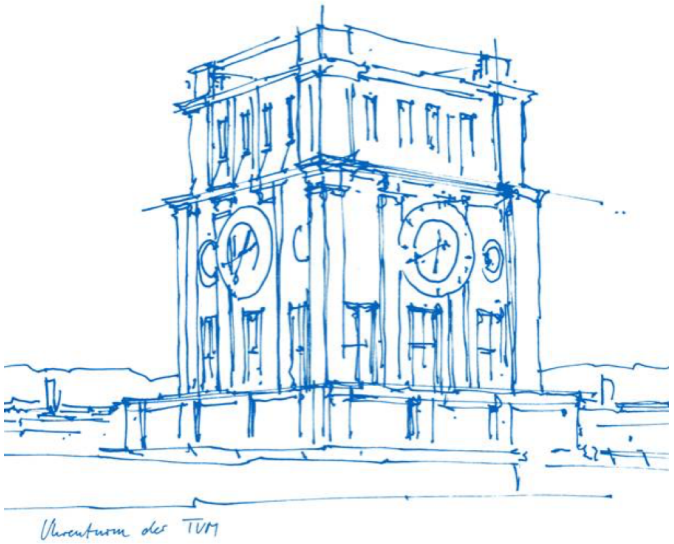
HORVÁTH & PARTNERS
MANAGEMENT CONSULTANTS

# Development of an Artificial Conversation Entity for continuous learning and adaptation to user's preferences and behavior

Final Presentation

18th February 2019


| | |
|---|---|
| Authors: | Harshita Agarwala, Robin Becker, Mehnoor Fatima, Lucian Riediger |
| Mentors: | Dr. Andrei Belitski, M.Sc. Olena Schüssler |
| Co-Mentor: | M.Sc. Laure Vuaille |
| Project Lead: | Dr. Ricardo Acevedo |
| Supervisor: | Prof. Dr. Massimo Fornaiser |

# Agenda

01 Motivation

02 Research

03 Data Generation

04 Natural Language Understanding

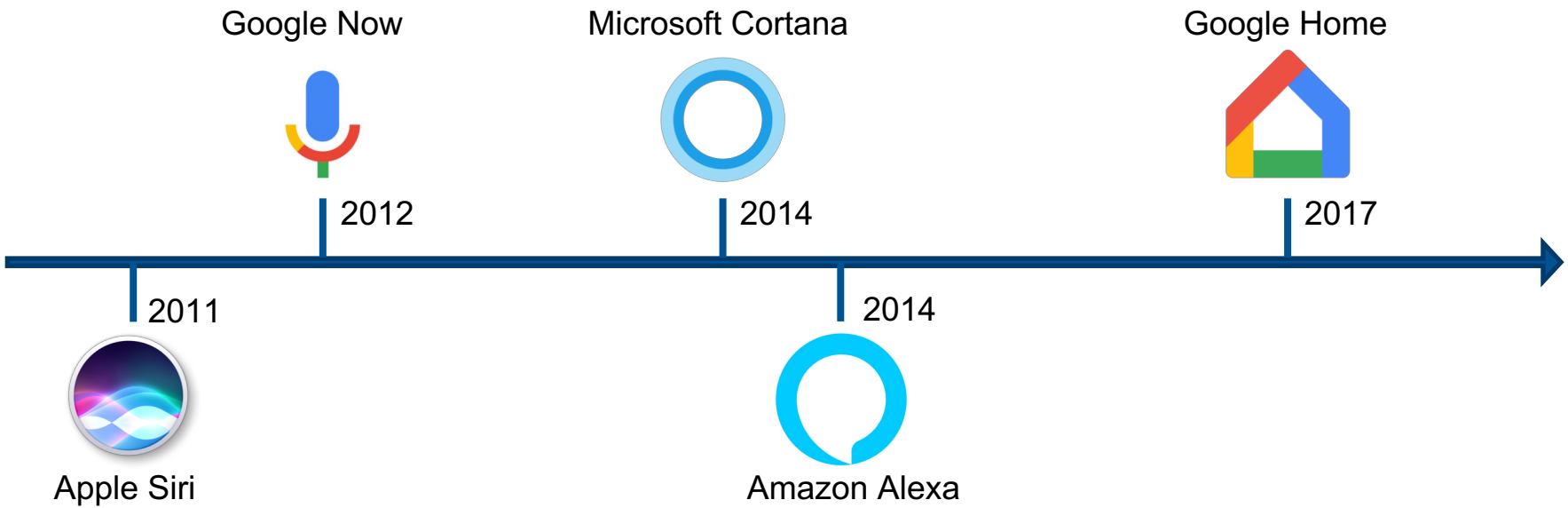
05 Recommender System

06 Dialogue Management

07 Outlook

# Motivation

# Technical progress in machine learning and NLP led to progress in ACEs like voice assistants and chat bots



# Broad spectrum of feasible input hinders ACEs from going into deeper conversations

Call person X.
Send a message to person X.
What are good Indian dishes?
When is mother's day?
How do you say X in French?
Open app X.
Navigate me to the nearest gas station.
What's the status on flight X?
What are today's news?
Do you love me?
Turn on/off wifi.
Set an alarm.
Calculate X-Y*Z.
Convert X into Y.
What's the weather in X?
What's the time in X?
Flip a coin.
Take a note.
Read my newest e-mail.
Play the new album of X.
Recommend a good restaurant.

