altice labs

A better way to create intelligent virtual assistants

Artificial Intelligence, Bot, Customer Experience, NLP, Virtual Assistants

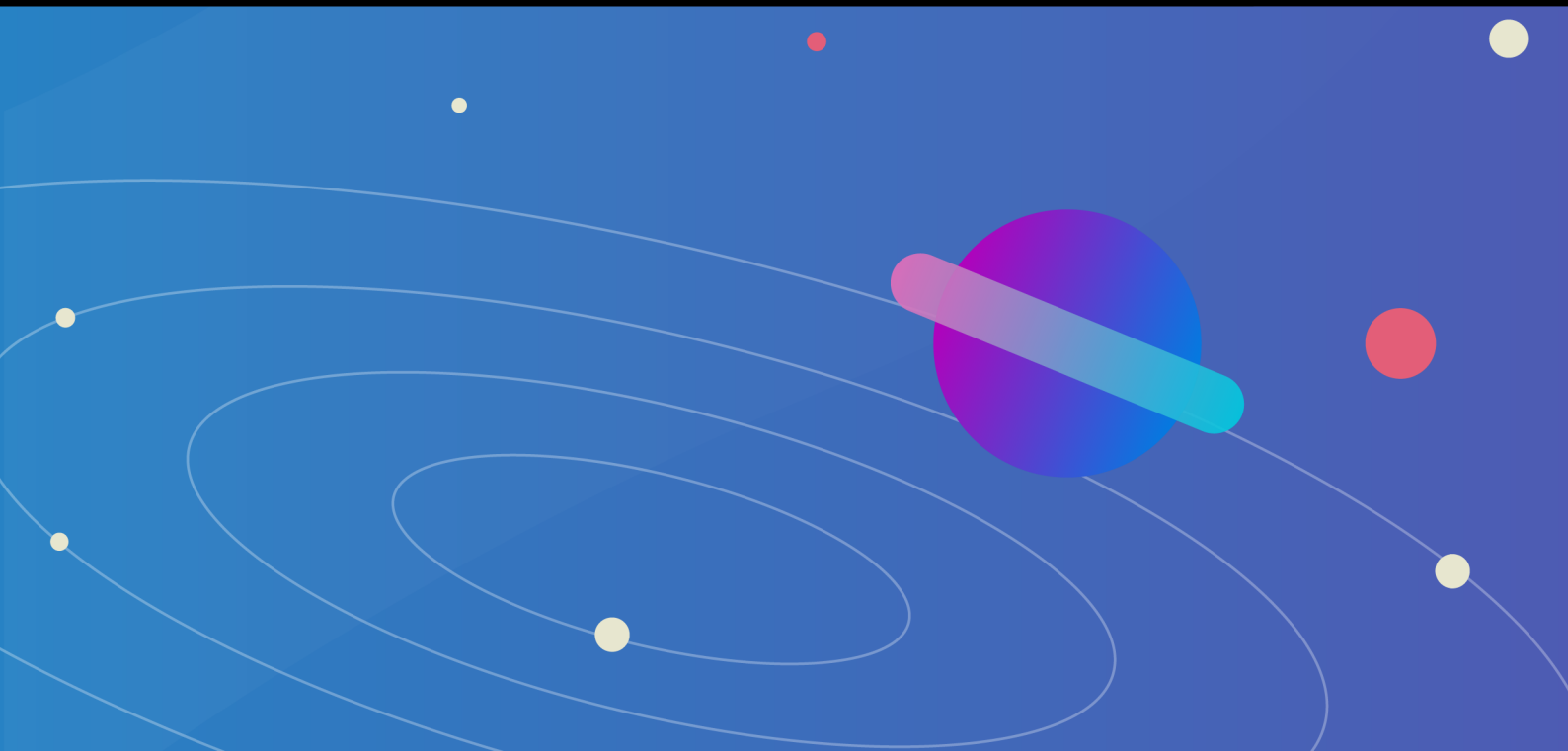
White paper

Version 1.0, May 2020

Overview

Companies that want to create intelligent virtual assistants to increase the customer experience and/or automate their customer interactions face many challenges, many of them related to the complexity of building an artificial intelligence (AI) assistant able to satisfy all business requirements.

In the current software market, projects that implement intelligent virtual assistants tend to start small, often with a single use case, which eventually reaches the gold status. But, as more use cases are developed, problems begin to appear. Namely: training dataset growth and the conflicts that arise in the intent detection, integration with IT systems, and the usability of current solutions that don't help to handle the virtual assistant teaching process. While the two last problems can be mitigated, dataset management is hard and very complex.



When the dataset grows, it is prone to become unmanageable, leading to the following problems:

