RASA

NLU Tutorial

Emanuele Bastianelli
e.bastianelli@hw.ac.uk

Before we start

- Clone

```
git clone
https://github.com/HWUConvAgentsProject/
CA2020_instructions.git
```

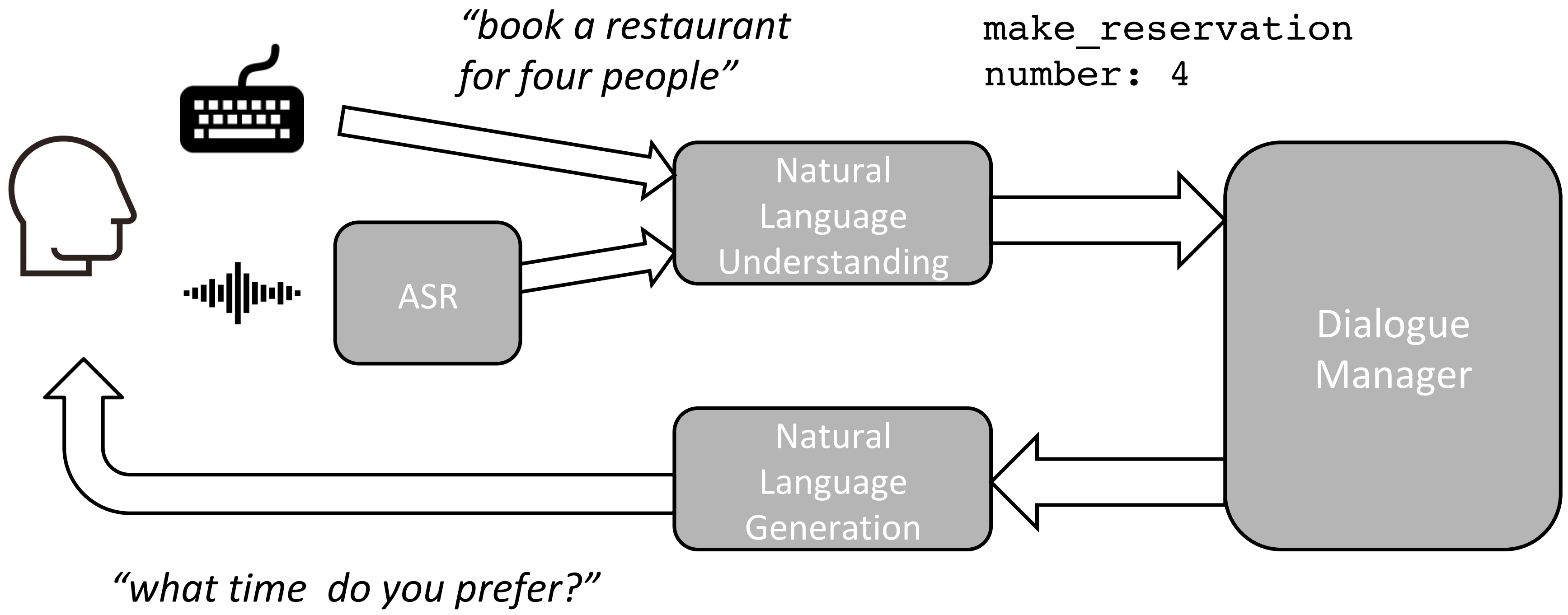
- Update

```
git pull
```

What we are going to do today

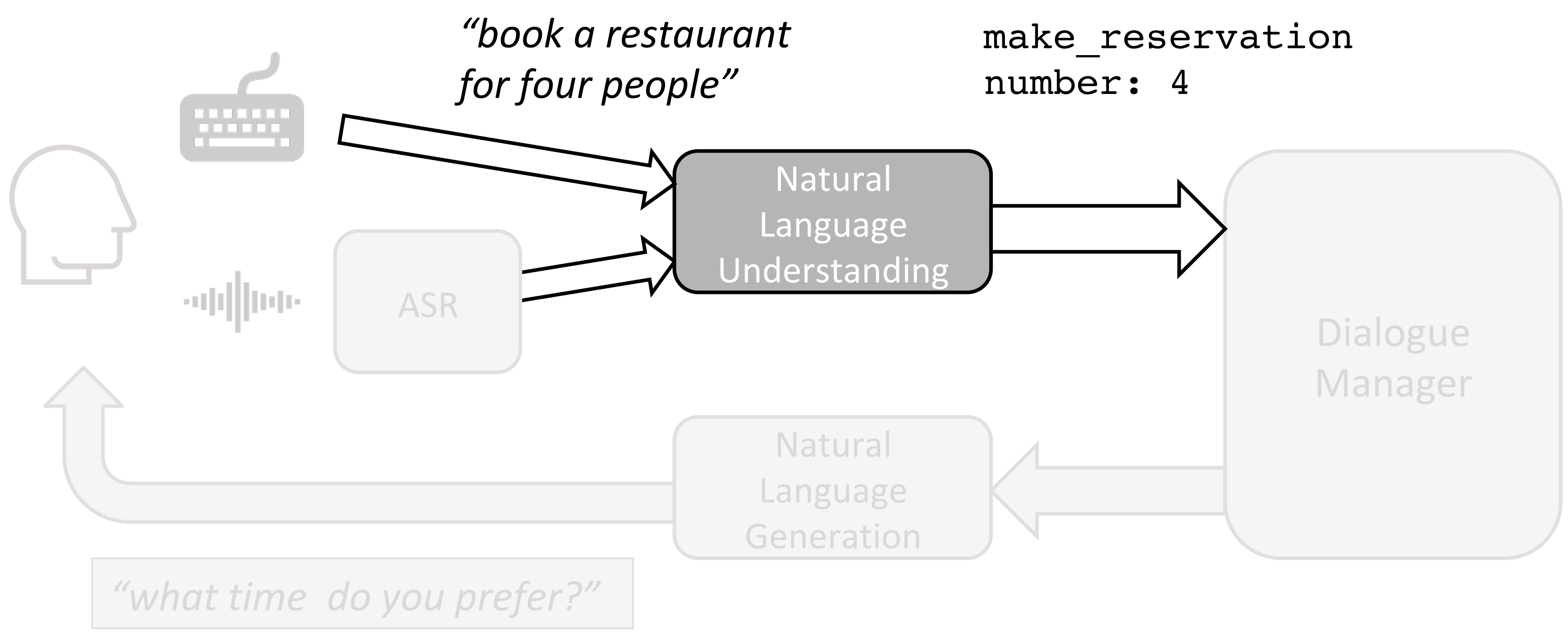
- Create RASA Project
- Understanding RASA input format
- Building and training RASA NLU pipelines
- Testing RASA NLU pipelines
- Understanding RASA output format

Dialogue Systems



"book a restaurant for four people"
make_reservation
number: 4
ASR
Natural Language Understanding
Dialogue Manager
Natural Language Generation
"what time do you prefer?"

Dialogue Systems



"book a restaurant for four people"
make_reservation
number: 4
Natural Language Understanding
ASR
Dialogue Manager
Natural Language Generation
"what time do you prefer?"

Recap: intent and slots

- What's NLU
  - What's an **intent**

$$\underbrace{\textit{I'd like to book a table in New York with Italian cuisine}}_{\texttt{make\_reservation}}$$

  - What's an **entity** (or slots, or arguments…)

$$\textit{I'd like to book a table in } \underbrace{\textit{New York}}_{\texttt{location}} \textit{ with } \underbrace{\textit{Italian}}_{\texttt{cuisine}} \textit{ cuisine}$$

Before we start 2

- Let’s test if everything is ok
  - `rasa`

- if not ...

Before we start 2

- Let’s install RASA