What are today's news?

Open them.

What about yesterday?

# Our ACE is should be able to go into depth by analyzing short and long term history of the conversation

Problem Statement:

- Hotel environment
- ACE is supposed to process classical hotel services such as delivering of drinks, food, towels etc
- Enable in depth conversation in order to increase customer satisfaction in terms of convenient use of ACE as well as recommended products
- System should not be run on a cloud

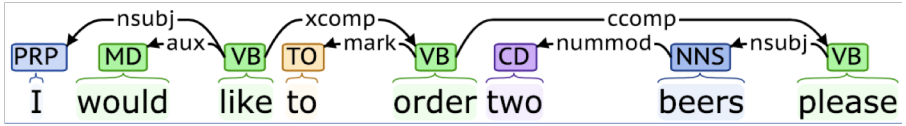
Assumptions:

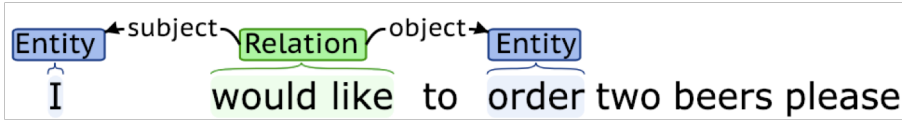
- Voice to text and text to voice components are given

# Research

# Parsers gramamtically analyze input and work with databases of words

“I would like to order two beers please”

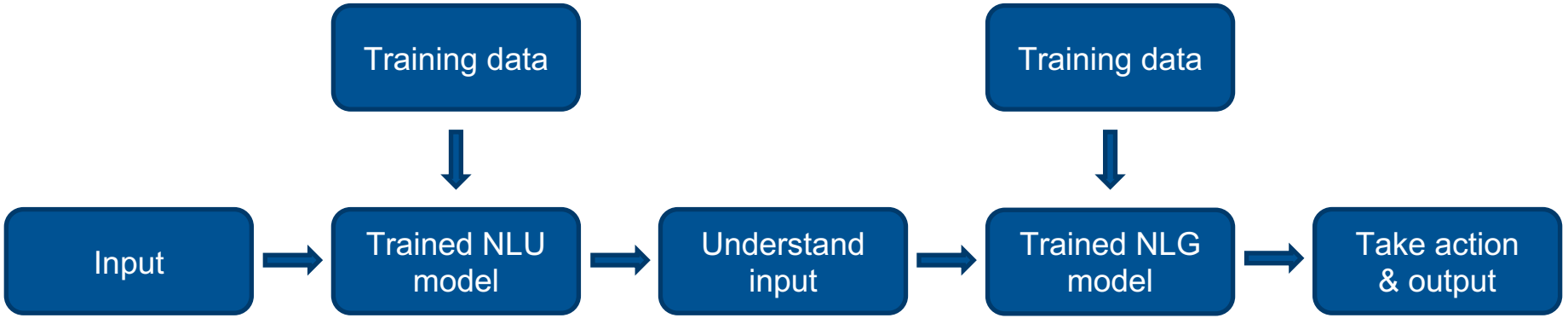




State of the art parsers


spaCy

PRINCETON UNIVERSITY
WordNet
A lexical database for English

# Machine-learning models usually cover the enitre pipeline and are often cloud based




RASA

Amazon Lex

LUIS

wit.ai

IBM Watson

Dialogflow

# The Rasa machine-learning framework is more suitable for our project

## Parser + WordNet

+ Use of existing parsers like Stanford NLP or spaCy
+ WordNet largest database of its kind
+ No need for training data

- Extension of WordNet (by brand names) might be complicated
- Only interpretation of input, output generation not included