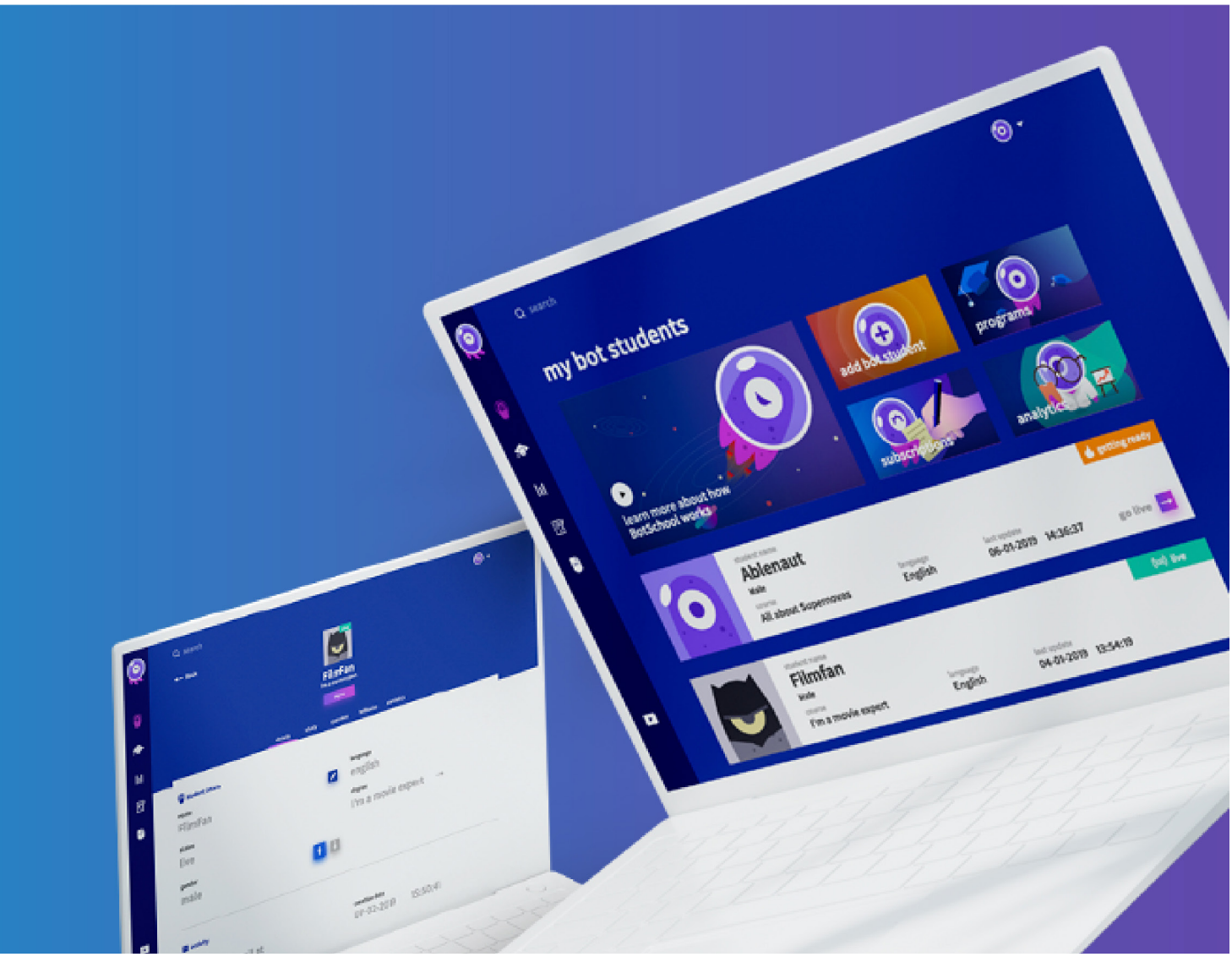
1. adding new knowledge (ex. new training phrases) to big datasets increases the likelihood of intent ambiguity, resulting in erroneous answers or fallbacks;
2. the entire model must be retrained for validation, wasting time at each validation on huge models;
3. there is no possibility of task parallelization, as adding two or more new use cases to the same model must be done in sync.

Altice Labs' BOTSchool, a scalable and multichannel platform that allows any type of business to create, teach and manage its virtual assistants, with support in Natural Language Processing (NLP) and Natural Language Understanding (NLU), reshapes the entire process based in a proprietary technology, which uses dataset splitting, with parallel training of such datasets that may be used by any virtual agent to provide answers.

Introduction

The intelligent virtual assistants market has been growing in recent years. Enterprises looking into customer engagement automation, increase efficiency in internal processes or even include conversational interfaces into their products and services are eager to apply AI to either increase revenues or reduce costs. Actually, despite the vast promise of advantages regarding the use of intelligent virtual assistants [1], the truth is that many projects in the area fail to achieve the desired return on investment (ROI).

This article attempts to define, in a clear way, the typical problems and difficulties associated with investments in intelligent virtual assistants, followed by how some of the capabilities provided by Altice Labs' BOTSchool (https://botschool.ai/home) address and solve those problems, improving ROI and guaranteeing successful investments in the area.



The problems of intelligent virtual assistants investments

Is the problem of the ROI in the value generation or the operational costs associated? Gartner has recently stated that "40% of chatbot/virtual assistant applications launched in 2018 will have been abandoned by 2020" [2]. But despite this number, there is also an expectation that by "2021, (...) one in six customer service interactions (...) will be handled by AI" [2]. This scenario will only be possible if conversation platform vendors manage to tackle some of the problems holding back a positive ROI in the implementation of virtual assistants.

So, what are the costs that are holding back the success of these solutions? They are, mainly, the type of human resources needed to build and maintain the solution, technology limitations associated with large dataset handling and with the solution, unfulfilled expectations or a combination of these factors.

Specialized human resources

According to Gartner, the "market for conversational platforms is moving at a high pace of innovation" with one of the identified trends being the "decreasing dependency on data science skills" [3]. But why is this decrease in dependency on data science skills so important? The decrease occurs mainly due to a shortage of these types of professionals since the demand for them by companies is growing at a faster pace than the availability, as seen in **Figure 1**, from Reuters [4].

