Rasa Certification Workshop

May 2020

Mady Mantha, Juste Petraityte, Karen White

Agenda

Day 1: Deep dive into NLU and dialogue management with Rasa Open Source, livecoding and testing

Day 2: Deep dive into DIET (Dual Intent and Entity Transformer) and TED (Transformer Embedding Policy) and create an MVP assistant

Day 3: Adding custom actions and implementing forms

Day 4: Deploying and improving your assistant using Rasa X

Day 5: Recap and certification

*More info blablabla

The Rasa Team

Mady Mantha
Sr. Technical Evangelist

Juste Petraityte
Head of Developer Relations

Karen White
Developer Marketing Manager

How to get help

- Please ask your questions in the **#workshop-help** Slack channel rather than the Zoom chat. Slack is the place the Rasa team will be monitoring most closely.
  - Karen, Mady, and Juste will be in Slack answering questions, as well as Arjaan, Melinda, and Ella from our Customer Success Engineering team
- Monday - Friday, the Rasa team will be dedicating time to answering your questions in Slack from 4 pm - 6 pm CEST (Central European Summer Time)
- Feel free to ask questions outside of these hours, but responses may be a little slower
- A note on time zones:
  - The Rasa team is based across the US and in Berlin. We'll do our best to answer questions within team members' working hours, but please keep in mind, some discussions may need to take place async rather than in real time.

Deep dive into DIET, TED, and some coding!

How to get help

- Please ask your questions in the **#workshop-help** Slack channel rather than the Zoom chat. Slack is the place the Rasa team will be monitoring most closely.
    - Karen, Mady, and Juste will be in Slack answering questions, as well as Arjaan, Melinda, and Ella from our Customer Success Engineering team
- Monday - Friday, the Rasa team will be dedicating time to answering your questions in Slack from 4 pm - 6 pm CEST (Central European Summer Time)
- Feel free to ask questions outside of these hours, but responses may be a little slower
- A note on time zones:
    - The Rasa team is based across the US and in Berlin. We'll do our best to answer questions within team members' working hours, but please keep in mind, some discussions may need to take place async rather than in real time.

Agenda

Day 1: Deep dive into NLU and dialogue management with Rasa Open Source, livecoding and testing

Day 2: Deep dive into DIET (Dual Intent and Entity Transformer) and TED (Transformer Embedding Policy) and create an MVP assistant

Day 3: Adding custom actions and implementing forms

Day 4: Deploying and improving your assistant using Rasa X

Day 5: Recap and certification

*More info blablabla

Day 2 Roadmap

- First half: talk, with some time for questions
- Second half: we'll code together
- Recap: talk, with some time for questions

What happened yesterday

- Rasa Open Source: NLU and dialogue management

- Created a bot!

Core Deep Dive

Domain

Defines the “world” of your assistant - what it knows, can understand, and can do

```
intents:
…
entities:
…
slots:
…
actions:
…
responses:
…
session_config:
…
```

What your bot can understand

What your bot can remember

What your bot can do (and say)

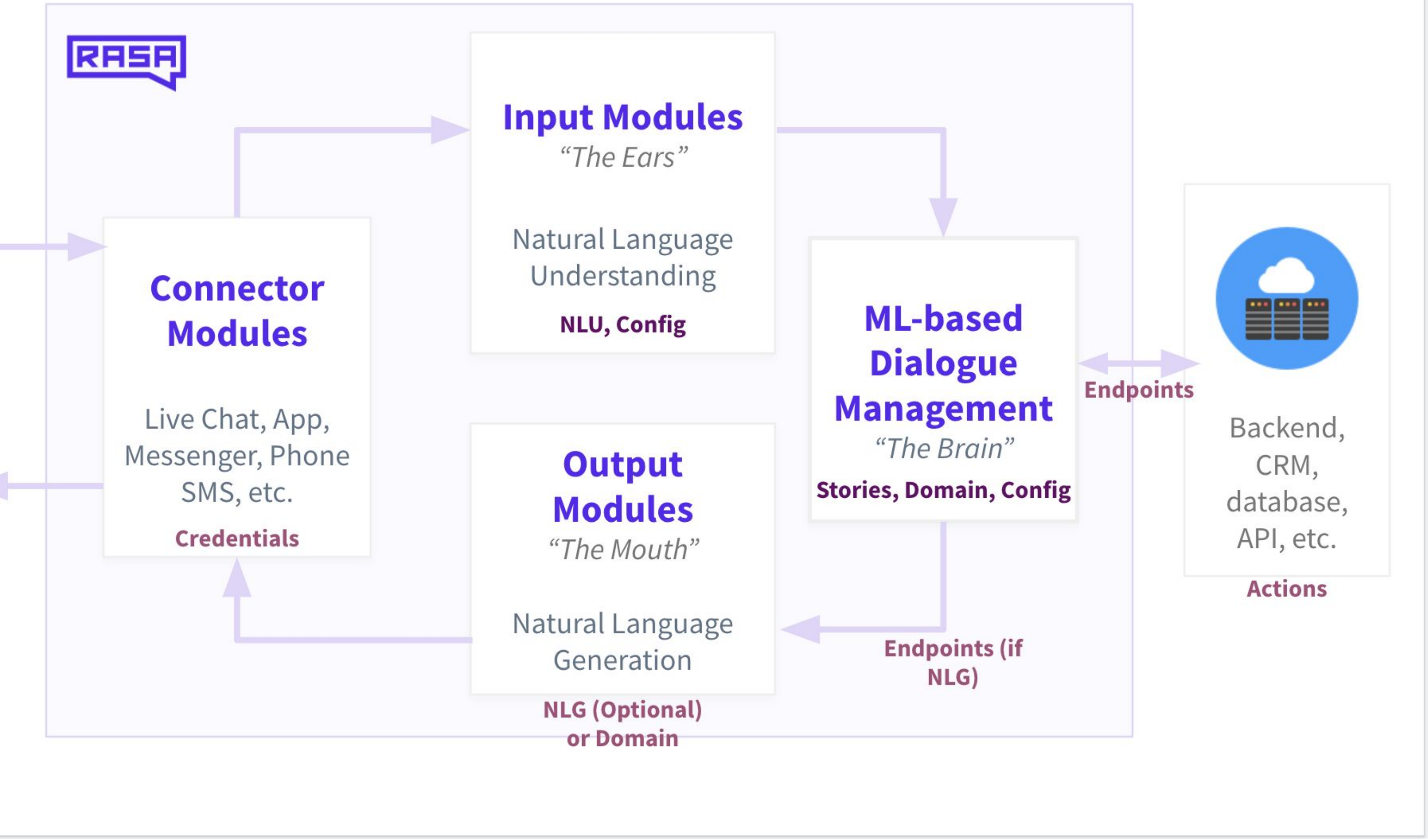
What your bot can say (and display)

