

Rasa

Haotian Ye

16-12-2021

Outline

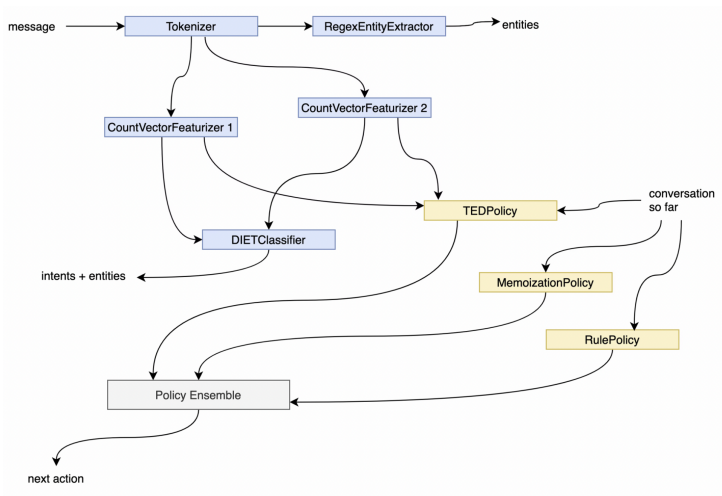
- 1 Design Overview
- 2 NLU
- 3 Dialogue Management
- 4 NLG
- 5 Summary
- 6 Literature & Sources

- **Rasa Open Source** (prev. Rasa NLU and Rasa Core)
- Decoupled, modular design
- Each component independently deployable
- End-to-end introduced in 2.x¹

¹Rasa Blog [1, 2]

- NLU in Rasa: intent + entities
- Pipelined: comprised of NLP modules (**components**)
- New graph-like structure in 3.x²

²Rasa Blog [3]

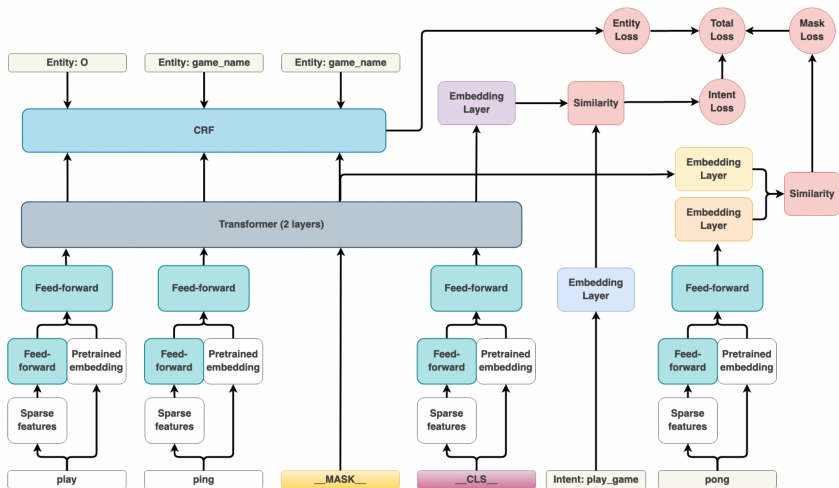


Rasa Blog [3]

