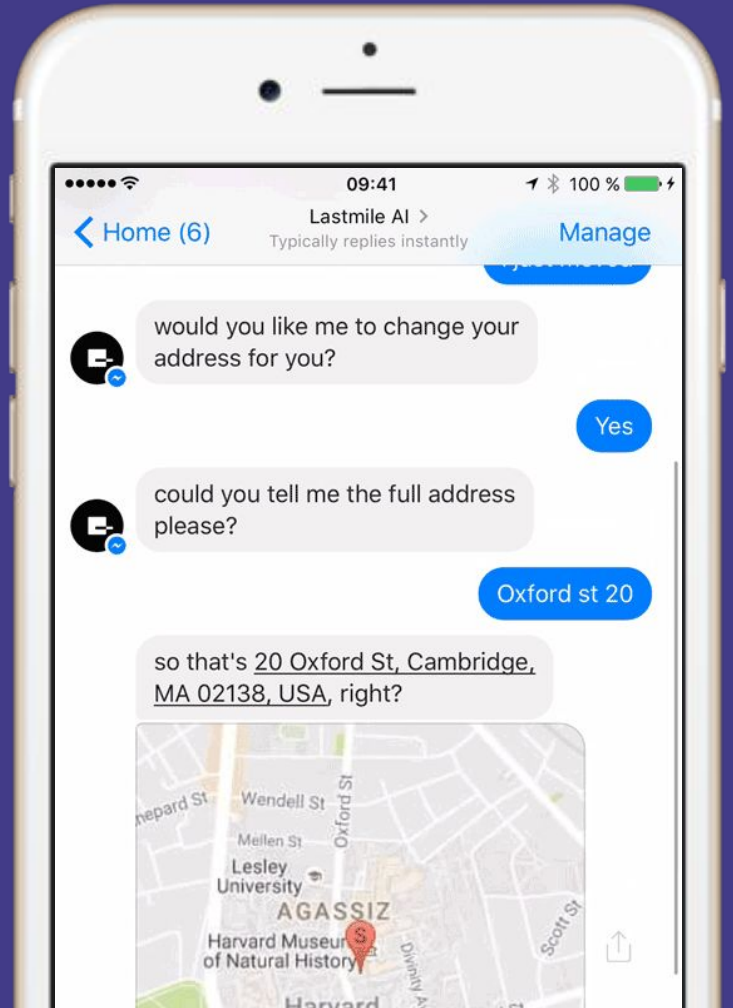
Conversational AI: Building clever chatbots

Tom Bocklisch, Lead ML Engineer @ LASTMILE

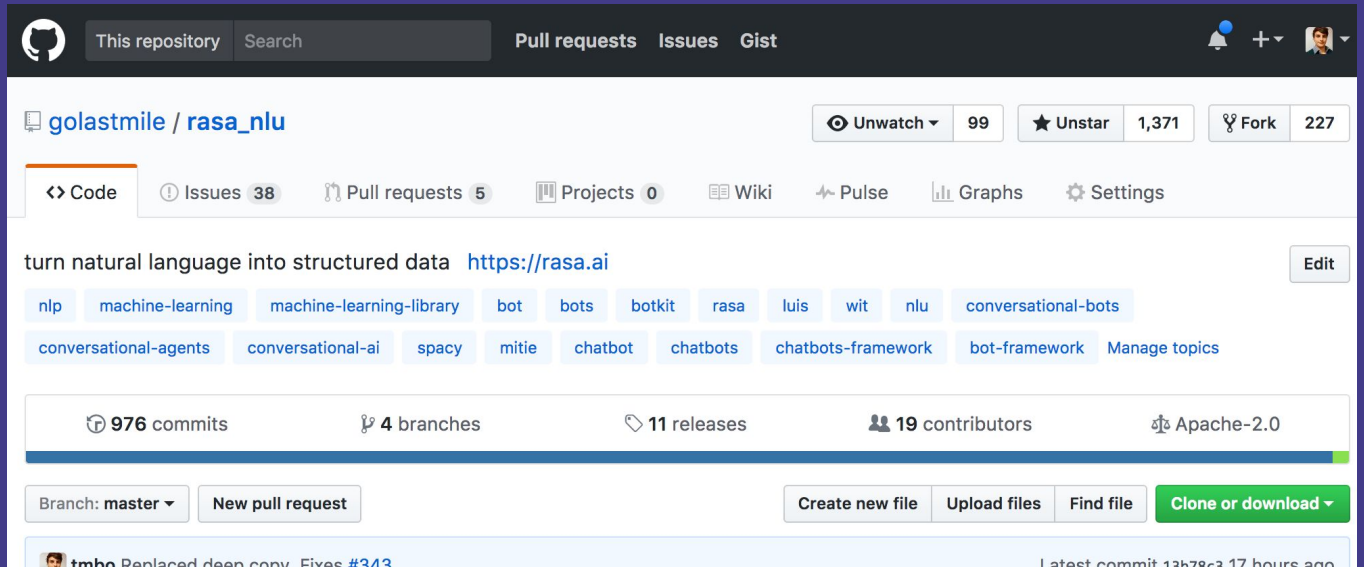
Conversational AI will dramatically change how your users interact with you.