What your bot considers a new conversation

Project setup: Files

`__init__.py`	an empty file that helps python find your actions
`actions.py`	code for your custom actions
`config.yml`	configuration of your NLU and dialogue models
`credentials.yml`	details for connecting to other services
`data/`**`nlu.md`**	your NLU training data
`data/`**`stories.md`**	your stories
`domain.yml`	your assistant's domain
`endpoints.yml`	details for connecting to channels like fb messenger
models/<timestamp>.tar.gz	your initial model

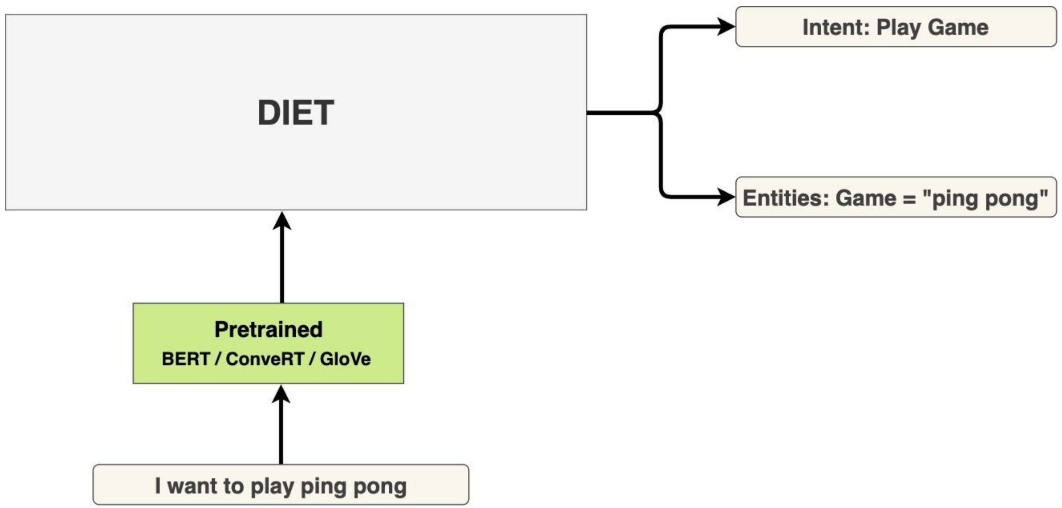
In other words



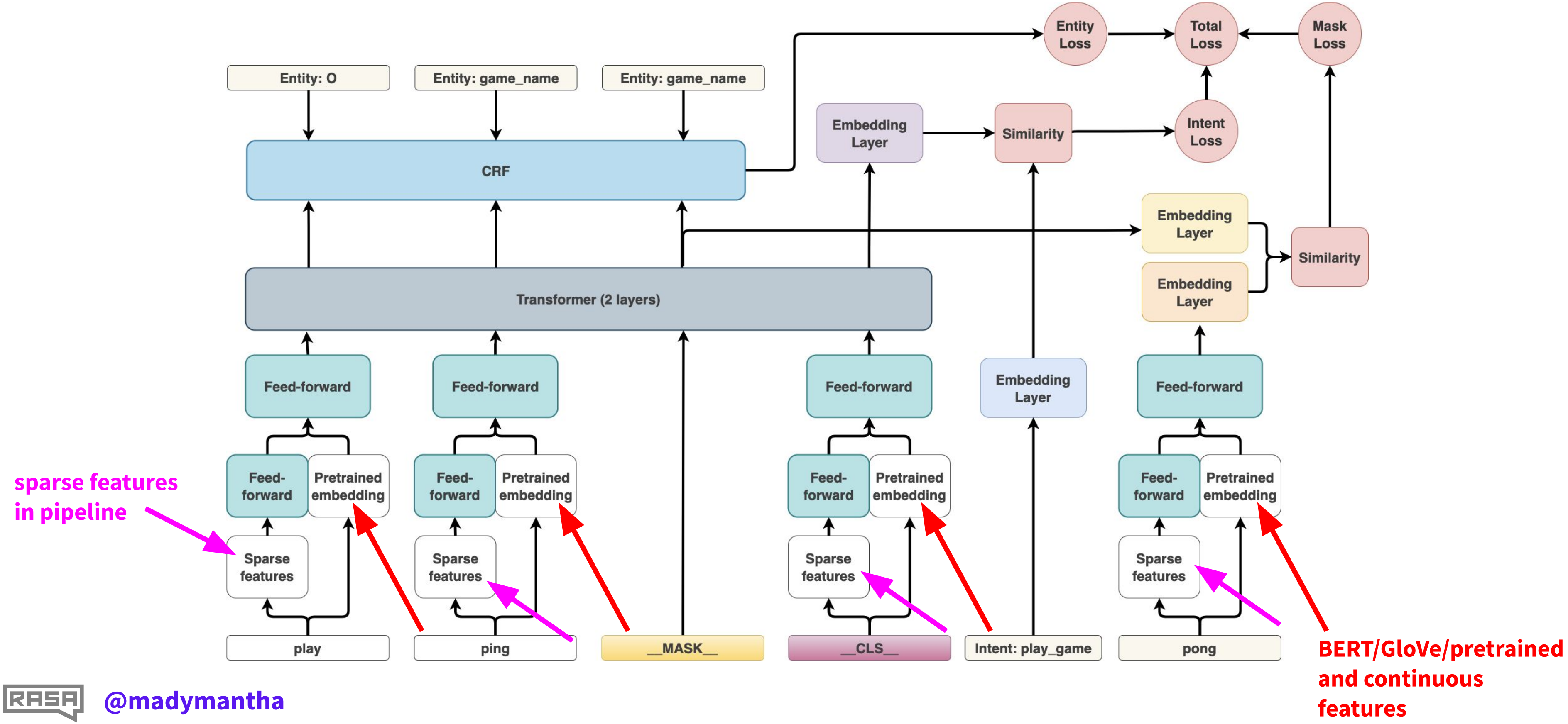
DIET is our new neural network architecture for NLU

What is DIET?

- New state of the art neural network architecture for NLU
- Predicts intents and entities together
- Plug and play pretrained language models



Feature: Customize what features go in!



How to use DIET in your Rasa project

Here's an example config.yml

Before the DIET model, you can specify any featurizer.

In our experiments, we use:

- Sparse features (aka no pre-trained model)
- GloVe (word vectors)
- BERT (large language model)
- ConveRT (pre-trained encoder for conversations)

```
language: en
pipeline:
- name: ConveRTTokenizer
- name: ConveRTFeaturizer
- name: DIETClassifier
  hidden_layers_sizes:
    text: [256, 128]
    label: []
  intent_classification: True
  entity_recognition: False
  use_masked_language_model: False
  BILOU_flag: False
  number_of_transformer_layers: 0
```

