# Building chatbot without cloud

## JS Meetup luxembourg

Florent FAUCHILLE

*September, 12th 2018*

# ELIZA

- ELIZA (creator Joseph Weizenbaum) (1966)
- Parody of a human psychotherapist

*USER*: Men are all alike.
*ELIZA*: IN WHAT WAY
*USER*: They're always bugging us about something or other.
*ELIZA*: CAN YOU THINK OF A SPECIFIC EXAMPLE
*USER*: Well, my boyfriend made me come here.
*ELIZA*: YOUR BOYFRIEND MADE YOU COME HERE
*USER*: He says I'm depressed much of the time.
*ELIZA*: I AM SORRY TO HEAR YOU ARE DEPRESSED

# ELIZA

…

*USER*: My father is afraid of everybody.

*ELIZA*: WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER

*USER*: Bullies.

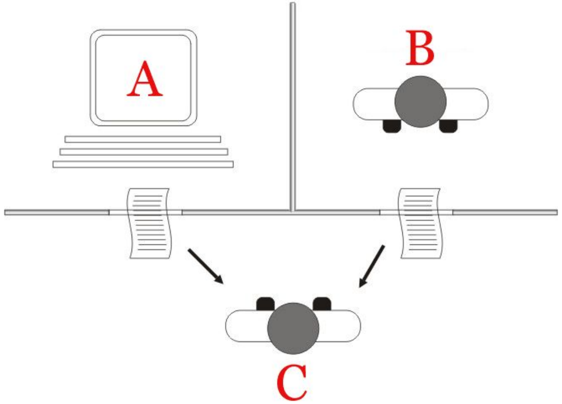
ELIZA: DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

- People became deeply emotionally involved with the program
- Weizenbaum tells the story of his secretary who would ask Weizenbaum to leave the room when she talked with ELIZA
- When he suggested that he might want to store all the ELIZA conversations for later analysis, people immediately pointed out the privacy implications

*Source: https://web.stanford.edu/class/cs124/lec/chatbot.pdf*

# PARRY

- PARRY (1968)
- Same structure as ELIZA
- Mimics a very specific personality, enabling PARRY to be the first chatbot that passes Turing test
- ELIZA met PARRY
  https://web.stanford.edu/class/cs124/lec/chatbot.pdf



https://en.wikipedia.org/wiki/Turing_test#/media/File:Turing_test_diagram.png

# ELIZA architecture is too limited...

- Chatbot that are rule-based
- Scope of conversation has to be narrow
- Some rules are becoming nearly impossible with grammar due the the natural language ambiguity

# Chatbot's challenges

- Recognize the user's intent and possibly extract information from it (Locations, dates, company names …)
- Manage who has control of the conversation (bot or user)

# NLU / NLP

- Natural Language Understanding/ Processing
- Instead of writing static rules on text, NLU is based on training examples of user sentences with intents
- A model is trained from all labeled user sentences
- Once trained, the model can predict user sentence’s intents that are not on training sets
- Much wider chatbots has been implemented taking advantages of DB like twitter or movie subtitles