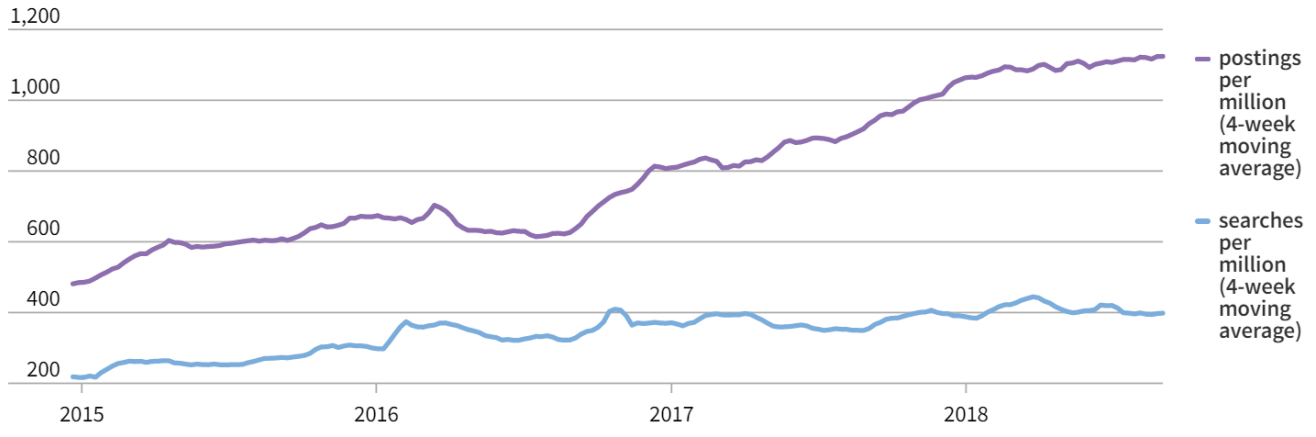


Figure 1 – AI specialist jobs offer and demand

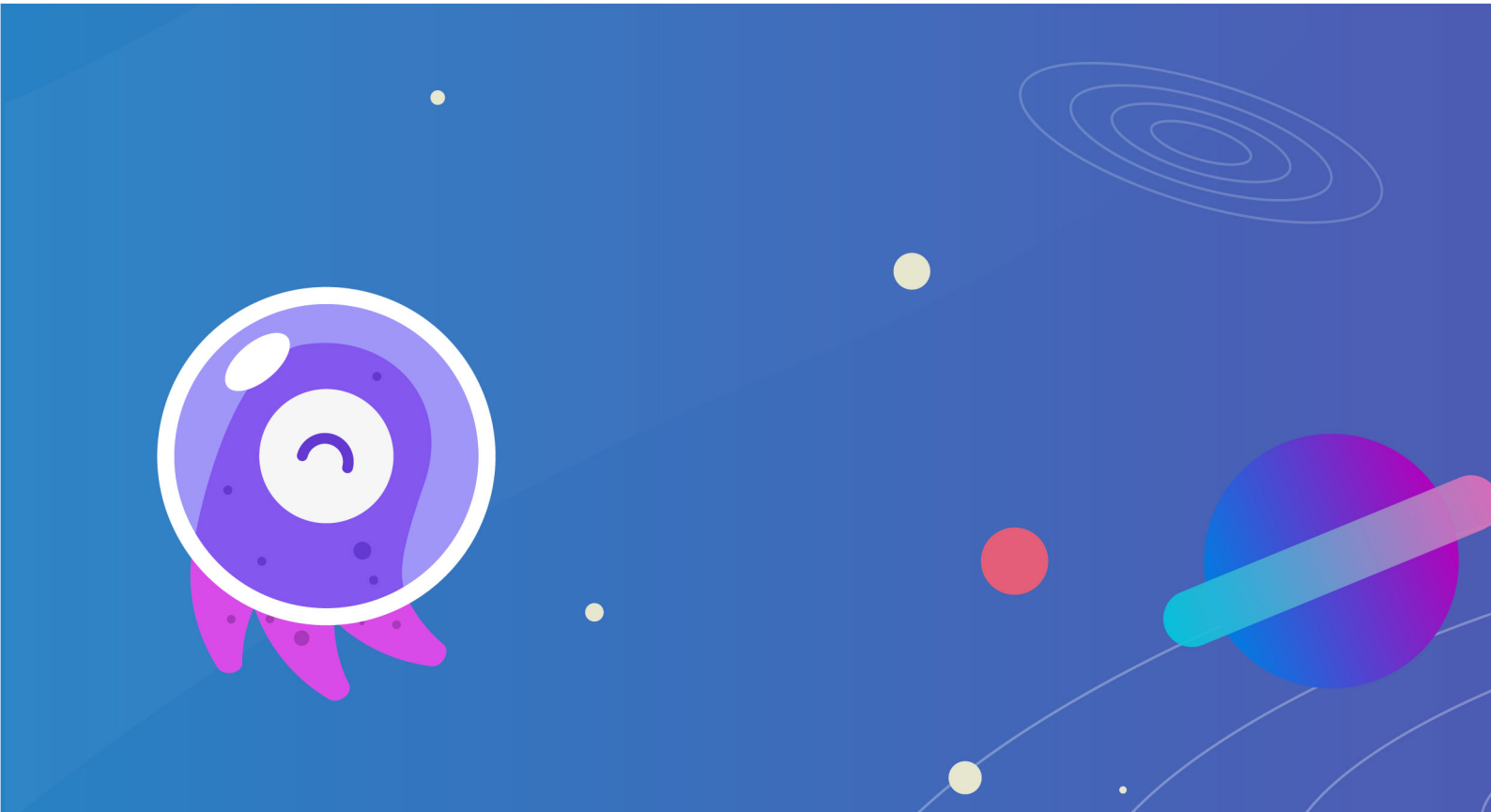
A shortage of this type of professionals leads to high wages, thus contributing negatively to the ROI of any AI-powered virtual assistant project. Failures in the market to address this issue have led to only large enterprises being able to plan positive ROI projects, with lower margins, while still leaving a large part of the market unattended.

Dataset model evolution and conversational flows

The knowledge used by the virtual assistant comes from the training datasets that allow the intention identification from a conversation, technically known as an intent. From that intent, the virtual assistant may provide an answer or ask for further information. In a way, any conversation between a virtual assistant and the customer follows a course that, in reality, is a sequence of conversational interactions.

In many virtual assistant solutions, the conversational flow and the dataset model are tightly coupled, with out-of-the-box solutions having little configurability. These solutions provide few benefits for new customers, being very complex to ensure the contexts of a conversation, which is a human cognitive capability. Yet, more flexible solutions require extensive/heavy code writing to interconnect the datasets with the conversational flows, to accommodate changes, either by functionality expansion, by the establishment of new conversational branches, or others!

Another problem with most of these solutions is that the virtual assistant is a single entity. As a consequence, work is hard to parallelize, as data scientists and developers must work over the same interface, and changes made by an individual may conflict with other changes. As the entire solution is a single entity, validation and testing may only be performed over the model once everyone has finished, as partial work would render the virtual assistant in an inconsistent state. The result is a non-agile development framework that requires effort to synchronize, slowing down the ability of the virtual assistant to evolve and keep up with fast-evolving business needs, compromising its correct contextual interactions.