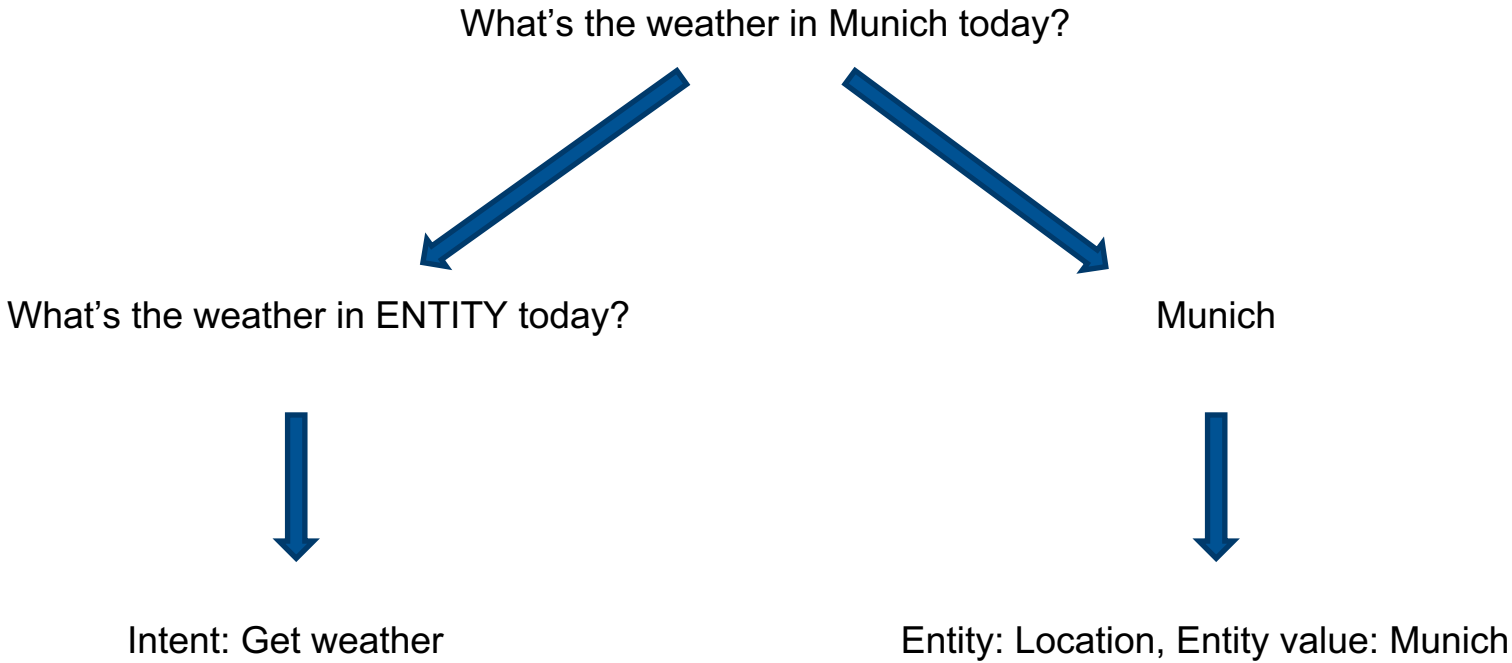
## Machine-learning model (Rasa)

+ Covers entire pipeline (including output generation)
+ Entire pipeline is customizable
+ Model can be trained to our specific environment
+ Gives confidence score for intents and entities

- Training data necessary

# Data generation

# Intents contain the basic message of a sentence and entities specify the values



# We gathered three types for a seamless conversation and robust intent classification

## Intents

Main intents:
- Order drink
- Change order
- Order food
- Get recommendations

Conversational intents:
- Cancel order
- Good
- Bad
- Hello
- Goodbye
- Thanks
- Confirm
- Negate
- How are you

General intents:
- Get weather
- Book restaurant
- Rate book
- Play music
- Search creative work
- Search screening event

## Entities

- Drink
- Food
- Location
- Amount
- Size
- Temperature
- Topping
- Sugar amount

# Placeholders in bare-bone sentences are replaced by entity values in a nested procedure

1. I would like to order a ENTITY please

2. Pick an arbitrary number of elements of this list:

3. Substitute placeholders with variables:
→ I would like to order AMOUNT SIZE ENTITY to LOCATION please

4. Substitute variables with values:
→ I would like to order 3 large water to my room please

Entity mapping list:

a → AMOUNT
ENTITY → SIZE ENTITY
ENTITY → TEMPERATURE ENTITY
ENTITY → DRINK and AMOUNT DRINK2
ENTITY → ENTITY to LOCATION
ENTITY → ENTITY with TOPPING
.
.
.
↓

Arbitrary sample:

a → AMOUNT
ENTITY → SIZE ENTITY
ENTITY → ENTITY to LOCATION

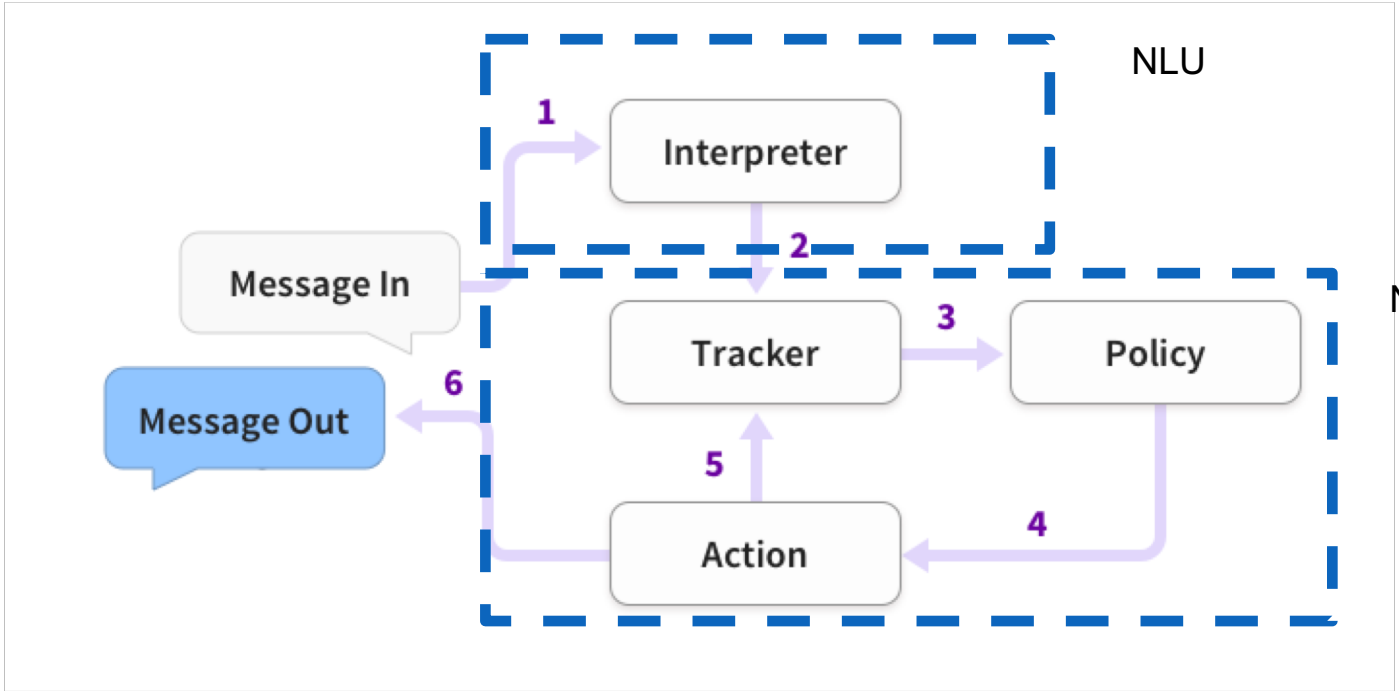
# We extended the data generation script to create more realistic and robust data

1. Adding noise to generated sentences in order to train a more robust model
- First approach to add a random string of characters between two words
- Second approach to add a real word as noise

2. Combine entities based on joint probabilities
- E.g. Coffee with milk and sugar instead of lemon

# Natural Language Understanding

# High level architecture of Rasa‘s framework



# The spaCy pipeline fits our situations the best

## spaCy

- Uses pre-trained vectorizer to analyze input
- Rule of thumb: use if less than 1000 labelled examples available

Advantage:

- Assume our training data contains wine as a drink, but not champagne
- Input of hotel guest: “I would like to order champagne”
- Since champagne and wine have similar vectors, the model is more likely to identify champagne as a drink, even though it never appeared in the training data