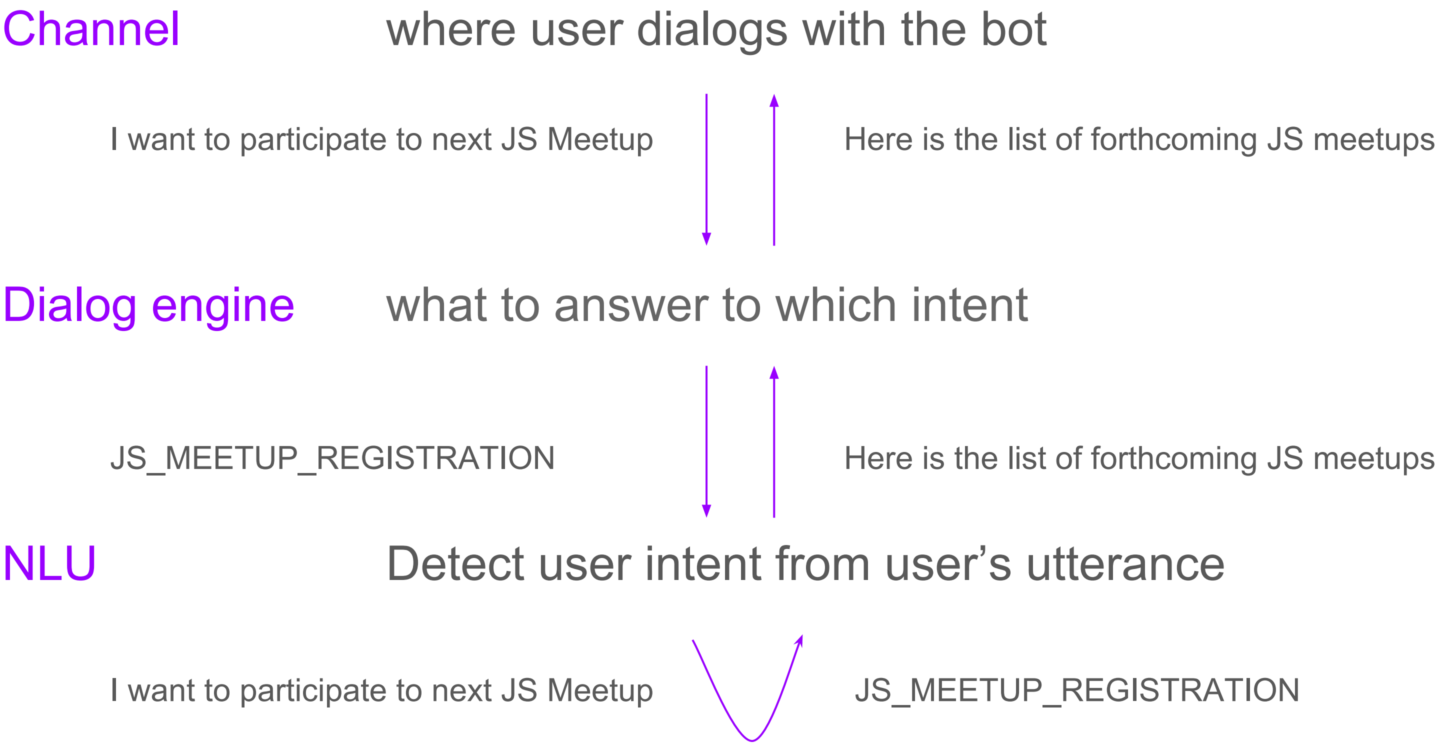
# NLU API’s

- There are no needs to implement a NLU API as of today
- Many proven solutions already exists both cloud-based and on premises
- LUIS (Microsoft), Dialog Flow (Google), IBM Watson …
- RASA-NLU, Snips-NLU

# JS Meetup chatbot

- [https://github.com/ffauchille/js-meetup-lu-chatbot](https://github.com/ffauchille/js-meetup-lu-chatbot)

# Chatbots logic



**Channel** where user dialogs with the bot

I want to participate to next JS Meetup

Here is the list of forthcoming JS meetups

**Dialog engine** what to answer to which intent

JS_MEETUP_REGISTRATION

Here is the list of forthcoming JS meetups

**NLU** Detect user intent from user’s utterance

I want to participate to next JS Meetup

JS_MEETUP_REGISTRATION

# JS Meetup chatbot’s layers

| | |
|---|---|
| Channel/Chat | forked from BotFramework’s WebChat |
| Directline | api to route channel’s to dialog engine |
| Dialogs | Microsoft bot builder |
| NLU | Rasa-nlu API |
| MeetupAPI | to get info on meetups |

*All open source and deployable on premise*

# NLU Trainer

- NodeJS stack
- Nlu-trainer TypeScript, React, Redux, RxJS
  https://github.com/ffauchille/nlu-trainer

- Nlu-trainer-api TypeScript, Restify, RxJS, MongoDB
  https://github.com/ffauchille/nlu-trainer-api

# Microsoft bot builder

- Microsoft Bot builder [https://github.com/Microsoft/BotBuilder/tree/master/Node](https://github.com/Microsoft/BotBuilder/tree/master/Node)
- Open source
- Many chat’s channels (Facebook Messenger, slack, twilio …) [https://docs.microsoft.com/en-us/azure/bot-service/bot-service-manage-channels?view=azure-bot-service-3.0](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-manage-channels?view=azure-bot-service-3.0)

# Some desirable chatbot’s features

- Recognize user intent with a configurable certainty
- Answer with rich content
- Manage dialog’s states
- User profiles
- Ask less questions as possible
- Not being stuck in a loop
- ...

# Bot builder main objects

| | |
|---|---|
| Session | containing user, channel and conversation’s info |
| Recognizer | calls NLU Api and get intents and entities |
| Connector | adapts payload based on channel’s type |
| Bot | handles dialog stack and storage |

# Bot builder conversations

| | |
|---|---|
| Step | a single Q&A |
| Dialog | list of steps |
| Conversation | list of dialogs |

Dialogs are stacked and conversation ends when no dialogs are left in the dialog stack

Data are persisted on different levels

- User level
- Conversation level
- Dialog level

# Live code

Let’s create a new dialog

- Create intent and example in the NLU trainer
- Register a new dialog matching the created intent
- Redeploy dialog api
- Test in WebChat

# Next steps

- Voice connector using Snips
- Compare NLU models in NLU trainer
- More connectors => gitter.io

# Sources

- Chatbot’s Design principles
  https://docs.microsoft.com/en-us/azure/bot-service/bot-service-design-principles?view=azure-bot-service-3.0
- RASA-nlu
  https://rasa.com/docs/nlu/0.13.2/

If you want to go deeper

- Dialog Systems and Chatbots
  https://web.stanford.edu/~jurafsky/slp3/25.pdf
- ELIZA and PARRY
  https://web.stanford.edu/class/cs124/lec/chatbot.pdf
- Speech and Language processing
  https://web.stanford.edu/~jurafsky/slp3/