Example of a live Skill:
A customer can change her address via Facebook Messenger

LASTMILE

An open source, highly scalable ML framework to build conversational software



LASTMILE

Introduction

We work on the core technology for next-generation conversational AI

LASTMILE

is a **technology company** developing **conversational AI**.

Goal: next-generation intelligent bots

Team: tight-knit, fast-moving team of researchers, engineers, designers and product people

Location: everywhere (honestly: Berlin, Edinburgh, Beijing)

Founders:
Dr. Alan Nichol (CTO)
Alexander Weidauer (CEO)

UNIVERSITY OF CAMBRIDGE

Advisory Board:
Chad Fowler (MD & CTO @ Wunderlist)
Matthaus Krzykowski (former Co-Founder @ Xyo)
Cat Noone (Designer & Founder @ Iris)

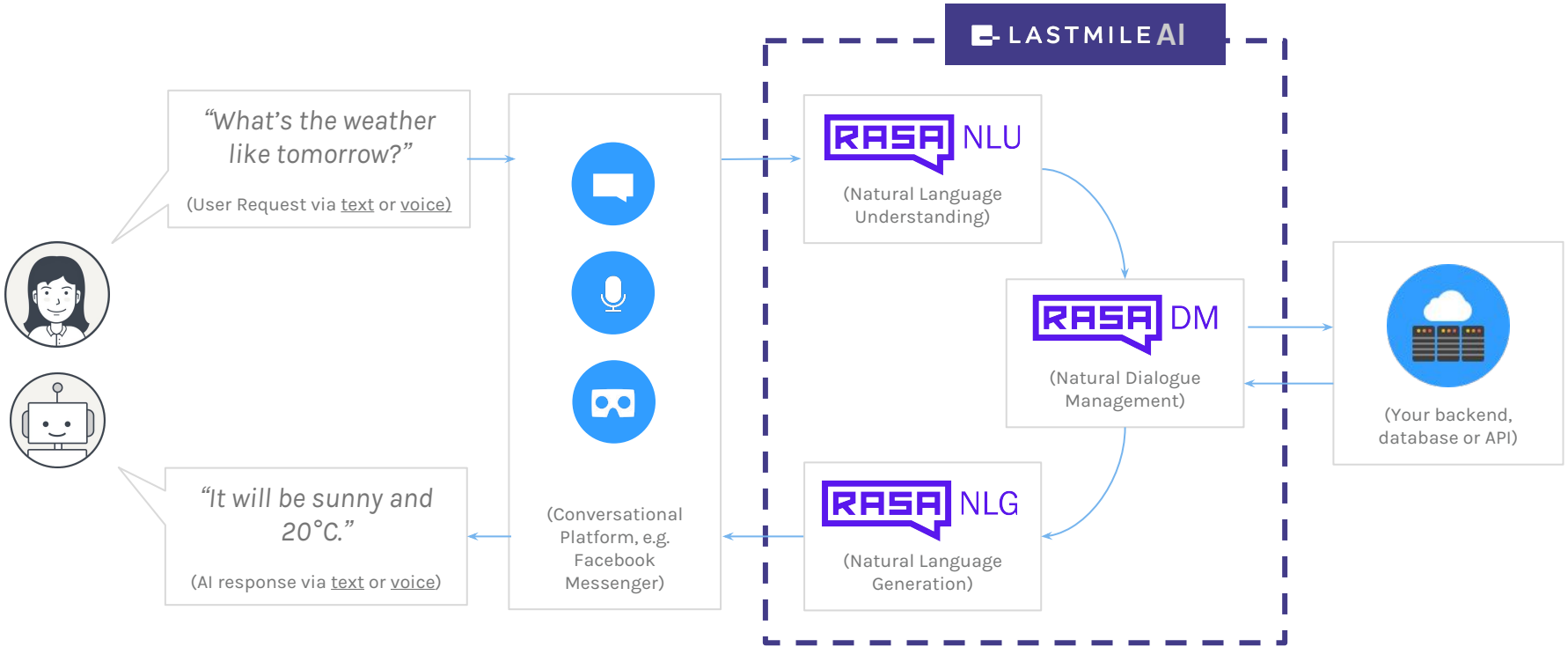
Investors:

techstars

Reference customers:

RAIFFEISEN
ERGO
SwissLife

Architectural Overview



Under The Hood

Under The Hood

Natural Language Understanding

What's the weather like tomorrow?

Natural Language Understanding

Example Intent Classification Pipeline

"What's the weather like tomorrow?" { "intent": "**request_weather**" }

Vectorization → Intent Classification

Example Entity Extraction Pipeline

"What's the weather like **tomorrow**?" { "date": "**tomorrow**" }

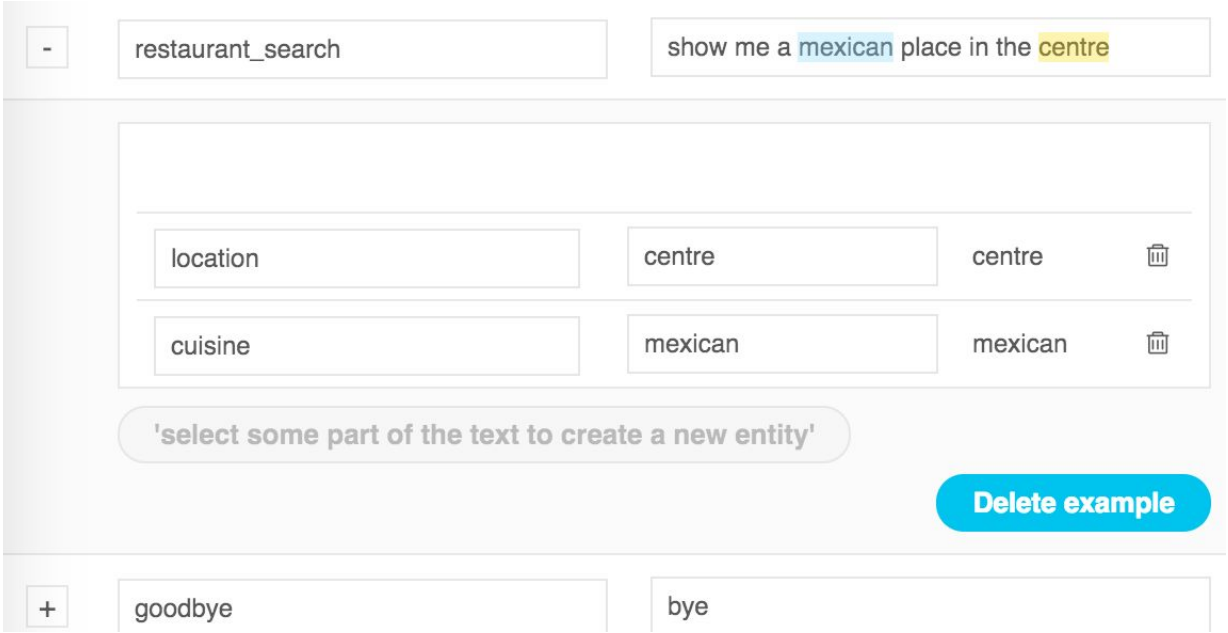
Tokenizer → Part of Speech Tagger → Chunker → Named Entity Recognition → Entity Extraction