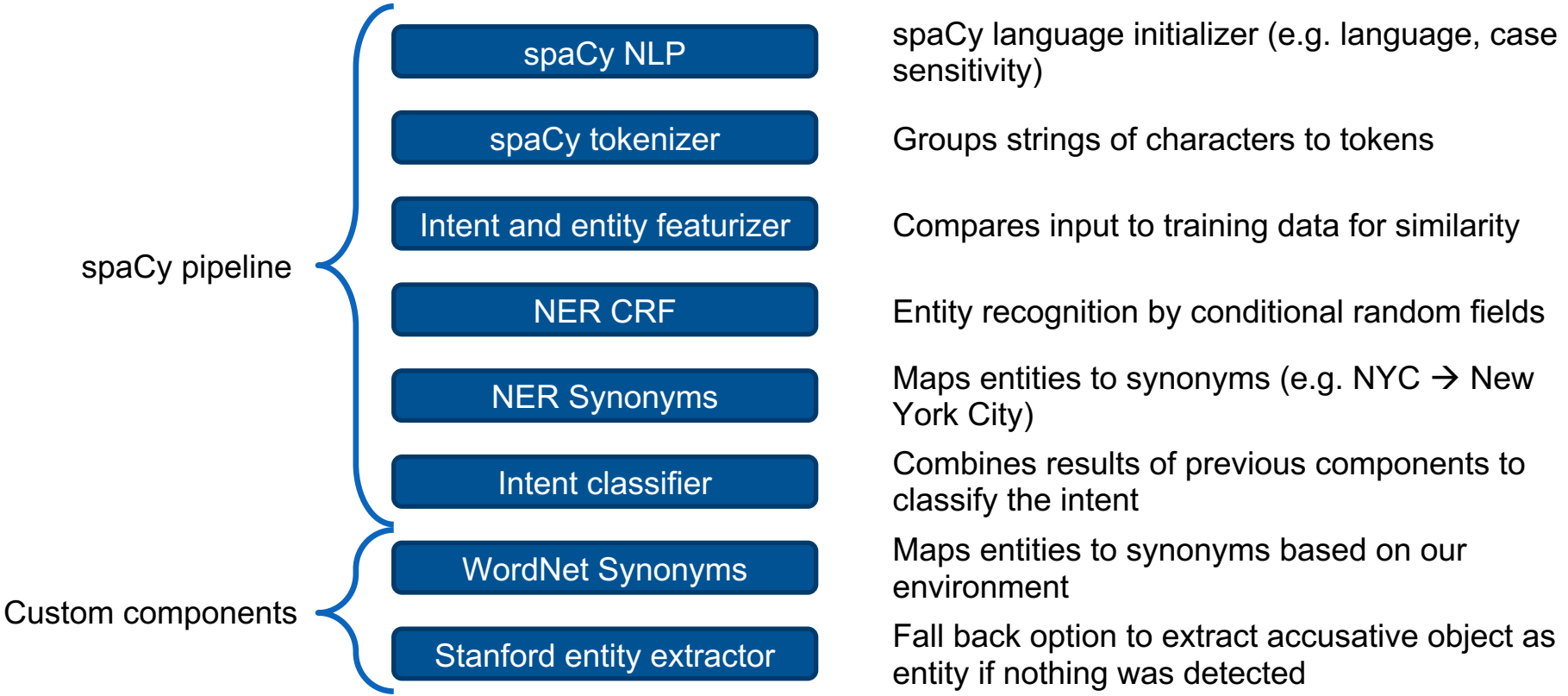
## TensorFlow

- Word vectorization is trained by training data
- Rule of thumb: use if more than 1000 labelled examples available

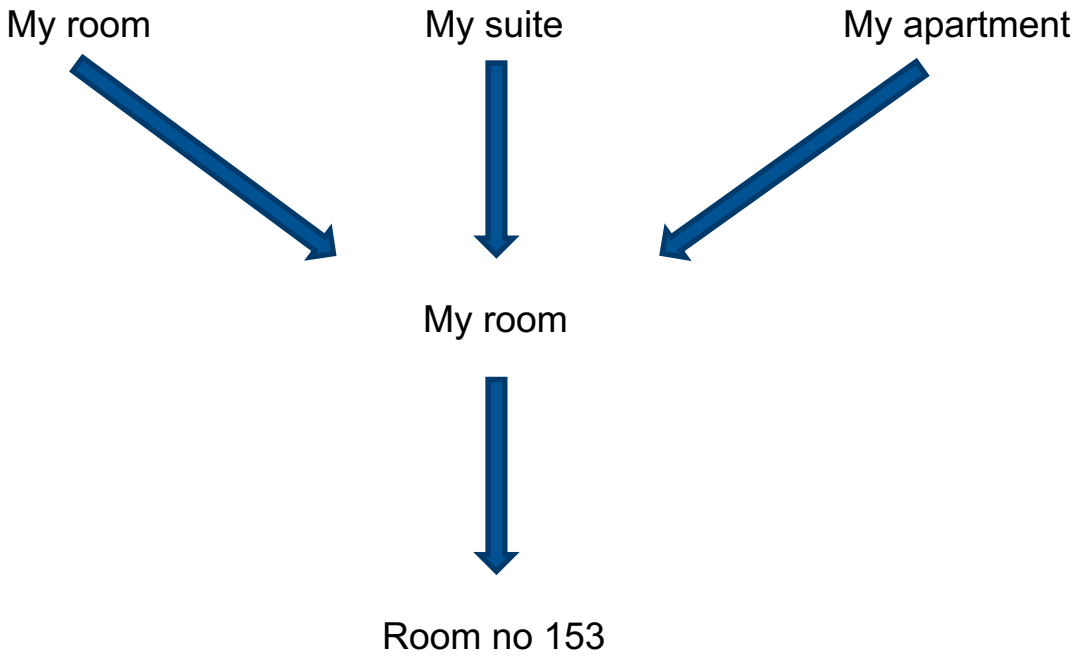
Advantage:

- Assume the task of the ACE is to differentiate accounting reports from others
- The word “balance” is usually closely related to words like “symmetry” if vectorized
- If ”balance” appears often together with “cash” or similar financial words in the training data, word vectorization will reflect this relation

# We extended the NLU pipeline to improve entity detection and machine-readability



# Synonym detection makes our system more machine-readable



# Recommender System

# The recommender system needs to be customizable to hotels‘ needs

Reminder / Environment:

- ACE should go more into depth than competitors
- Recommender system should be easily customizable for hotels with different needs
- Recreational hotels usually have guests for up several days, maybe even weeks but low frequency
- Business hotels usually have guests for one or a few days only, but with higher frequency of the same guest