Experiments on the NLU-benchmark dataset

- Repo is on github
- Domain: human-robot interaction (smart home setting)
- 64 different intents
- 54 different entity types
- ~26k labelled examples

Previous state of the art:

- HERMIT NLU (Vanzo, Bastianelli, and Lemon @ SIGdial 2019)
- uses ELMo embeddings

Result 1: DIET outperforms SotA even without any pretrained embeddings

Previous state of the art: intent: 87.55 entities: 84.74

sparse	dense	mask loss	**Intent**	**Entities**
✓	✗	✗	87.10±0.75	83.88±0.98
✓	✗	✓	88.19±0.84	85.12±0.85
✗	GloVe	✗	89.20±0.90	84.34±1.03
✓	GloVe	✗	89.38±0.71	84.89±0.91
✗	GloVe	✓	88.78±0.70	85.06±0.84
✓	GloVe	✓	89.13±0.77	86.04±1.09
✗	BERT	✗	87.44±0.92	84.20±0.91
✓	BERT	✗	88.46±0.88	85.26±1.01
✗	BERT	✓	86.92±1.09	83.96±1.33
✓	BERT	✓	87.45±0.67	84.64±1.31
✗	ConveRT	✗	89.76±0.98	86.06±1.38
✓	ConveRT	✗	89.89±0.43	**87.38±0.64**
✗	ConveRT	✓	**90.15±0.68**	85.76±0.80
✓	ConveRT	✓	89.47±0.74	86.04±1.29

Result 2: GloVe embeddings perform better than BERT

sparse	dense	mask loss	**Intent**	**Entities**
✓	✗	✗	87.10±0.75	83.88±0.98
✓	✗	✓	88.19±0.84	85.12±0.85
✗	GloVe	✗	89.20±0.90	84.34±1.03
✓	GloVe	✗	89.38±0.71	84.89±0.91
✗	GloVe	✓	88.78±0.70	85.06±0.84
✓	GloVe	✓	89.13±0.77	86.04±1.09
✗	BERT	✗	87.44±0.92	84.20±0.91
✓	BERT	✗	88.46±0.88	85.26±1.01
✗	BERT	✓	86.92±1.09	83.96±1.33
✓	BERT	✓	87.45±0.67	84.64±1.31
✗	ConveRT	✗	89.76±0.98	86.06±1.38
✓	ConveRT	✗	89.89±0.43	**87.38±0.64**
✗	ConveRT	✓	**90.15±0.68**	85.76±0.80
✓	ConveRT	✓	89.47±0.74	86.04±1.29

