

**Customer Support Bot**

Team :

BELBACHIR SAID

AIT ABDELMALK HASNAA

Supervised by :

ABDELHAK MAHMOUDI

**Master Ingénierie de Données et**

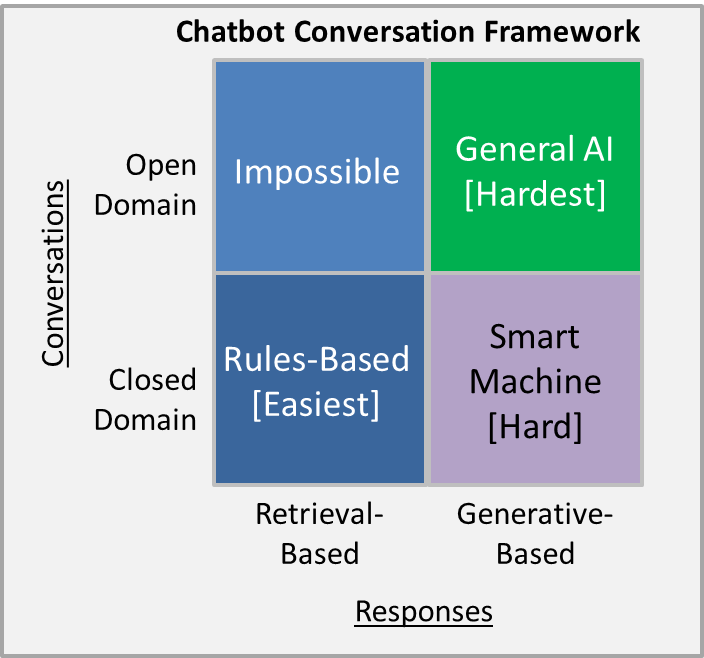
**Développement Logiciel**

Introduction:

Chatbots, also called Conversational Agents or Dialog Systems, are a hot topic. Microsoft is making big bets on chatbots, and so are companies like Facebook (M), Apple (Siri), Google, WeChat, and Slack. There is a new wave of startups trying to change how consumers interact with services by building consumer apps like Operator or x.ai, bot platforms like Chatfuel, and bot libraries like Howdy’s Botkit. Microsoft recently released their own bot developer framework.

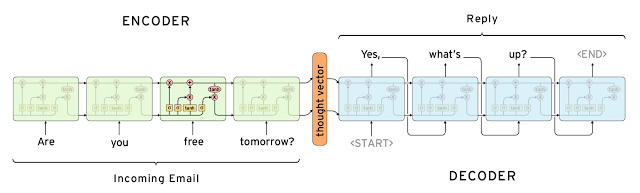
Many companies are hoping to develop bots to have natural conversations indistinguishable from human ones, and many are claiming to be using NLP and Deep Learning techniques to make this possible. But with all the hype around AI it’s sometimes difficult to tell fact from fiction.

Models :



**Retrieval-based models (easier)** use a repository of predefined responses and some kind of heuristic to pick an appropriate response based on the input and context. The heuristic could be as simple as a rule-based expression match, or as complex as an ensemble of Machine Learning classifiers. These systems don’t generate any new text, they just pick a response from a fixed set.

**Generative models (harder)** don’t rely on pre-defined responses. They generate new responses from scratch. Generative models are typically based on Machine Translation techniques, but instead of translating from one language to another, we “translate” from an input to an output (response).

[](http://www.wildml.com/wp-content/uploads/2016/04/nct-seq2seq.png)

Both approaches have some obvious pros and cons. Due to the repository of handcrafted responses, retrieval-based methods don’t make grammatical mistakes. However, they may be unable to handle unseen cases for which no appropriate predefined response exists. For the same reasons, these models can’t refer back to contextual entity information like names mentioned earlier in the conversation. Generative models are “smarter”. They can refer back to entities in the input and give the impression that you’re talking to a human. However, these models are hard to train, are quite likely to make grammatical mistakes (especially on longer sentences), and typically require huge amounts of training data.

Deep Learning techniques can be used for both retrieval-based or generative models, but research seems to be moving into the generative direction. Deep Learning architectures like [Sequence to Sequence](http://arxiv.org/abs/1409.3215) are uniquely suited for generating text and researchers are hoping to make rapid progress in this area. However, we’re still at the early stages of building generative models that work reasonably well. Production systems are more likely to be retrieval-based for now.

#### Long vs. Short Conversations

The longer the conversation the more difficult to automate it. On one side of the spectrum are **Short-Text Conversations (easier)** where the goal is to create a single response to a single input. For example, you may receive a specific question from a user and reply with an appropriate answer. Then there are **long conversations (harder)** where you go through multiple turns and need to keep track of what has been said. Customer support conversations are typically long conversational threads with multiple questions.

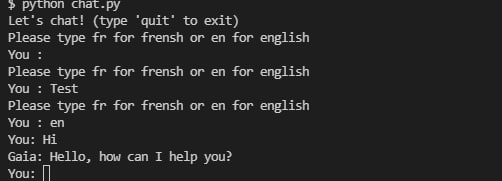
#### Open Domain vs. Closed Domain

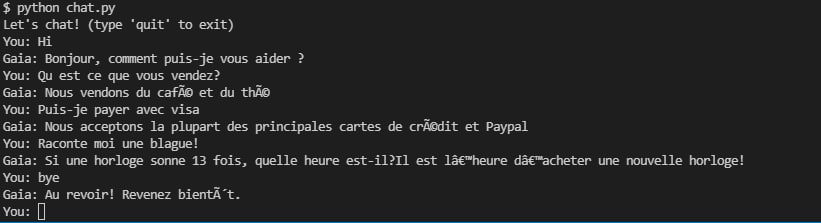
In an **open domain (harder)** setting the user can take the conversation anywhere. There isn’t necessarily have a well-defined goal or intention. Conversations on social media sites like Twitter and Reddit are typically open domain – they can go into all kinds of directions. The infinite number of topics and the fact that a certain amount of world knowledge is required to create reasonable responses makes this a hard problem.

In a **closed domain (easier)** setting the space of possible inputs and outputs is somewhat limited because the system is trying to achieve a very specific goal. Technical Customer Support or Shopping Assistants are examples of closed domain problems. These systems don’t need to be able to talk about politics, they just need to fulfill their specific task as efficiently as possible. Sure, users can still take the conversation anywhere they want, but the system isn’t required to handle all these cases – and the users don’t expect it to.

Realisation

Our project : Gaia Customer Support Bot is an exemple of the implementation of chat bots with closed domain on both languages English and French for an online coffee store : the user has the choice between those two languages then the scenario of conversation :the greeting ; the user can ask for the products on sell, the time of delivery and the paying method then as a funny part he can also ask for a jock ☺:





Next step :

Adding Arabic and Darija .