→ Use as much information as possible

# Recommendations are based on three different contexts

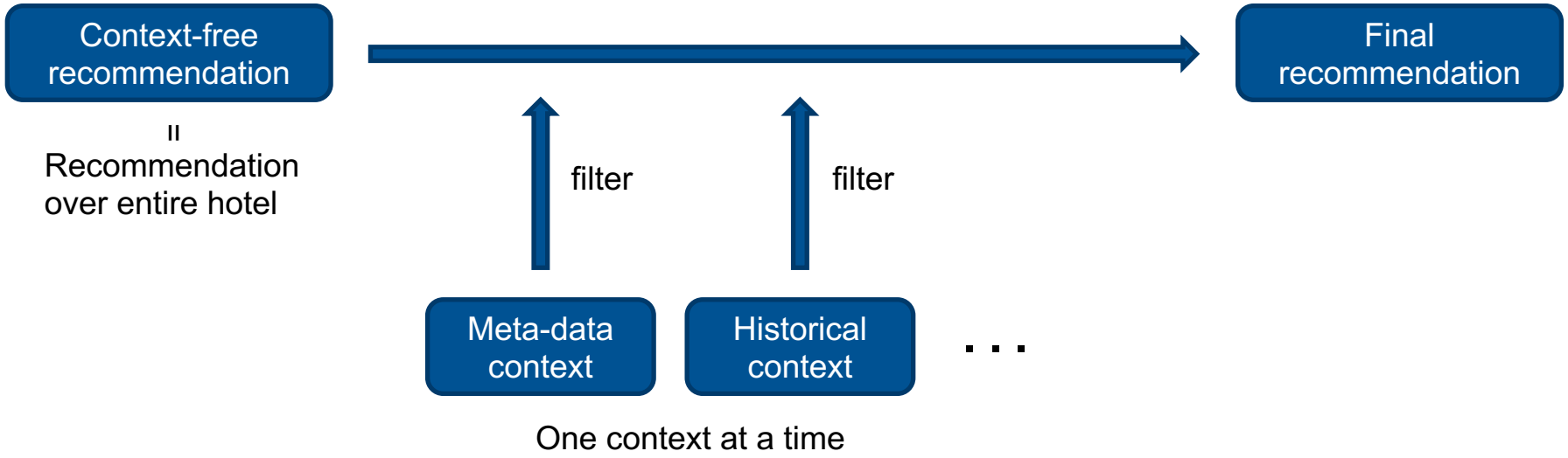
Meta-data context:
Age: Underaged, adult, senior
Time of day: Morning, daytime, evening
Nutrition habits: Alcoholic, non alcoholic

Entity context:
Recommendation for combination of entities (drink-drink, drink-temperature)

History context:
Incorporation of past interactions of customers

# Recommendations are filtered, context by context

How contextual post-filtering works:



# Recommendations are based on three different contexts

Meta-data context:
- Finite, discrete information
- Dense data

→ Simple relative frequency

Entity context:
- Correlations between entities
- Item-item collaborative filtering

→ Cosine similarity

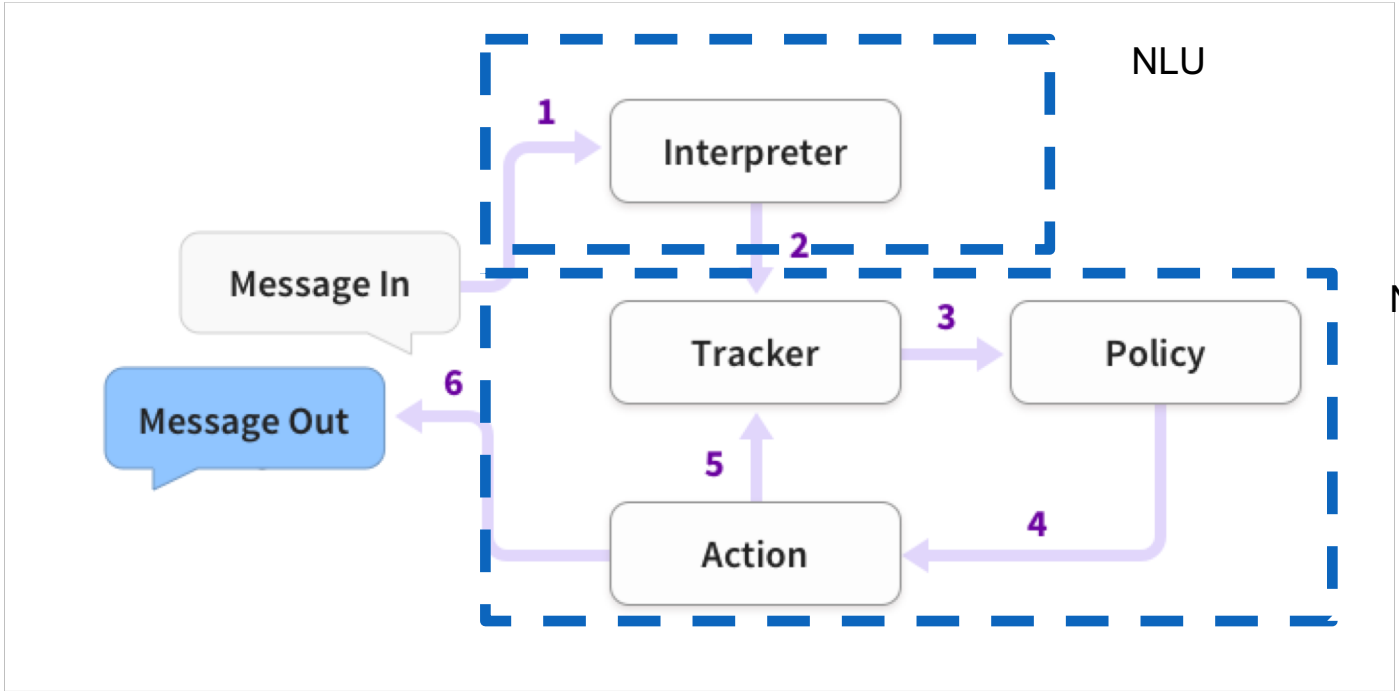
History context:
- Considers all past interactions
- Collaborative filtering with implicit feedback data