Result 3: ConveRT embeddings perform best on the NLU-benchmark dataset

sparse	dense	mask loss	**Intent**	**Entities**
✓	✗	✗	87.10±0.75	83.88±0.98
✓	✗	✓	88.19±0.84	85.12±0.85
✗	GloVe	✗	89.20±0.90	84.34±1.03
✓	GloVe	✗	89.38±0.71	84.89±0.91
✗	GloVe	✓	88.78±0.70	85.06±0.84
✓	GloVe	✓	89.13±0.77	86.04±1.09
✗	BERT	✗	87.44±0.92	84.20±0.91
✓	BERT	✗	88.46±0.88	85.26±1.01
✗	BERT	✓	86.92±1.09	83.96±1.33
✓	BERT	✓	87.45±0.67	84.64±1.31
✗	ConveRT	✗	89.76±0.98	86.06±1.38
✓	ConveRT	✗	89.89±0.43	**87.38±0.64**
✗	ConveRT	✓	**90.15±0.68**	85.76±0.80
✓	ConveRT	✓	89.47±0.74	86.04±1.29

Result 4: DIET outperforms fine-tuning BERT

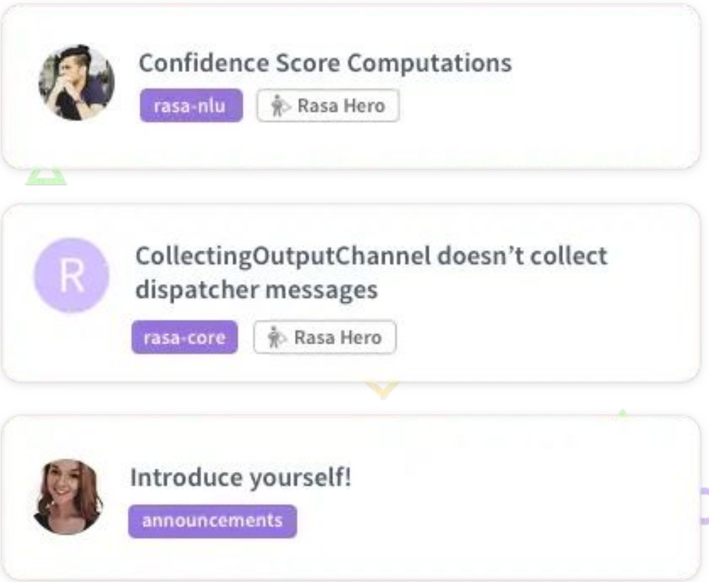
		Intent	Entities
Fine-tuned BERT	F1	89.67±0.48	85.73±0.91
	R	89.67±0.48	84.71±1.28
	P	89.67±0.48	86.78±1.02
sparse + ConveRT†	F1	**89.89±0.43**	**87.38±0.64**
	R	**89.89±0.43**	**87.15±0.97**
	P	**89.89±0.43**	**87.62±0.94**

Which featurizer is best depends on your dataset, so try different ones!

We don’t believe in “one size fits all” machine learning

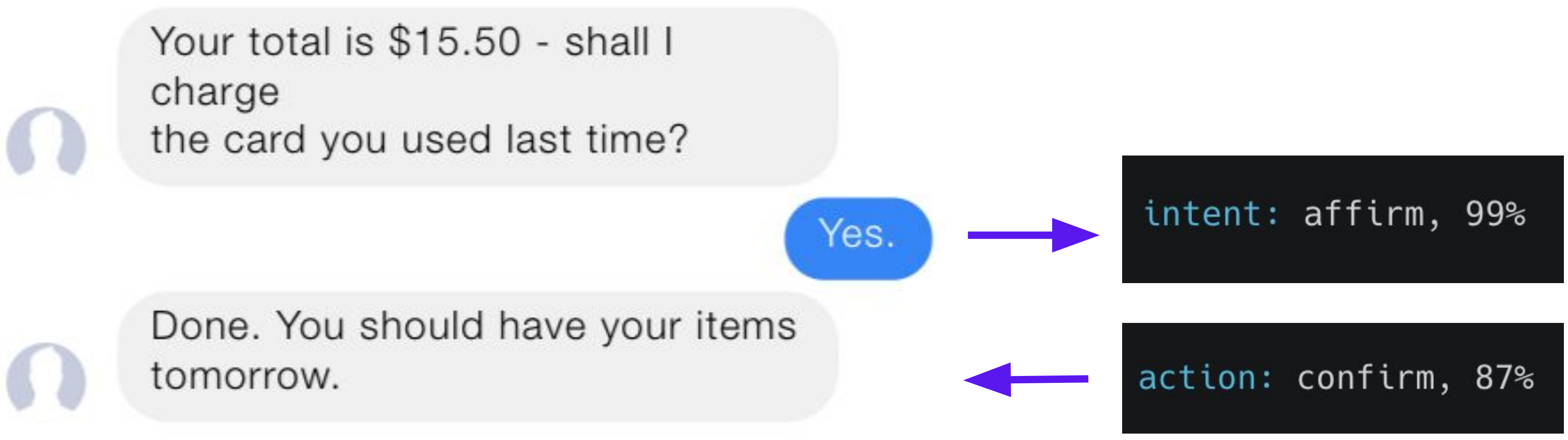
- We aim to provide sensible defaults and suggestions
- BUT even more important that these models are easy to customize

