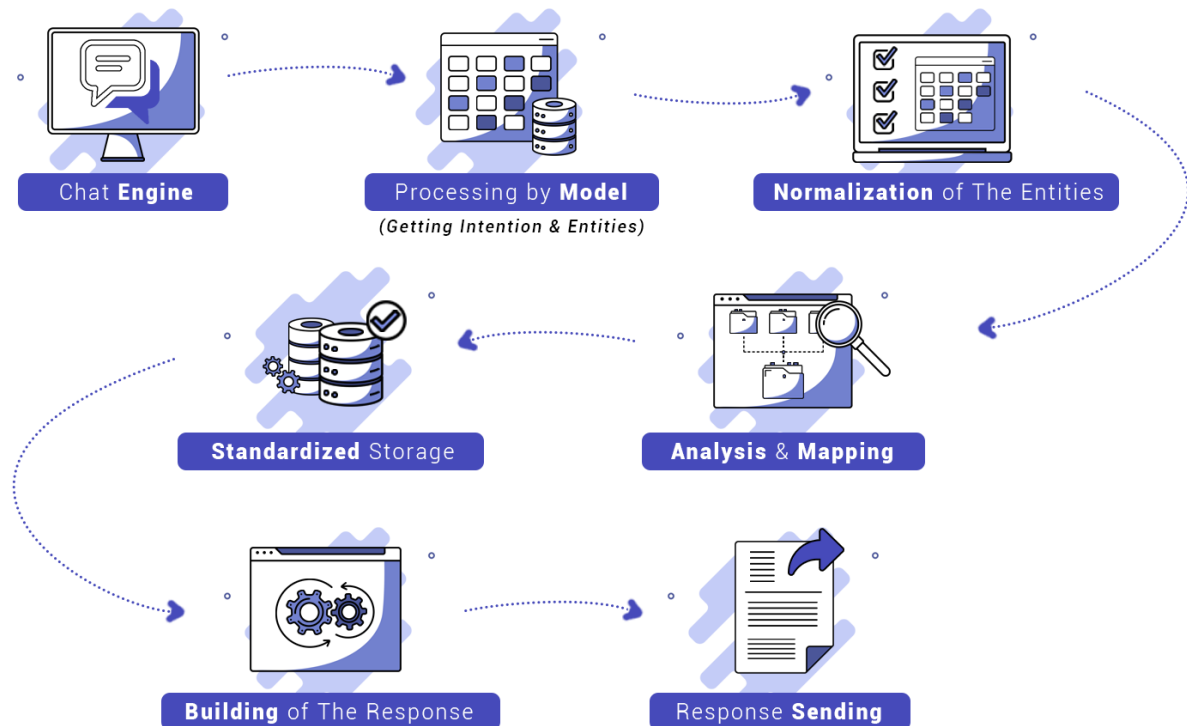
The choice to set up our own architecture was obvious from the beginning of the project. This choice allowed us to implement a more flexible and adaptive management of conversations, without remaining locked in a strict interpretation of the model: It has allowed us to add intelligence to our model.

This architecture allows us to add a "user profile" dimension to our model. This allows us to add an interpretation to a response by considering the user's business domain and the requests they are likely to make.

It also simplifies the management of the conversation history. Storing a live conversation will allow us to manage the flow of the conversation, but at the same time create a history, which will be extremely useful for advancing our model, and for acting quickly in case of problems.

Moreover, this choice allowed us to set up a systematic standardization of our entities, using only known data, without leaving room for interpretation.

Here is a presentation of our simplified architecture:



Each message that will be sent to the bot will go through all these steps. The message will first be sent to the NLP model, which will analyze it to determine the entities of interest. These entities will then be normalized, i.e., transformed into a finite number of pre-defined values. These values will then be linked in a dictionary to the attributes to which they correspond. Finally, these attributes will be stored in a database which allows us to reconstruct conversations, and which constitutes the

"memory" of our tool. It is this database that will allow us to implement the concept of conversation and grounding.

We will now show how our architecture works with a concrete example from a conversation.

A concrete flow of conversation

For this example, we will repeat the conversation presented at the beginning of this whitepaper. The user wants to buy Apple shares. To book his request, we need a certain amount of information:

- The name of the company, with its short code
- The market on which to buy it
- The number of shares
- The price at which he intends to buy them (by default, the market price)
- The quantity (by default, only one)

Plus, as this user is registered with us, we know that he is specialized in the American markets. He will therefore, logically, buy shares on the New-York markets.



For a given user, the initial query will therefore be interpreted by the NLP, but also in the light of what is known about the user. Here, our user is obviously interested in buying shares, we are not going to suggest buying commodities, for example.

This can be done by allowing the user access to only certain intentions, or by checking the interpretation of our model in the light of what we know about the user.

Once the user's general intention is understood, we will ask him for the additional information necessary to execute his request, in this case, the market on which he wants to execute it.

We can then propose a list of markets that are appropriate to its field. Here we see that we only have a list of two markets to propose, which simplifies the exchange.

Our user chooses the market, specifies the quantity, and then executes the query.

Conclusion

In this third episode, we have seen how to deal with a flow of conversations, and all the tools that we put at its service.

Indeed, the architectural choice we have made allows us to add intelligence in addition to pure NLP, notably user management. It is possible to imagine even more cogs that will improve your AI and give you assets to allow a better analysis of your performance, for instance. Thus, once a conversation has been recorded on the platform of your choice, you can imagine all sorts of monitoring and analysis tools, or even automatic creation or annotation!

Stay tuned for next season!

Web

www.terrano.com

Mailbox

sales@terrano.com

Address

Terrano SA, Route de Pré-Bois 29,
1215 Geneva - Switzerland