Huge dataset problems

Also related to the previous problem, invariably when the desired number of interactions grows, the synchronization issues become even worse. Some technological limitations of having a single virtual assistant entity managing an entire dataset also become apparent. When an AI entity is dealing with data trying to identify something, in our case, the intent, it uses matching learning techniques coupled with linguistic knowledge on how humans structure sentences. The problem is that with the increase of the number of patterns/identification of intents, likely there will be the probability of mix up. Ambiguity in intent identification is one of the main reasons why projects in this area fail, as no one wants to have a virtual assistant incorrectly identifying what the customer is discussing.

Just like humans, context can be used to reduce or remove ambiguity. The handling of contexts in traditional systems like the ones described previously increases the complexity of managing the training datasets, amplifying the problems stated when discussing the dataset model evolution. Moreover, each time a change is performed (e.g. new training phrases, add intents or entities extraction) the new models generated have to be validated, and data needs to be trained again, increasing the validation effort/time, making the development a frustrating job, as a tiny change can lead to a systemic shift in the model generated.

BOTSchool Solution

These challenges were some of the drivers that led to the development of the Altice Labs' BOTSchool solution, shaped to be easy to use by non-technical users. BOTSchool integration with external IT systems isn't a programming challenge, and the entire virtual assistant model can be worked on by multiple people, enabling it to easily grow.

Designer Flow

To democratize the implementation of virtual assistants, BOTSchool provides a graphical user interface (GUI) that enables the design of a conversational flow with the contextual support between a virtual assistant with a customer to be handled transparently and easily. BOTSchool allows the design, creation and deployment of virtual assistants without writing a single line of code, making it a codeless, easy-to-use solution.

Looking at the sample designer flow graph from BOTSchool's GUI, in **Figure 2**, BOTSchool allows the definition of intents as the connections in a graph, with the nodes being the desired actions, branching evaluations or the answers to provide after identifying the intent.

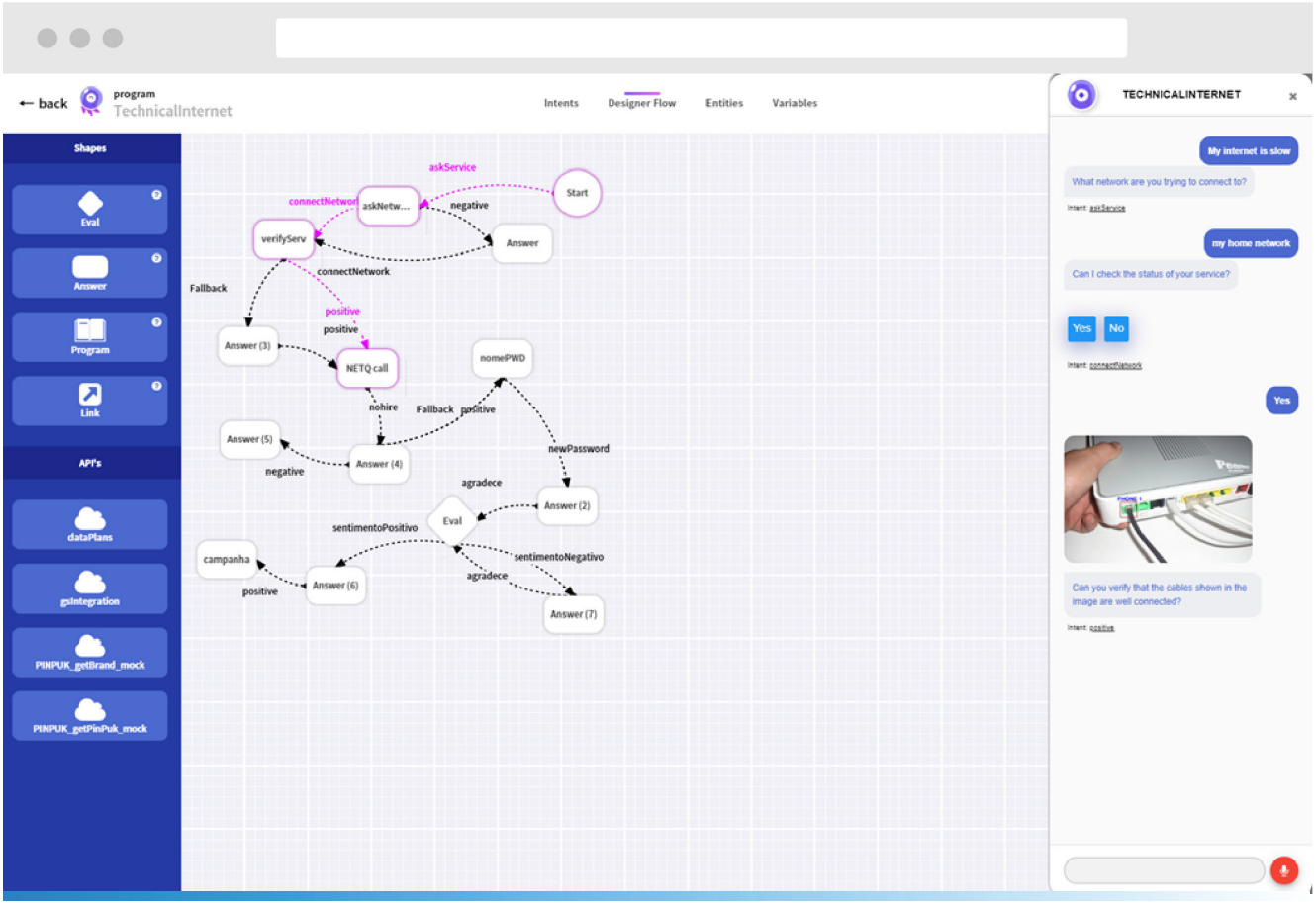


Figure 2 – BOTSchool's designer flow

This kind of interface allows people with domain knowledge, but that aren't data scientists, to develop a fully working intelligent virtual assistant.

APIs for IT integrations

The obvious question from the previous statement is that if no code is needed, how can BOTSchool integrate with IT systems for the more complex customer interactions?