Share your results and compare notes with 8000+ developers at **forum.rasa.com**



Transformer Embedding Dialogue policy (TED)

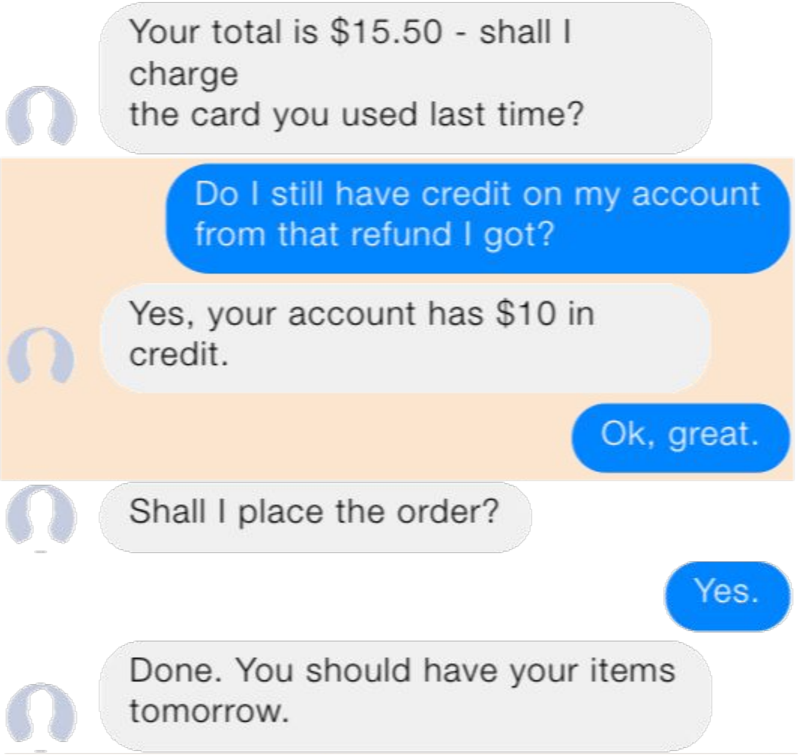
Conversational AI requires NLU and Dialogue management



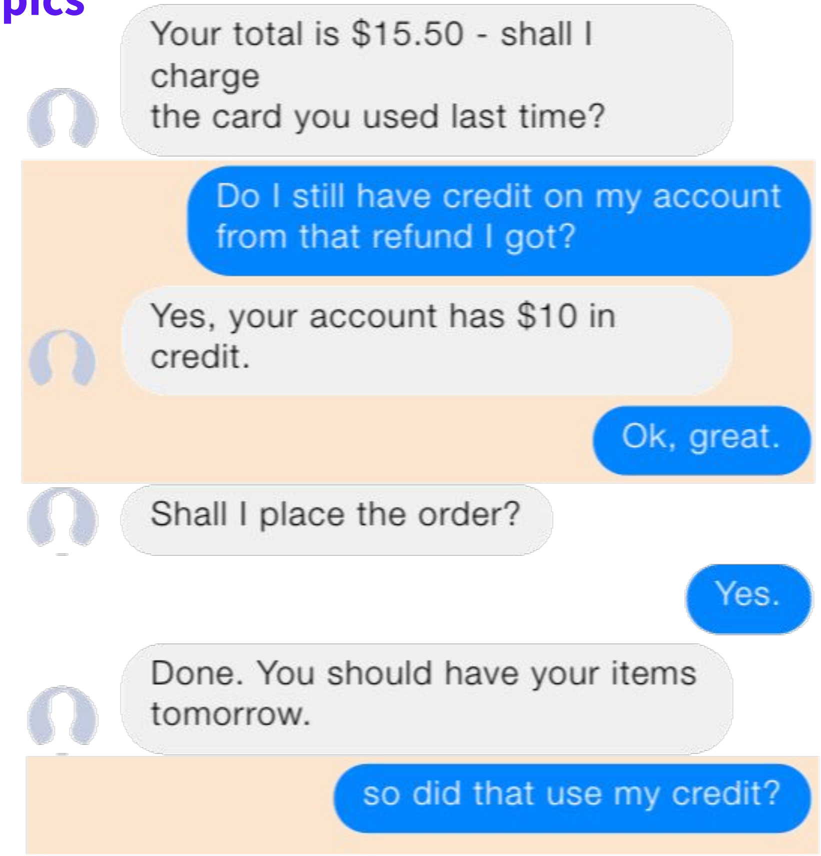
Happy paths are already solved

```python
class CheckoutForm(FormAction):
    """Perform steps for checkout"""

    def name(self):
        return "checkout_form"

    @staticmethod
    def required_slots(tracker):
        return ["address", "card_number", "shipping_method"]
```