→ Weighted matrix factorization

Cold start problem:
- Manually curate meta-data contexts
- Estimate item-item correlations with websearch
- Sample customers from clusters of entities

# Dialogue Management

# High level architecture of Rasa‘s framework



# We use Rasa Core to have a real conversation and consolidate components

Motivation:

- To consolidate outputs of all the components (Rasa NLU, customer history, recommendations)
- To curate a conversation between the user and the bot

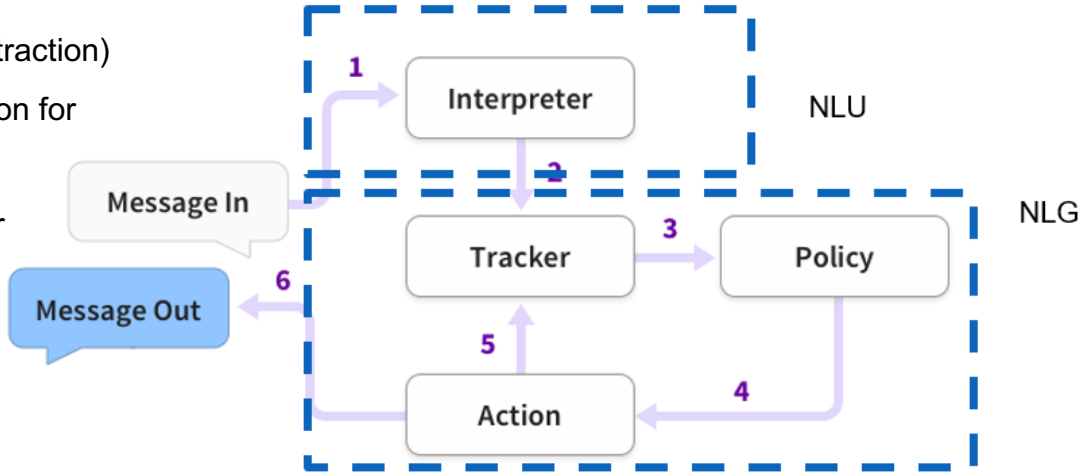
Other Requirements:

- Open source framework
- Keeps the context of the conversation
- Developed with zero or less training data

# Rasa Core pipeline completes the NLP framework

Rasa Core is an open source machine learning framework used for dialogue management in the development of conversational softwares

1. The input message is received and passed to an Interpreter (Intent & Entities Extraction)
2. Trackers keep track of the state of conversation for a single user
3. Policy receives the current state of the tracker
4. Policy decides about the next action
5. Chosen Action is logged in tracker
6. Response sent to the user



# Rasa Core allows us to add features that differentiate our prototype from normal ACEs

1. Slot Filling

- Used for collecting multiple pieces of information in a row

I. Required Slots are detected
II. Mapped to entities or text
III. Slot Validation
IV. Slot Setting

2. Co-referencing
- Used for short term history and context (More details on the next slide)

3. Recommendations
- Takes in the context of the conservation and recommends accordingly

# Co-referencing ensures a natural conversation

1. ”I would like to get a coke”
→ Intent: order drink, entity: drink, entity value: coke

Main entity: drink - existing

2. “Actually, make it two”
→ Intent: book restaurant, entity: time, entity value: two

Main entity: restaurant name - missing

3. Check if time difference between both inputs is sufficiently low

4. Check for similar entities
→ a = Amount, two = rating

5. Change flow of dialogue
→ “I would like to get two coke”

# Demo

# Outlook

# We developed an ACE that operates on little data and increases customers convenience

Summary:

- Developed a framework for artificial conversational entity for a resort environment

    1. Data Generation and Optimization
    2. Automatic Synonyms Detection for ambiguous entities
    3. Recommender System based on customers personal and general preferences
    4. Dialogue Management
    5. Co-referencing

- Operates on little yet sufficient sample data

Outlook:

- Integrating speech-to-text and text-to-speech components
- Gathering more input data from customers, improving the model with reinforcement learning
- Extending it to other service domains

# Q&A

1. Motivation
- Breadth vs. depth
- Problem statement & assumptions
2. Research
- Parsers
- Machine-learning models
3. Data Generation
- Intents & entities
- Data generation script & extensions
4. Natural Language Understanding
- High-level architecture
- Pipelines
- Custom components
5. Recommender System
- Data in different contexts
- Post filtering
6. Dialogue Management
- High-level architecture
- Rasa Core pipeline
- Rasa Core features
- Co-referencing
7. Outlook

# Backup

# Performance measuring of training data

Intent recognition

| | Models | Precision | Recall | F1-Score |
|---|---|---|---|---|
| 0 | Baseline (B) | 0.98 | 0.98 | 0.98 |
| 1 | B+Noise (BN) | 0.98 | 0.98 | 0.98 |
| 2 | BN + Lookup Table (BNL) | 0.98 | 0.98 | 0.98 |
| 3 | BN + Synonym (BNS) | 0.98 | 0.98 | 0.98 |
| 4 | BN + Lookup + Synonym (BNLS) | 0.98 | 0.98 | 0.98 |
| 5 | B+One Real Word Noise (BO) | 0.99 | 0.99 | 0.99 |
| 6 | B+Two Real Word Noise (BT) | 0.99 | 0.99 | 0.99 |
| 7 | Half Amount Barebone (HBB) | 0.83 | 0.83 | 0.83 |