Demo

Demo

1. Create your training data

E.g. using the contributed rasa NLU gui at
https://golastmile.github.io/rasa-nlu-trainer/



Demo

2. Configure the model

Configure the model

In [2]:

```
model_config = {
  "pipeline": ["nlp_spacy",
               "ner_crf",
               #"ner_spacy",
               "intent_featurizer_spacy",
               "intent_classifier_sklearn"],
  "language": "en"
}
```

3. Train

In [3]:

```
# Train NLU model
config = RasaNLUConfig(cmdline_args=model_config)

trainer = Trainer(config)
training_data = load_data("example-data/demo-rasa.json")

# run the training
interpreter = trainer.train(training_data)
logging.info("done")
```

Training the model

```
INFO:root:Trying to load spacy model with name 'en'
INFO:root:Added 'nlp_spacy' to component cache. Key 'nlp_spacy-en'.
INFO:root:Training data format at example-data/demo-rasa.json is rasa_nlu
INFO:root:Training data stats:
        - intent examples: 38 (4 distinct intents)
        - found intents: affirm, goodbye, greet, restaurant_search
        - entity examples: 7 (2 distinct entities)
        - found entities: cuisine, location

INFO:root:Starting to train component nlp_spacy
INFO:root:Finished training component.
INFO:root:Starting to train component ner_crf
INFO:root:Finished training component.
INFO:root:Starting to train component intent_featurizer_spacy
INFO:root:Finished training component.
INFO:root:Starting to train component intent_classifier_sklearn
[Parallel(n_jobs=1)]: Done  12 out of  12 | elapsed:    0.1s finished
INFO:root:Finished training component.
INFO:root:done
```

```
Fitting 2 folds for each of 6 candidates, totalling 12 fits
```

Demo

4. Use Model

Playing around with the trained model

In [27]:

```
# i am looking for an italian restaurant in Vienna
result = interpreter.parse(
  "i am looking for an italian restaurant in Vienna")
pprint(result)
```


```
{
  "entities": [
    {
      "start": 20,
      "extractor": "ner_crf",
      "end": 27,
      "value": "italian",
      "entity": "cuisine"
    },
    {
      "start": 42,
      "extractor": "ner_spacy",
      "end": 48,
      "value": "Vienna",
      "entity": "GPE"
    }
  ],
  "intent": {
    "confidence": 0.80703667042349947,
    "name": "restaurant_search"
  },
  "text": "i am looking for an italian restaurant in Vienna",
```

Under The Hood

Under The Hood

Dialogue Handling

previous
Action

"What's the weather like tomorrow?"

Intent

Entities

State

SVM
Recurrent NN
...

next
Action

"It will be sunny and 20°C."