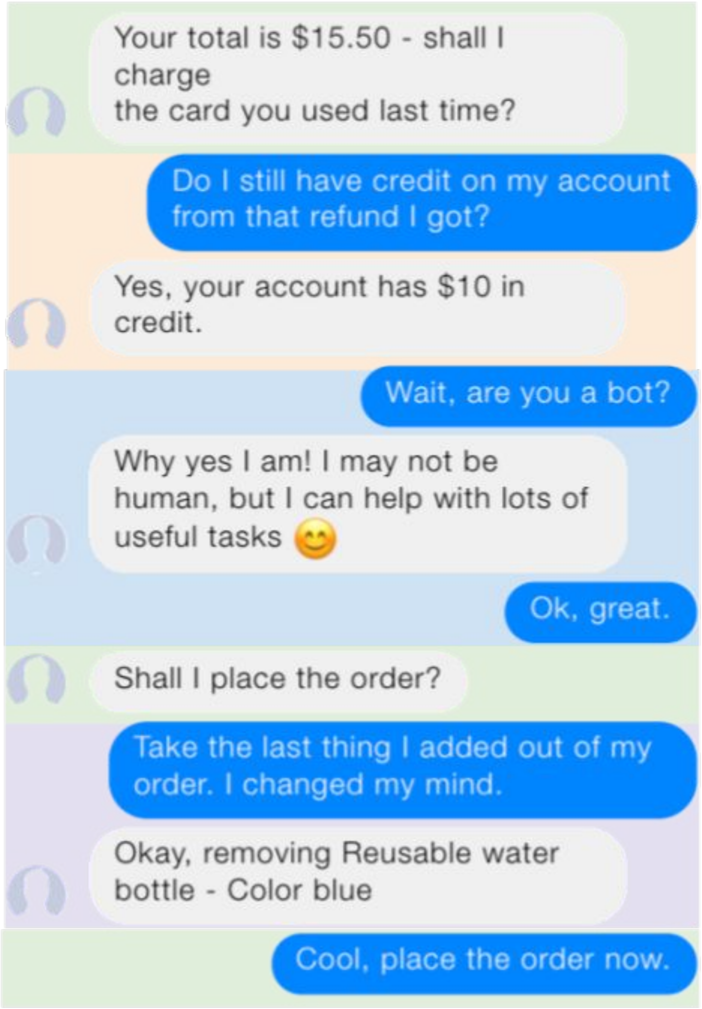
You can’t predict user behavior



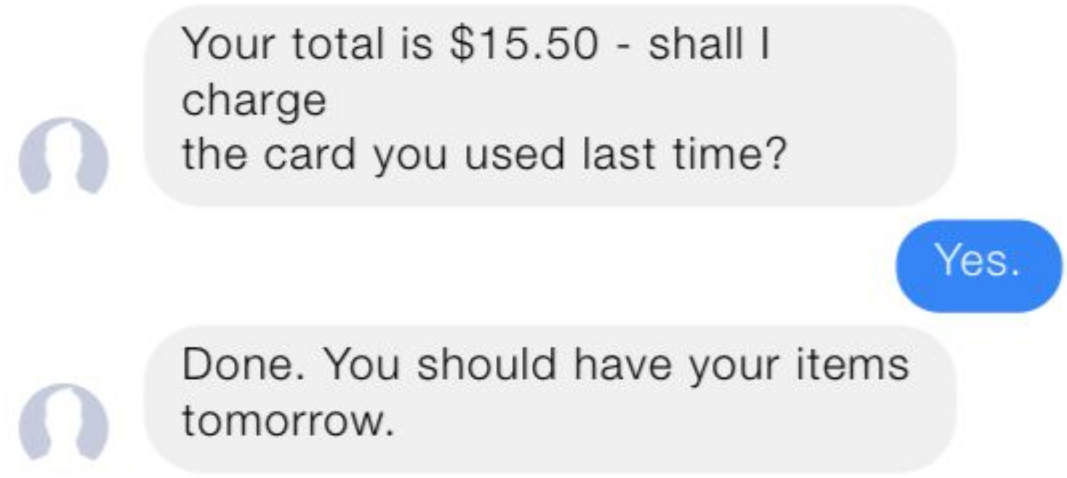
Users will interject and loop back to earlier topics