The solution takes advantage of RESTful APIs and other available tools to plan such APIs, like Postman [5]. This API design is performed in the fulfillments area of the BOTSchool GUI, as shown in **Figure 3**. Once this definition is completed, users may drag the special API shape in their designer flow and use it in the context of a detected intent, as well as define the information that is exchanged and the variables that are either retrieved or updated, everything without writing a single line of code.

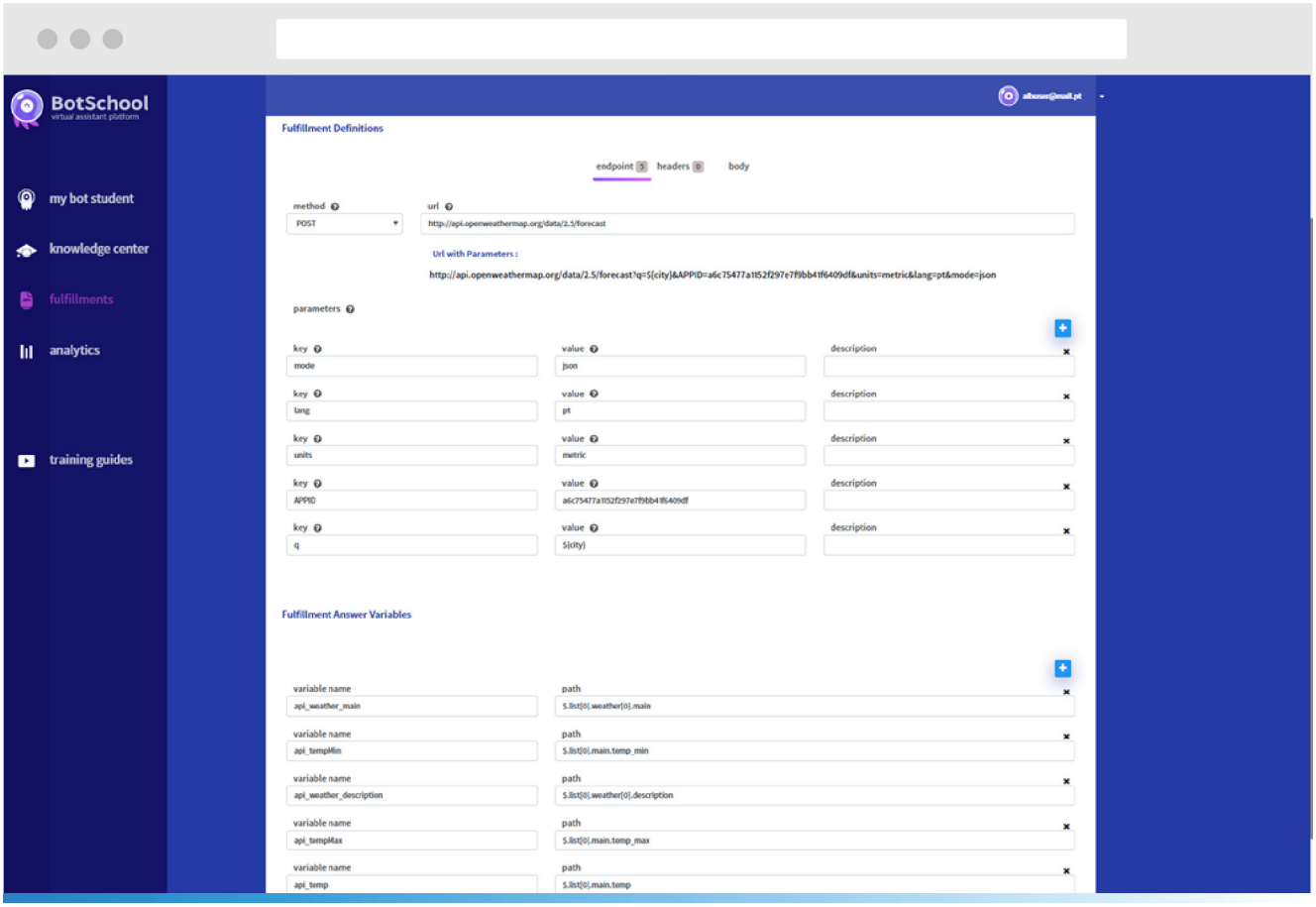


Figure 3 – Fulfillment's API design

Handling the dataset management problems

Dataset segmentation

The first thing that was needed was to change the way the training datasets were handled. BOTSchool's solution has been inspired by how humans learn and are taught. There is a clear separation between students (in our case, bots/virtual assistants) and knowledge. The knowledge that is available on books or information libraries allows individuals to specialize and evolve in various topics. This specialization results in the supporting of multiple smaller training datasets that in BOTSchool are known as programs. Using a school metaphor, a program is similar to a subject, just like Mathematics, English, Chemistry, etc.

From the AI virtual assistant creation standpoint, it is possible to segment or separate multiple domains and sub-domains into different programs. Each program can be worked independently by a single "teacher", the BOTSchool users, just the same way there are specialists in certain areas of knowledge in a school environment. There is no need for all this information to be clumped up together in a single dataset.

The "curriculum" of a program can be worked independently. Each program has its individual designer flow, provides a simple test virtual assistant in the interface where users can test what they are designing and, finally, programs may be in one of two important status: published or unpublished. Once a program is "published", it becomes available, and it may be shared or associated with any existent virtual assistant, without the need to generate the model that supports the program again.

This segmentation into programs and the possibility of, independently, developing a program enables the setup of projects with clear responsibilities across a team of users. The parallelization of work allows faster development and deployment of new versions of the virtual assistants. Program segmentation also addresses the problem of the time needed to teach huge datasets. Fine-tuning a program or segment of the dataset doesn't require the entire dataset to be trained and to be tested.

So, how can a virtual assistant student learn or is able to use all this knowledge?

Managing the virtual assistant knowledge

Unlike other solutions, BOTSchool's virtual assistant entities are separated from the knowledge they use, which means that the virtual assistant student is capable of learning from existing programs. In BOTSchool, users decide what knowledge a particular student shall learn. **Figure 4** shows how the interface allows users to teach specific programs to their students, with the learn operation, or even how to remove the knowledge of a program with the unlearn operation.