  1. `conda activate Alana`
  2. `pip install --no-cache-dir rasa`
  3. `pip install --no-cache-dir –r` **`requirements.txt`**
  4. `rasa`

Create a RASA project

- Create a RASA project
  1. `mkdir rasa-nlu-tutorial`
  2. `cd rasa-nlu-tutorial`
  3. `rasa init --no-prompt`
  4. `rasa shell`

Create a RASA project

- Directory structure:
  - `data/`
  - `models/`
  - `__init__.py`
  - `actions.py`
  - `config.yml`
  - `credentials.yml`
  - `domain.yml`
  - `endpoints.yml`

Create a RASA project

- Directory structure:
  - `data/`
  - `models/`
  - `__init__.py`
  - `actions.py`
  - `config.yml`
  - `credentials.yml`
  - `domain.yml`
  - `endpoints.yml`

Create a RASA project

- Directory structure:
  - `data/`
    - `nlu.md`
    - `stories.md`
  - `models/`
  - `__init__.py`
  - `actions.py`
  - `config.yml`
  - `credentials.yml`
  - `domain.yml`
  - `endpoints.yml`

Create a RASA project

- Directory structure:
  - `data/`
    - `nlu.md`
    - `stories.md`
  - `models/`
  - `__init__.py`
  - `actions.py`
  - `config.yml`
  - `credentials.yml`
  - `domain.yml`
  - `endpoints.yml`

The bAbI dataset

- https://research.fb.com/downloads/babi/
- Using the (6) dialog bAbI tasks
  - Dialogues about restaurant booking
- We're using the dialog-babi-task5-full-dialogs-trn.txt file for this tutorial, but
  - `CA2020_instructions/rasa_tutorial/nlu/babi_nlu.md`

The bAbI dataset

- Example dialogue:

U: good morning
A: hello what can i help you with today
U: i'd like to book a table with french food in paris
A: i'm on it. how many people would be in your party?
U: six people please
A: which price range are looking for?
U: a cheap price range please
A: ok let me look into some options for you
A: what do you think of this option: Chez Gladine?
U: it's perfect
A: great let me do the reservation
U: thanks

The bAbI dataset

- Example dialogue:

U: good morning
A: hello what can i help you with today
U: i'd like to book a table with french food in paris
A: i'm on it. how many people would be in your party?
U: six people please
A: which price range are looking for?
U: a cheap price range please
A: ok let me look into some options for you
A: what do you think of this option: Chez Gladine?
U: it's perfect
A: great let me do the reservation
U: thanks

The bAbI dataset

- Example dialogue:

U: good morning	`greet`
A: hello what can i help you with today	
U: i'd like to book a table with french food in paris	`make_reservation`
A: i'm on it. how many people would be in your party?	
U: six people please	`inform`
A: which price range are looking for?	
U: a cheap price range please	`inform`
A: ok let me look into some options for you	
A: what do you think of this option: Chez Gladine?	
U: it's perfect	`affirm`
A: great let me do the reservation	
U: thanks	`thanking`

The bAbI dataset

- Example dialogue:

U: good morning	→	`greet`
A: hello what can i help you with today		
U: i'd like to book a table with french food in paris	→	`make_reservation`
A: i'm on it. how many people would be in your party?		
U: six people please	→	`inform`
A: which price range are looking for?		
U: a cheap price range please	→	`inform`
A: ok let me look into some options for you		
A: what do you think of this option: Chez Gladine?		
U: it's perfect	→	`affirm`
A: great let me do the reservation		
U: thanks	→	`thanking`

bAbI intents

- Recap on the dataset: the bAbI
  - Intent defined for the dataset
    - `greet`: *hello, hi, good morning, …*
    - `affirm`: *yes, of course, right, …*
    - `deny`: *no, I don't like it, …*
    - `make_reservation`: *can I book a table for six people…*
    - `inform`: *a cheap one, my number is 555, I like indian cuisine, …*
    - `repair_inform`: *actually I prefer spanish cuisine, …*
    - `get_info`: *can I have the address of the restaurant, …*
    - `thanking`: *thanks, many thanks, …*

bAbI entities

- Recap on the dataset: the bAbI
  - Entities defined for the dataset
    - `location`: *in **paris**, in **new york**, …*
    - `cuisine`: *an **indian** restaurant, with **spanish** cuisine, …*
    - `number`: *for **six** people, a table for **two**, …*
    - `price_range`: *a **cheap** restaurant, an **expensive** one, …*
    - `info`: *can I have the **address**, what's restaurant **number** …*
    - `phone_number`: *my phone number is **555-1234**, …*

RASA NLU Input format

- Markdown format (`.md`) or JSON (`.json`)
  - `.md` more human readable (main format)
  - `.json` (legacy, needed for week 5)
- `CA2020_instructions/rasa_tutorial/nlu/nlu.md`
  - to be placed in a `nlu.md` (or `.json`) under `data/`
- Four sections:
  - Common examples
  - Synonyms
  - Regex features
  - Lookup tables

Markdown: examples

- Common examples syntax

```
## intent:intent1
- word1 [word3](entity1) word4 word5 [word6 word7](entity2)
…

## intent:intent2
…
```

- Example

```
## intent:greet
- hi

## intent:make_reservation
- i want [spanish](cuisine) cuisine in [New York](location)
```

Markdown: examples

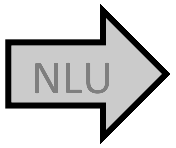
- ASSIGNMENT 1: train rasa nlu
  - `rasa train nlu`

- ASSIGNMENT 2: launch rasa nlu shell
  - `rasa shell nlu`
  - parse “*can you book a restaurant in new york*”

RASA NLU output format

- Json output format

can you book a restaurant in new york



```
{
    “text”: “can you book a restaurant in new york”,
    “intent”: {
        “name”: “make_reservation”,
        “confidence”: 0.8012622594833374
    },
    “entities”: [
      {
        “start”: 29,
        “end”: 37,
        “entity”: “location”,
        “value”: “new york”,
        “confidence”: 0.7535573507062703,
        “extractor": "CRFEntityExtractor“
      }
    ],
    “intent_ranking”: [...]
}
```

Markdown: synonyms (1/2)

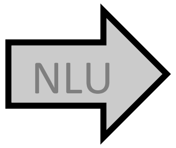
- Synonyms syntax

```
## intent:intent1
- word1 [word3 word4](entity1:synonym_of) word5 word6
…
```