Entity recognition

| | Models | Precision | Recall | F1-Score |
|---|---|---|---|---|
| 0 | Baseline (B) | 0.96 | 0.96 | 0.96 |
| 1 | B+Noise (BN) | 0.96 | 0.96 | 0.96 |
| 2 | BN + Lookup Table (BNL) | 0.98 | 0.98 | 0.97 |
| 3 | BN + Synonym (BNS) | 0.98 | 0.98 | 0.98 |
| 4 | BN + Lookup + Synonym (BNLS) | 0.98 | 0.98 | 0.97 |
| 5 | B+One Real Word Noise (BO) | 0.99 | 0.99 | 0.99 |
| 6 | B+Two Real Word Noise (BT) | 0.99 | 0.99 | 0.99 |
| 7 | Half Amount Barebone (HBB) | 0.86 | 0.87 | 0.85 |

# Metadata Context

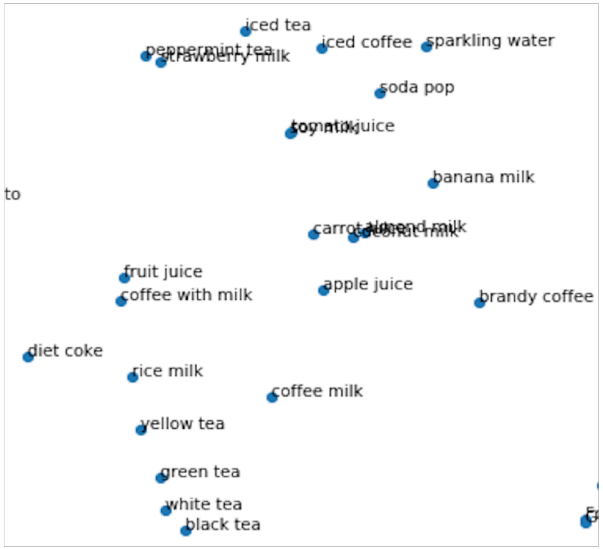
- Finite, discrete information about customer, e.g. male and female for gender context
- Continuous contexts can be discretized: time becomes morning, noon, evening etc.
- Few possible values means dense data

→ Simply consider relative frequency: How often is entity ordered in the morning vs. in the evening

→ Cold start problem removed by manually curating data

# Entity Context

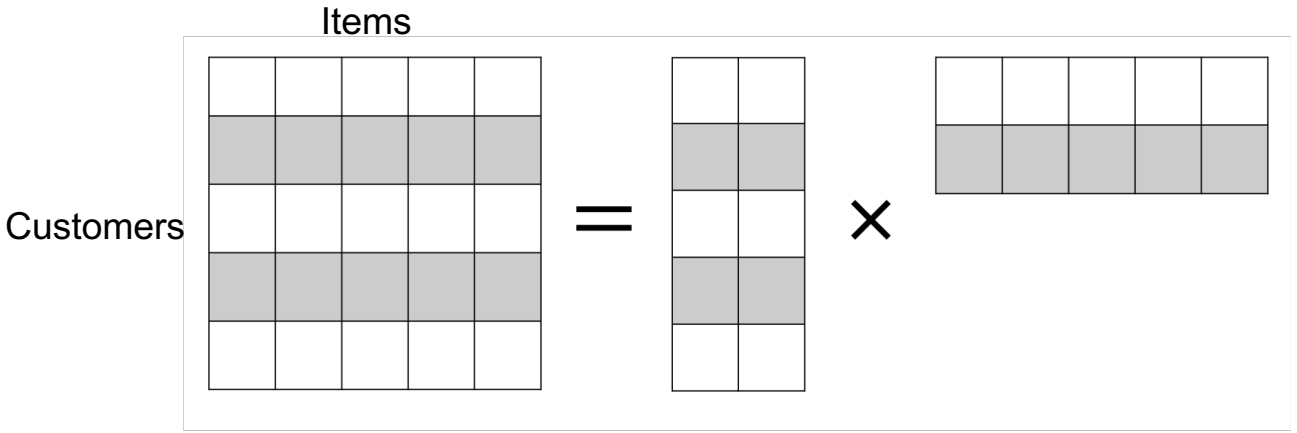
- Item-item collaborative filtering
- Many algorithms exist: Cosine similarity as a baseline
- But: Coldstart problem
- Idea: Use web search engine to estimate correlations




# History Context

- Based on all past interaction with the system
- Collaborative filtering with implicit feedback data

→ Weighted matrix factorization:




Remove cold start problem: sample customers from random clusters of similar entities

# System Integration

Obtain final recommendation through weighted average of baseline and contexts:

- Recommendations can be as general as necessary
- Cold start problem can be avoided for specific contexts
- Recommendations can be adapted to the hotel's needs

Use cases:

- Allow customer to ask for recommendations
- Automatically recommend when high enough confidence is reached
- Present customer with filtered list of available items instead of exhaustive list
- Improve ACE itself: improved default slot filling