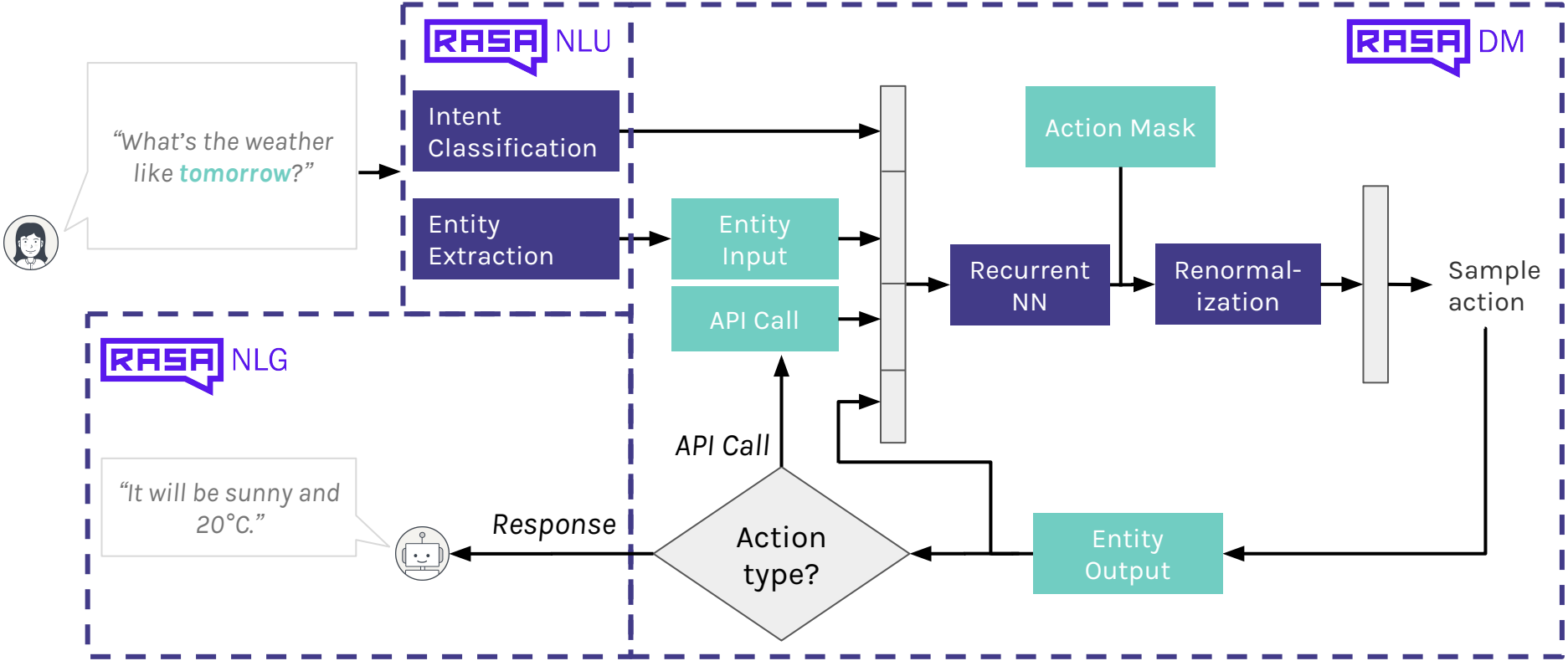
"Thanks."

updated
State

after next
Action

Detailed Dialogue Handling

Similar to LSTM-dialogue prediction paper: https://arxiv.org/abs/1606.01269

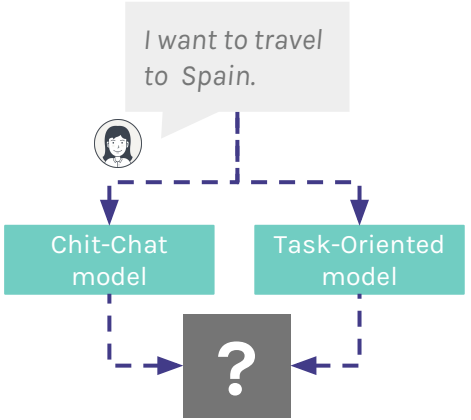


Final Thoughts

Open Challenges

Challenges for curious minds:

- Combination of different dialogue models



- Unsupervised multi-language entity recognition

- Dialogue generalisation (e.g. optional questions)

Current Research

Good reads for a rainy day:

- Last Words: Computational Linguistics and Deep Learning (blog)
  https://goo.gl/IGSRuj
- Memory Networks (paper)
  https://arxiv.org/pdf/1410.3916
- End-to-End dialogue system using RNN (paper)
  https://arxiv.org/pdf/1604.04562.pdf
- MemN2N in python (github)
  https://github.com/vinhkhuc/MemN2N-babi-python

Summary

3 take home thoughts:

- Conversational AI is a big part of the future
- Deep ML techniques help advance state of the art NLU and conversational AI
- Open source is strategically important for enterprises implementing AI

Get in touch!

LASTMILE

Tom Bocklisch
Lead ML Engineer
tom@golastmile.com