- Example

```
## intent:make_reservation
- i'd like to book a restaurant in [NYC](location:new york)
```

can you book a restaurant in ***NYC***



```
{
    "text": "can you book a restaurant in NYC",
    "intent": "make_reservation",
    "entities":[
        {
          "entity": "location",
          "value": "new york",
          ...,
        }
}
```

Markdown: synonyms (2/2)

- Synonyms syntax (2nd way)

```
## synonym:referred_entity_filler
- word1 word2
- word1 word2 word3
…
```

- Example

```
## synonym:new york
- the big apple
- new york city
- NYC
…
```

DISCLAIMER: defining synonyms this way does not automatically add examples to your dataset. You still need to add examples with the synonyms to have them correctly identifies.
Ex: `i'd like to book a restaurant in [NYC](location)`

Markdown: synonyms

- ASSIGNMENT 3: try using synonyms
  1. parse "*book a restaurant in NYC*"
  2. add synonyms to the `nlu.md` file

```
## intent:make_reservation
- i'd like to book a restaurant in [NYC](location:new york)
- can you book a restaurant in [NYC](location:new york)
- i'd like to book a table in [new york city](location)

## synonym:new york
- new york city
```

  3. re-train rasa nlu: `rasa train nlu`
  4. parse again "*book a restaurant in NYC*"
  5. parse "*book a restaurant in new york city*"

Markdown: regex features

- Regex syntax

```
## regex:entity_type
- regex1
- regex2
…
```

- Example

```
## regex:phone_number
- [0-9]+-[0-9]+
```

DISCLAIMER: as for the synonyms, this does not automatically add examples to your dataset. You still need to add examples with the synonyms to have them correctly identifies.
Ex: `my phone number is [555-04932](phone_number)`

Markdown: regex features

- ASSIGNMENT 4: try using regex
  1. parse “*my phone number is 33-0392934*”
  2. add regex to `nlu.md` file

```
## regex:phone_number
- [0-9]+-[0-9]+
```

  3. re-train rasa nlu: `rasa train nlu`
  4. parse again “*my phone number is 33-0392934*”

Markdown: lookup tables

- Lookup table syntax

```
## lookup:entity_type
path/to/txt/file
```

- Example

```
## lookup:cuisine
data/lookup_tables/cuisines.txt
```

- Lookup table file
  - `.txt` file with list of entity values, one per line

DISCLAIMER: this does not automatically add examples to your dataset. It only defines a regex for each line, which matches exactly the related string.

Markdown: lookup tables

- ASSIGNMENT 5: try using lookup tables
  1. parse "*can i have the directions*"
  2. add lookup table file from `CA2020_instructions/rasa_tutorial/nlu/infos.txt` to your folder
  3. add lookup to your `nlu.md` file

```
## lookup:info
data/infos.txt
```

  4. re-train rasa nlu: `rasa train nlu`
  5. parse again "*can i have the directions*"

RASA Input format

- JSON
  - https://rasa.com/docs/rasa/nlu/training-data-format/#json-format

```
{
    "rasa_nlu_data": {
        "common_examples": [],  ←——   defined as the output
        "regex_features" : [],
        "lookup_tables"  : [],
        "entity_synonyms": []
    }
}
```

Training RASA - Pipelines

- https://rasa.com/docs/rasa/nlu/choosing-a-pipeline/

- Configuration of a nlu pipeline
  - `config.yml`

- Three main pre-defined pipelines
  - **supervised_embeddings**
  - pretrained_embeddings_spacy
  - pretrained_embeddings_convert

Training RASA - Pipeines

- supervised_embedding pipeline
  - https://rasa.com/docs/rasa/nlu/components/

```
language: "en"

pipeline:
- name: "WhitespaceTokenizer"
- name: "RegexFeaturizer"
- name: "CRFEntityExtractor"
- name: "EntitySynonymMapper"
- name: "CountVectorsFeaturizer"
- name: "CountVectorsFeaturizer"
  analyzer: "char_wb"
  min_ngram: 1
  max_ngram: 4
- name: "EmbeddingIntentClassifier"
```

Training RASA

- Training rasa via command line
  - `rasa train nlu`

- Training rasa via Python API
  - script in
    `CA2020_instructions/rasa_tutorial/nlu/train_nlu.py`

Testing RASA

- Testing via command line
  - `rasa shell nlu`

- Using RASA http API
  1. `rasa run --enable-api -m models/`**`[model_name]`**
  2. `curl localhost:5005/model/parse -d '{"text":"can i book a table in madrid"}'`

- Testing via Python API
  - script in
    `CA2020_instructions/rasa_tutorial/nlu/test_nlu.py`

Useful links

- Some useful links
  - https://rasa.com/docs/
  - https://rasa.com/docs/rasa/user-guide/rasa-tutorial/
  - https://rasa.com/docs/rasa/nlu/about/