What I wish I knew before starting my first Chatbot project

LAPPIS

Jan 14 · 4 min read

For the past 14 months, we have been working on a FLOSS FAQ chatbot project in collaboration with the Ministry of Culture in Brazil. We have come a long way testing, evaluating chatbot frameworks, NLP techniques, and user experience with chatbots, doing applied research while developing TAIS (chatbot in Brazilian Portuguese), our chatbot for the Law of incentive to the culture, known as “LEI Rouanet”, and its service, “SALIC”. Chatbot presence is growing in the private sectors, but it is still rare in government agencies.

Even if chatbots are trending, the documentation to guide its project and decision making is most of the times not very clear and is dispersed. There is a quite broad range of option to chatbot projects, from services such as Dialog Flow, Facebook, Watson, LUIS, to open sources solutions, such as Rasa, Botkit, Parlai, among others.

We have made many mistakes along the way. Mainly due to the lack of mature documentation and community still emerging. Here we list what we wish we have known before starting our chatbot project.

Chatbot is an ecosystem: a chatbot platform is the integration of several components. Some solutions provide only a subset of these components and it makes the comparison between platforms challenging. The components of a complete chatbot platform is despited in Fig. 1.

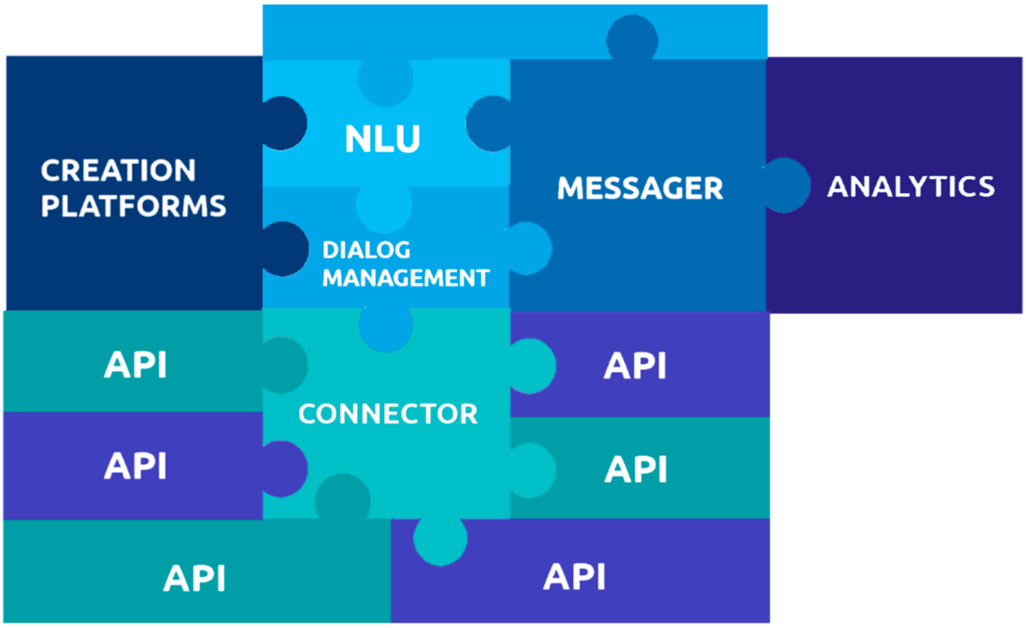

Fig1. Chatbot Platform: components that, integrated, form a chatbot platform.

- **Fullstack team**: developing a chatbot demands a big set of abilities, differently from most of the software projects. It requires software engineers, DevOps, dialog designer or chatbot scriptwriter, data scientist, business strategist, and so forth. They need to work closely and make decisions together. Figure 1 shows an ideal chatbot team. The absence of one of these abilities in the team jeopardizes the chatbot quality (We have made this mistake).

Fig.2 — Ideal Fullstack chatbot team

- **Algorithms impact a lot the user experience**: how the Natural Language Processing is done, the techniques used, impact the ability of the chatbot to understand user intentions. The strategy/policy chosen to deal with dialogs (menus, guided flows, learning from example conversations) impacts enormously the user experience (UX). Each chatbot platform available has one of several strategies to process the conversation. Make sure the Chabot solution you choose it meets your project requirements.

- **Compliance with data regulations**: ay attention to how your organization deals with data regulations, such as the General Data Protection Regulation (GDPR), not all solutions comply with these regulations.

- **Chatbots take time to mature**: you don't want your chatbot to become a lousy meme! So, understand that a chatbot project is iterative and incremental. It takes time to have a cool chatbot ;P Make sure all stakeholders understand that! Controlled chatbot presentations make clients dazzle, and they tend to pressure the team to put it as soon as possible in production. Take time to test and validate your chatbot! The beta test is the way to go (a more controlled ambient and engaged community with mostly constructive feedbacks).

- **Corpus in other Languages than English —** it is a big problem! So, try to find communities in your country working with chatbots, and share training datasets :P

- **Analytics —**how users interact with chatbots tends to be very different from how you thought they would behave! User interaction data is the most precious asset of a chatbot. It gives you the real interests/doubts of your customer, and you can have data-driven business strategies! Additionally, it tells your team where the chatbot is making mistakes and where to focus improvements.

- **Continuous Delivery —** enabling Continuous delivery allows to instantiate a chatbot, and test it very often. It is important for users to perceive a better and more accurate interaction with the chatbot continuously , increase of its knowledge; Think about Continuous deploy, DevOps, use of dockers.

We have chosen to have a FLOSS chatbot, completely on-premise. We have chosen Rasa Rasa-Core and Rasa-NLU as our chatbot creation service and Rocket.chat as our distribution service (how to interact with users). We also used elasticsearch + kibana as our analytics. Our complete architecture is depicted in Figure 2. It is under the open-source license GPL3 and all project source code, documentation is available in https://github.com/lappis-unb/tais.

Fig.3 — Our Chatbot architecture avaliable in https://github.com/lappis-unb/rasa-ptbr-boilerplate

We will detail some of our lesson learned and some results, technical issues in future posts.

Chatbots Rasa Rocket Chat Egovernment Machine Learning