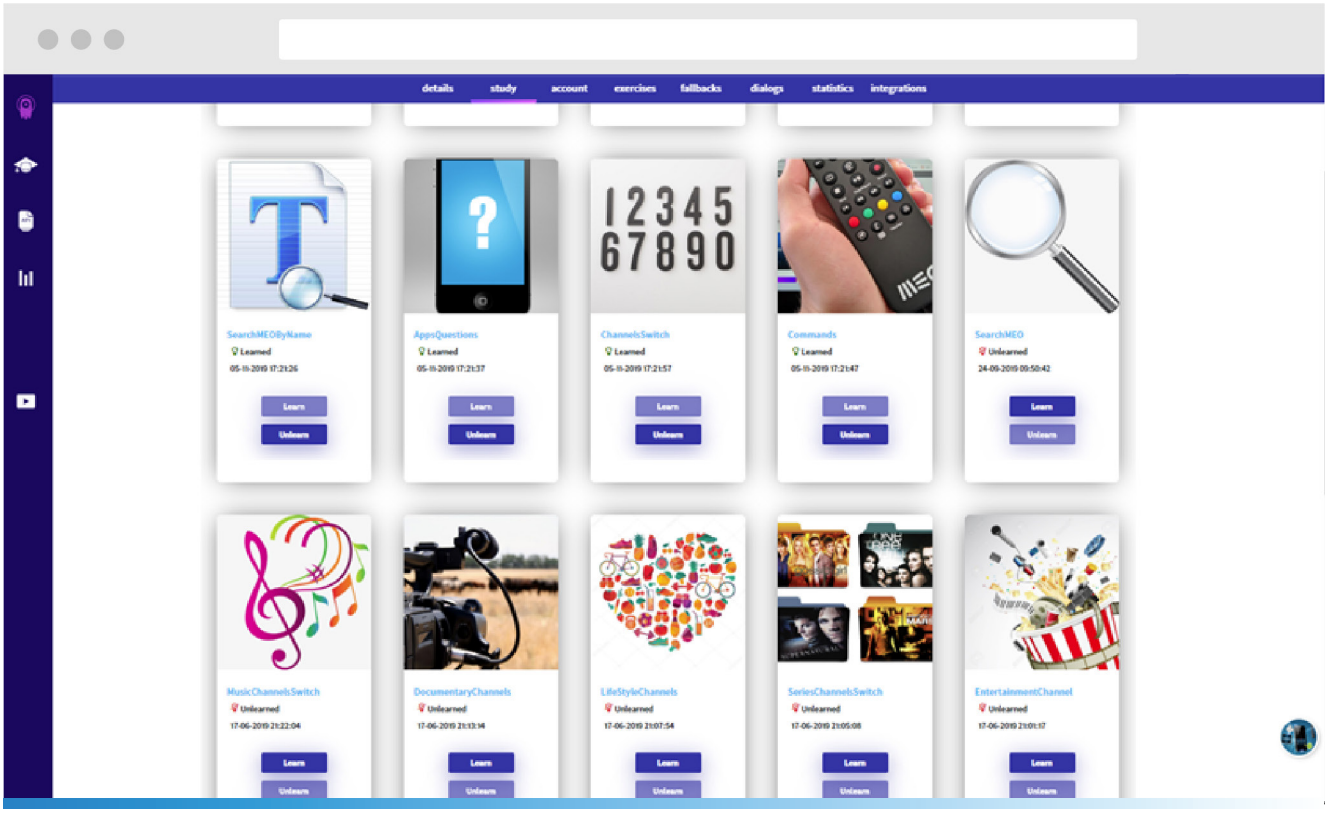


Figure 4 – Student learn and unlearn interface

A student may learn or unlearn each program individually. It's not necessary to wait for all programs to be fully developed and taught to finally teach it to a student. What this means is that, if for some reason it is necessary to update two programs, it is possible to teach first one of the new programs that were finished earlier, without having to wait for the other program update.

Another important feature is that, while programs are being updated to a new version, each student holds the last taught program. Consequently, it is possible to develop and test new versions of programs without affecting the virtual assistant in production.

Ambiguity reduction

Another relevant functionality of BOTSchool relies on the natural language processing (NLP) engine that processes utterances (a phrase from the user), by testing them against all existing models/programs associated with a virtual assistant, much like a broadcast operation that shall receive individual results from each model. This process is exemplified in **Figure 5**.