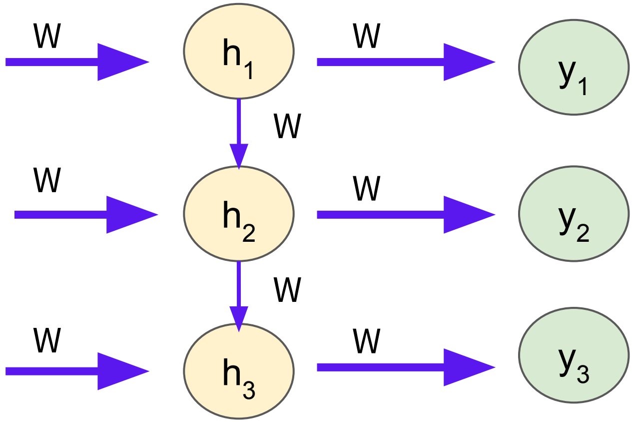


Real conversations don't follow the happy path

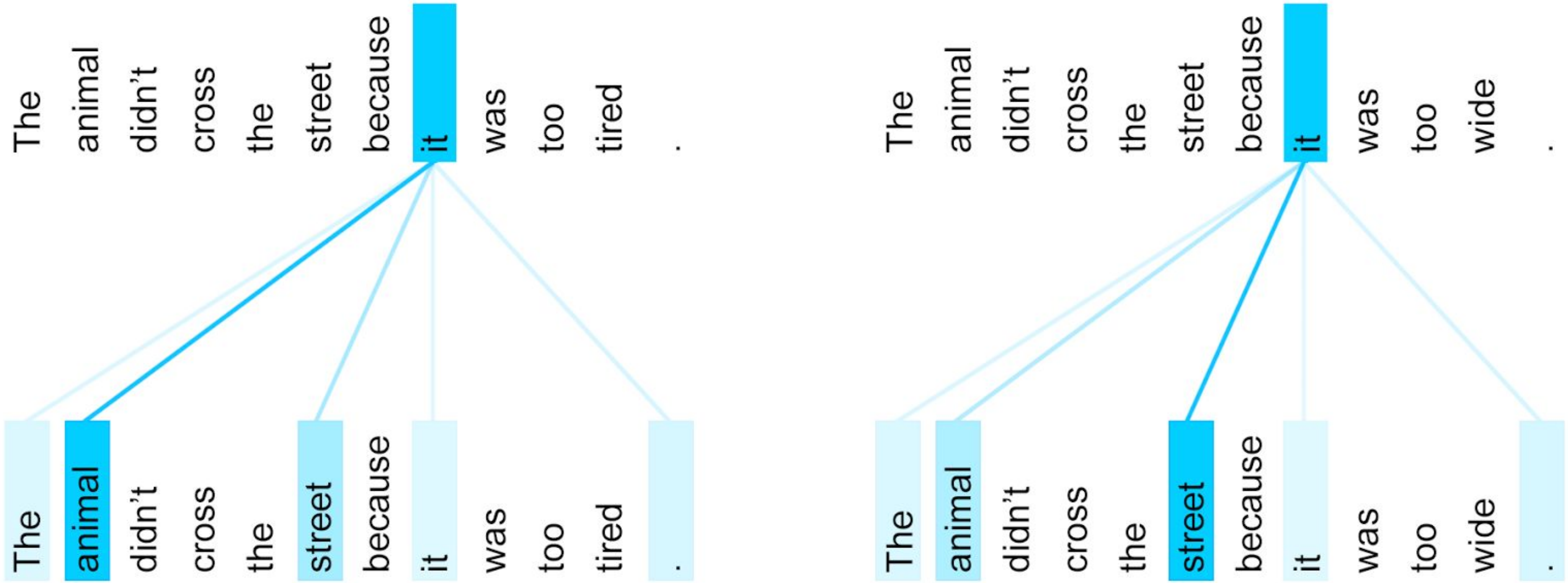


People typically use a recurrent neural net (RNN) to model dialogue





But not all input should be treated equally



https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Transformers (AKA self-attention) are now state of the art for many tasks

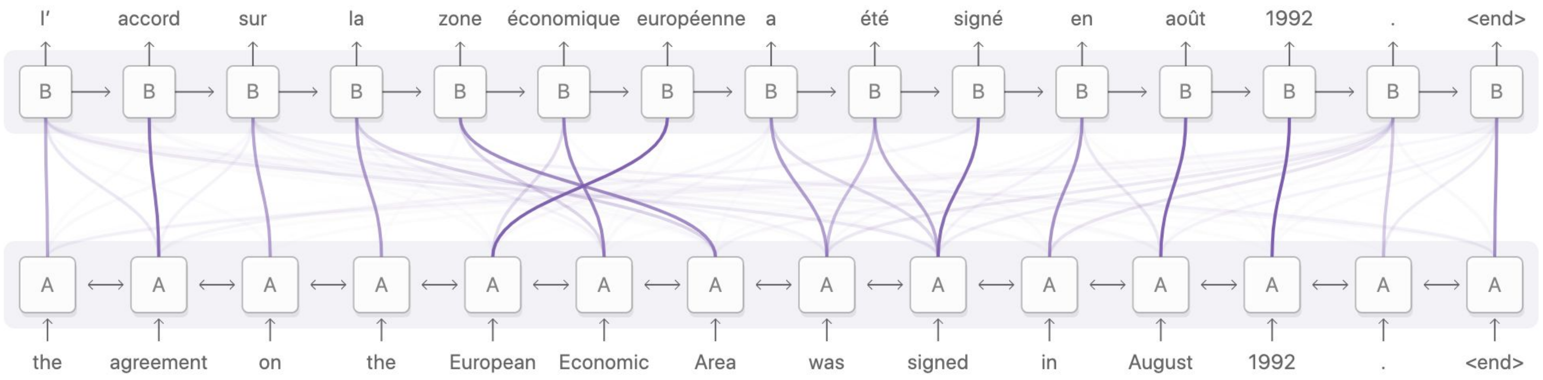
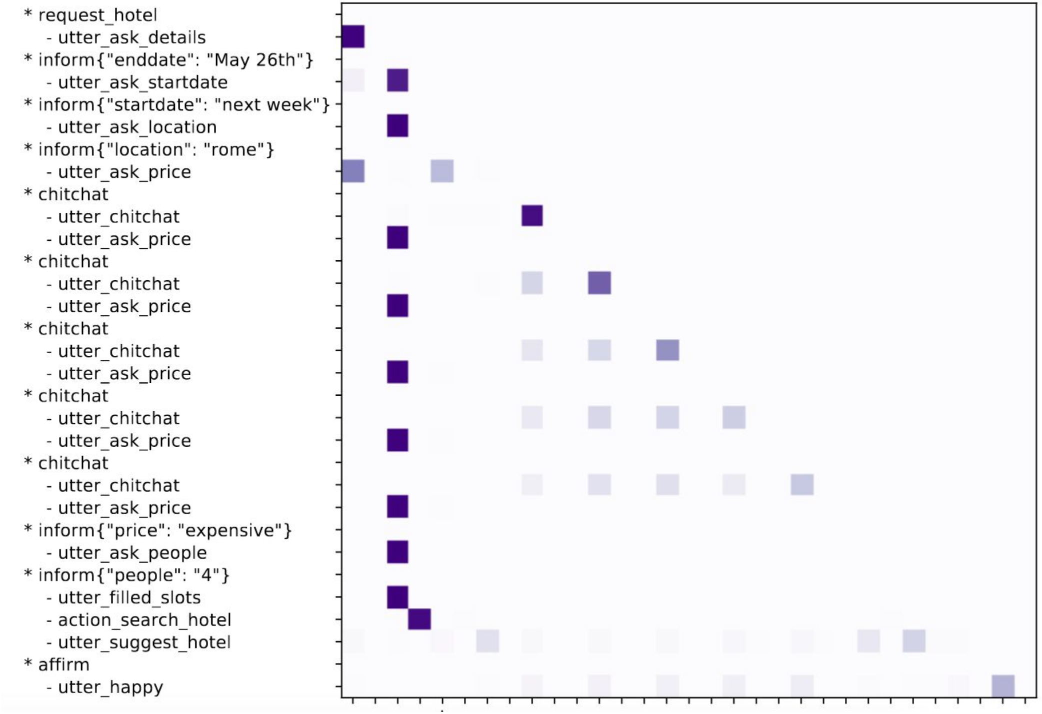