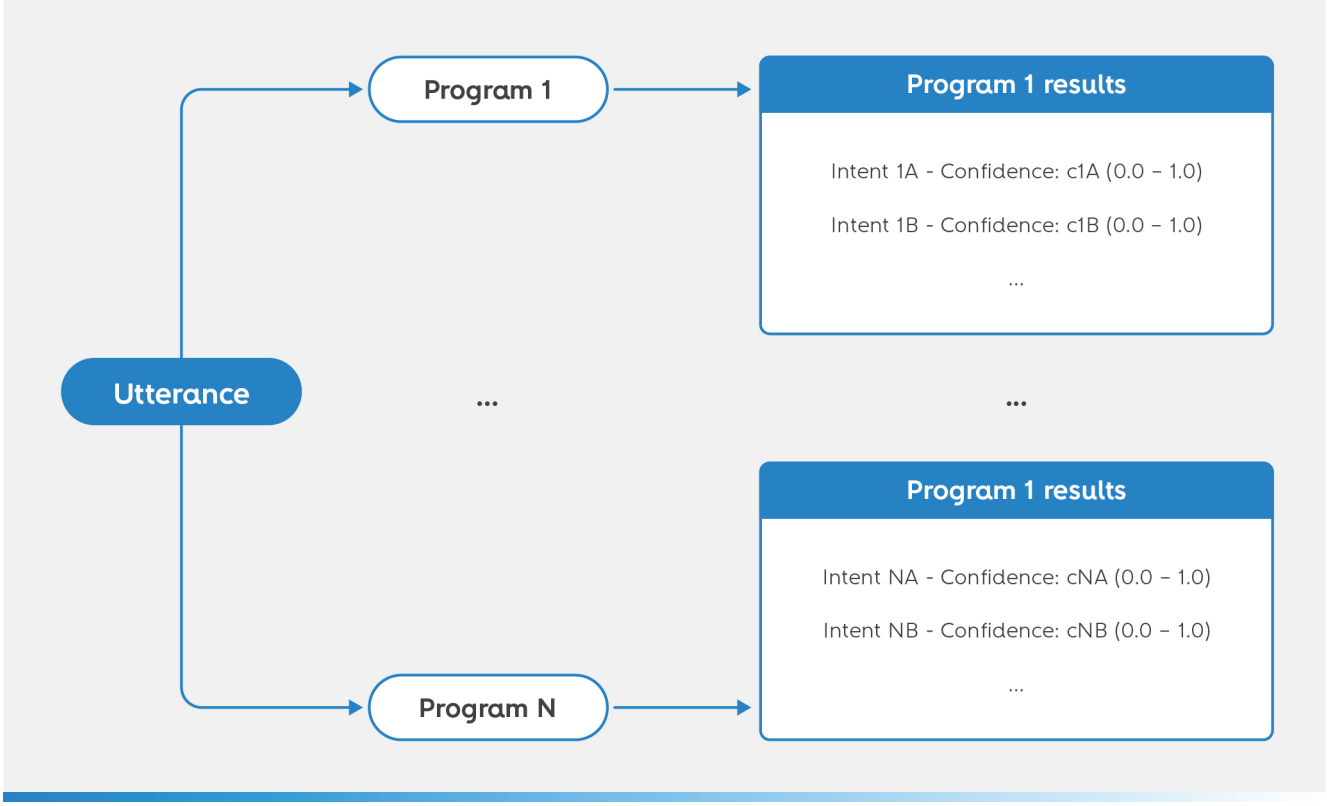


Figure 5 – Broadcasting the utterance to test it with all programs

The method will then determine the most accurate intent, by testing and pruning the results, effectively eliminating or drastically reducing ambiguity, before picking the correct intent. The first test is the elimination of all intents in the results which have a confidence level below the defined threshold for their corresponding program. This process may leave a single valid intent, or more often, a set of intents which we may classify as "intent candidates", as they are the ones who have a likelihood of being the correct intent to which the utterance maps into.

Just like a feedback loop system, BOTSchool uses past information to determine future output, and all intent candidates go through a second level of tests.

In the flow shown in **Figure 6**, F(x) is a model algorithm that is used to select the best accurate intent. Given the intent candidates, check which of them are from the same program of the previous intent (if any has been identified so far) and remove those who aren't. In case no intent candidates are from the previous intent's program, keep all candidates. If more than one intent remained from the previous step, select the highest confidence intent, which shall then be the new intent, and update the program of previous intent.

These simple tests can drastically reduce ambiguity when compared with a non program separation of the dataset. But why does this happen? Human conversations are naturally contextual. Once conversations start on a specific topic, it is natural for them to continue to be related to the past topic or context for a certain number of interactions or exchanges. By using contexts, where contexts are defined naturally by the program, it is possible to reduce ambiguity, as intents within the same context have a higher probability of being the correct ones. This assumption does not invalidate the possibility of changing topics in the future, once the conversational exchange reaches its natural end. For example, once a conversation on a topic ends, such as a technical problem with a router, it is possible to shift to a new topic, such as doubts regarding charges in a bill. But while the conversation is centred on the technical problem, all intents are more likely to be related to such topic.

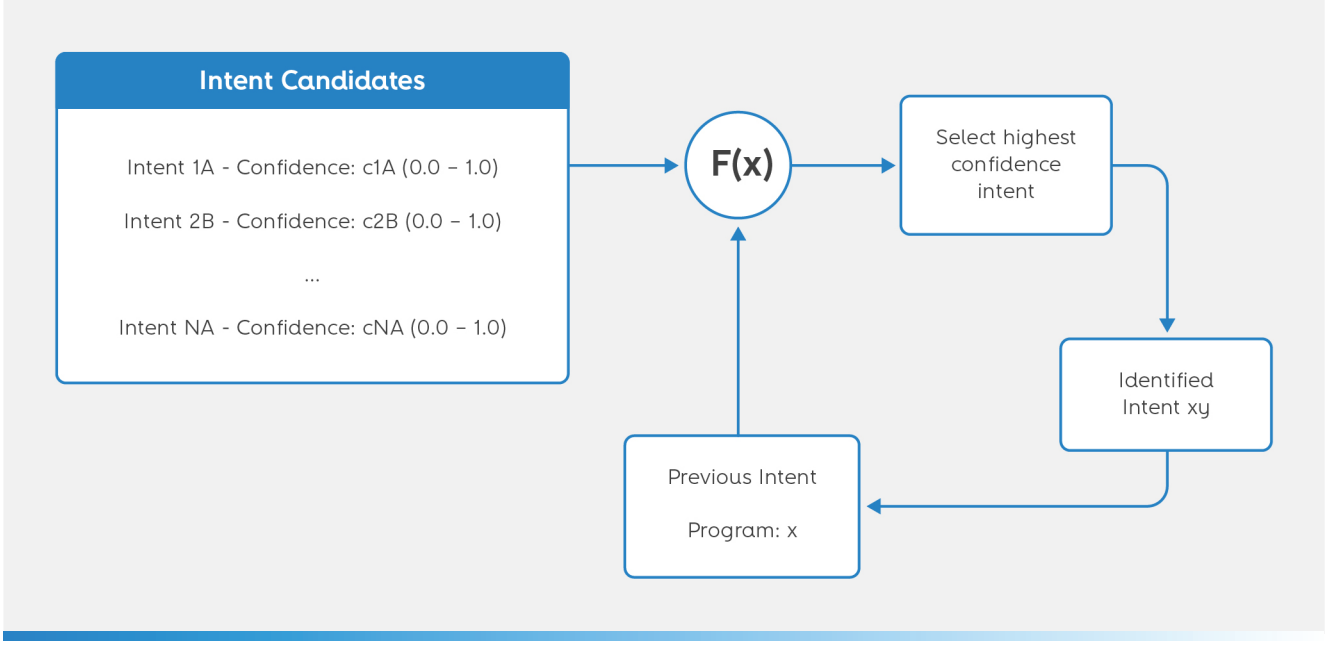


Figure 6 – Using previous program x to help select an intent candidate

Finally, the last step shall use identified entities to give more relevance to intent candidates that hold more identifiable entities. The following illustrative example aims to explain how this step is achieved: Consider a weather query interaction "*Can you tell me the weather forecast in London for tomorrow*". This sentence has two entities: location (London) and date (tomorrow). Also, consider that the bot uses a program with the following training phrases in two different intents:

- Intent 1: "Please tell me the weather for <location>."
- Intent 2: "Can you tell me the weather for <location> for <date>?"

It is expectable that both intent 1 and intent 2 have high confidence. However, notice that intent 2 has two entities in the definition of one of its training phrases, which the utterance also holds. So, when comparing the likelihood of the utterance belonging to intent 1 or intent 2, the decision may be based on the fact that intent 2 has two matched entities versus only one in intent 1. Consequently, intent 2 has more matched entities and as such, should be considered the correct entity.

This pipeline solution, shown in **Figure 7**, is able to significantly reduce intent ambiguity while providing a logical structure to the planning and execution of AI-powered virtual assistant projects.

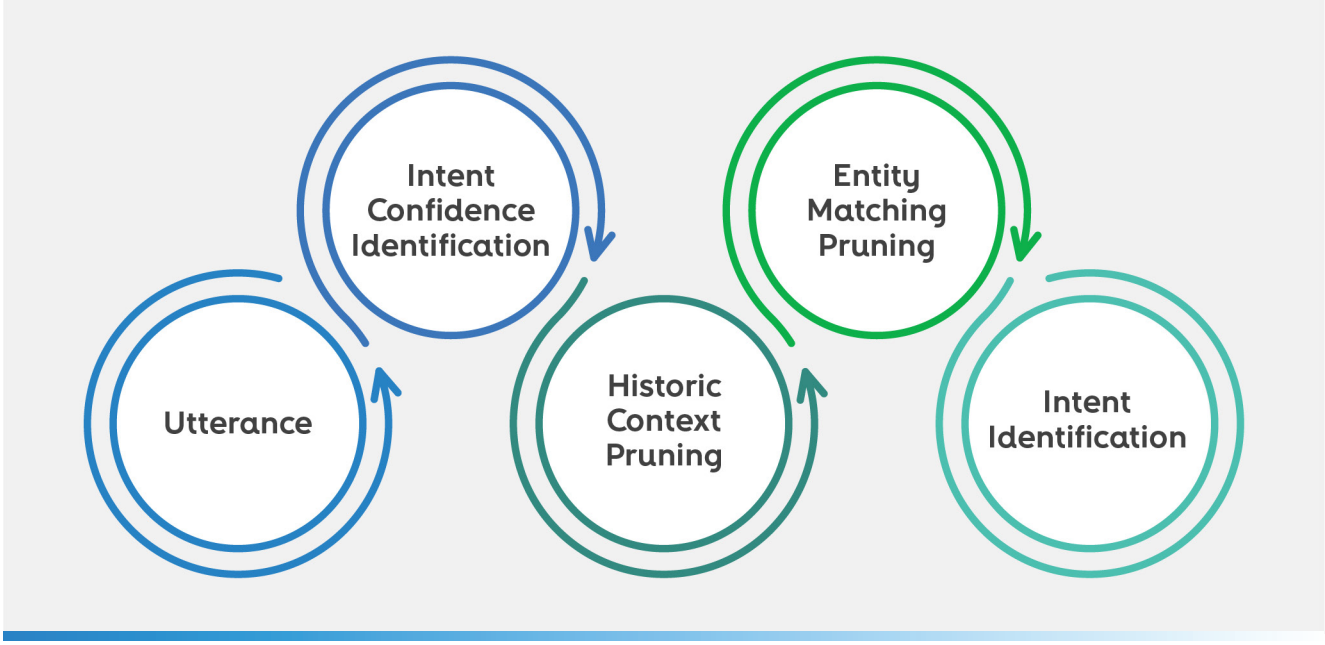


Figure 7 – User utterance processing pipeline

Wrap-up

BOTSchool was developed following a set of principles that place the business success of customers as the primary goal. The main problems holding back the adoption of AI-powered virtual assistants in companies are the difficulty in obtaining a positive ROI which has been mainly hindered by such projects. Altice Labs highlights two main cost drivers that prevent wide-scale adoption of this technology: specialized human resources and project management issues resulting from technological limitations. To overcome these problems, Altice Labs focused on changing the way virtual assistants are created and managed (patent pending), driving costs down, improving customer experience and increasing the prospects of a positive ROI.

BotSchool
virtual assistant platform

https://botschool.ai/home

References

J. M. Sousa, J. M. Monteiro, A. M. Martinho, and S. S. Furão, "Virtual Assistants, the future digital interface," *InnovAction #3*, pp. 64–73, 2018.

M. Revang, B. Elliot, B. Manusama, and A. Mullen, "Market Guide for Virtual Customer Assistants," *Gartner*, 2019.

B. Manusama, A. Mullen, M. Revang, A. Lee, and V. Baker, "Market Guide for Conversational Platforms," *Gartner*, 2019.

A. Saphir, "As companies embrace AI, it's a job-seeker's market," Reuters, 2018. [Online]. Available: https://www.reuters.com/article/us-usa-economy-artificialintelligence/as-companies-embrace-ai-its-a-job-seekers-market-idUSKCN1MP10D.

Postman Inc., "Postman," 2019. [Online]. Available: https://www.postman.com/.

Contacts

Address

Rua Eng. José Ferreira Pinto Basto
3810 - 106 Aveiro (PORTUGAL)

Phone

+351 234 403 200

Media

contact@alticelabs.com
www.alticelabs.com