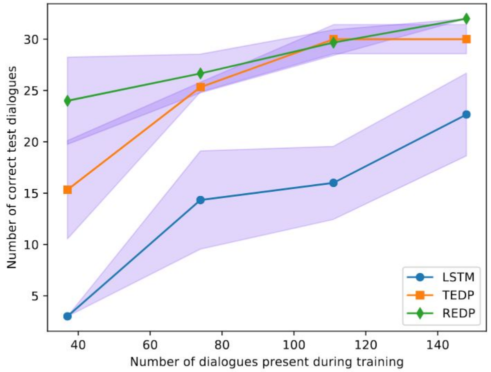
Diagram derived from Fig. 3 of Bahdanau, *et al.* 2014

https://distill.pub/2016/augmented-rnns/

We found out that the Transformer Embedding Dialogue policy can untangle sub-dialogues



[paper]



Livecoding

Let’s create an action together!

Testing

Use the Rasa CLI to test your assistant

End to End Evaluation

Run through test conversations to make sure that both NLU and Core make correct predictions.

```
$ rasa test
```

NLU Evaluation

Split data into a test set or estimate how well your model generalizes using cross-validation.

```
$ rasa test nlu -u
data/nlu.md --config
config.yml
--cross-validation
```

Core Evaluation

Evaluate your trained model on a set of test stories and generate a confusion matrix.

```
$ rasa test core
--stories
test_stories.md --out
results
```

Interactive Learning: Talk to your bot yourself

- **Correct** your bot's predictions as you go

- Save conversations as **training stories** or **E2E stories**

Run `rasa test` locally when building a minimum viable assistant

- Test your model after training to make development more productive and reliable

```
#### This file contains tests to evaluate that your bot behaves as expected.
#### If you want to learn more, please see the docs: https://rasa.com/docs/ra

## happy path 1
* greet: hello there!
  - utter_greet
* mood_great: amazing
  - utter_happy

## happy path 2
* greet: hello there!
  - utter_greet
* mood_great: amazing
  - utter_happy
* goodbye: bye-bye!
  - utter_goodbye

## sad path 1
* greet: hello
  - utter_greet
* mood_unhappy: not good
  - utter_cheer_up
  - utter_did_that_help
* affirm: yes
  - utter_happy
```

```
Your Rasa model is trained and saved at '/Users/ty/Documents/product_mgmt/docs/rasa/tem
(rasa_env) BERMB00017:temp ty$ rasa test
Processed Story Blocks: 100%|
2020-03-24 14:24:31 INFO     rasa.core.test  - Evaluating 7 stories
Progress:
100%|
2020-03-24 14:24:32 INFO     rasa.core.test  - Finished collecting predictions.
2020-03-24 14:24:32 INFO     rasa.core.test  - Evaluation Results on END-TO-END level:
2020-03-24 14:24:32 INFO     rasa.core.test  -  Correct:          7 / 7
2020-03-24 14:24:32 INFO     rasa.core.test  -  F1-Score:         1.000
2020-03-24 14:24:32 INFO     rasa.core.test  -  Precision:        1.000
2020-03-24 14:24:32 INFO     rasa.core.test  -  Accuracy:         1.000
2020-03-24 14:24:32 INFO     rasa.core.test  -  In-data fraction: 0.943
2020-03-24 14:24:32 INFO     rasa.core.test  - Evaluation Results on ACTION level:
2020-03-24 14:24:32 INFO     rasa.core.test  -  Correct:          35 / 35
2020-03-24 14:24:32 INFO     rasa.core.test  -  F1-Score:         1.000
2020-03-24 14:24:32 INFO     rasa.core.test  -  Precision:        1.000
2020-03-24 14:24:32 INFO     rasa.core.test  -  Accuracy:         1.000
2020-03-24 14:24:32 INFO     rasa.core.test  -  In-data fraction: 0.943
2020-03-24 14:24:32 INFO     rasa.core.test  -  Classification report:
                     precision    recall  f1-score   support
```

Livecoding

Interactive learning + testing

Recap...and what we’re going to do tomorrow!

- DIET: multi-task transformer architecture for NLU
- TED: transformer architecture for dialogue
- Actions
- Testing