Example Pipeline

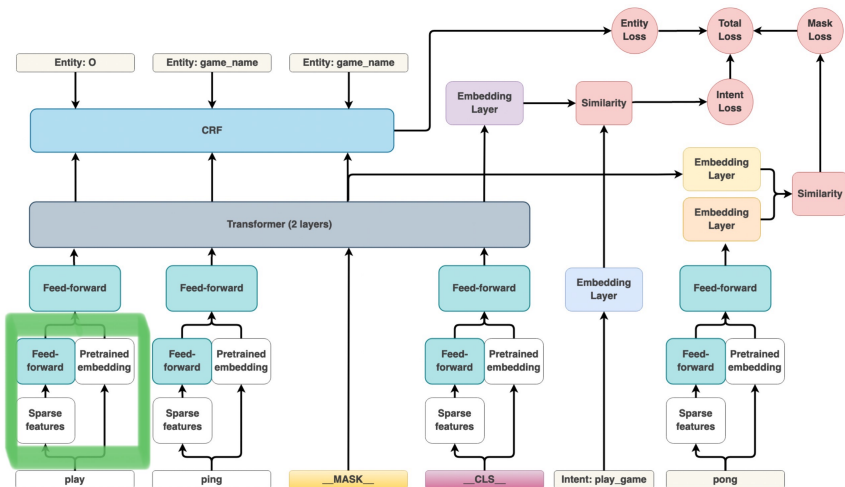
```
pipeline:
- name: SpacyNLP
  model: "en_core_web_md"
- name: SpacyTokenizer
- name: SpacyEntityExtractor
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100
  num_transformer_layers: 6
  embedding_dimension: 100
  entity_recognition: False
```

- “Dual Intent and Entity Transformer”
- Intent classification + entity recognition in one structure
- Word order aware (vs BOW)
- Combines dense + sparse features from input



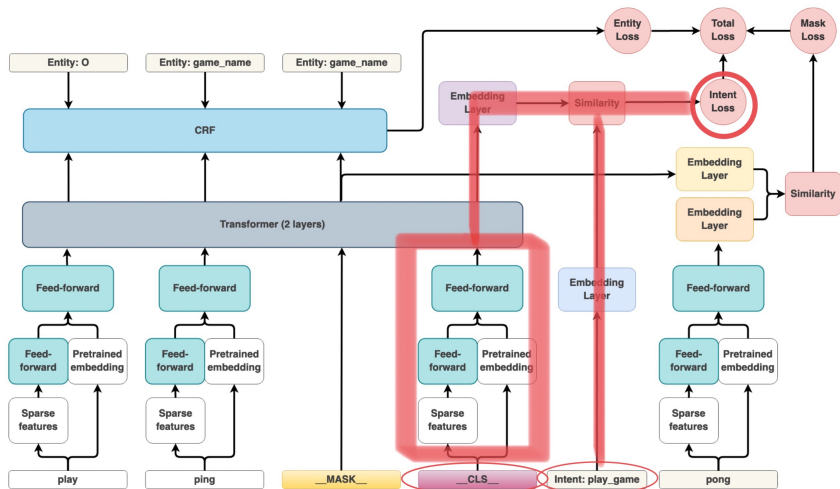
Bunk et al. (2020)

DIET Features



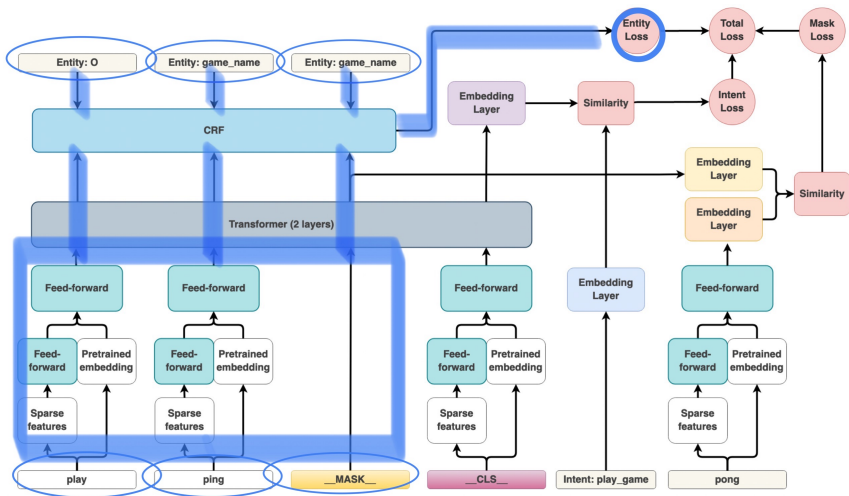
Bunk et al. (2020)

DIET Intent Classification



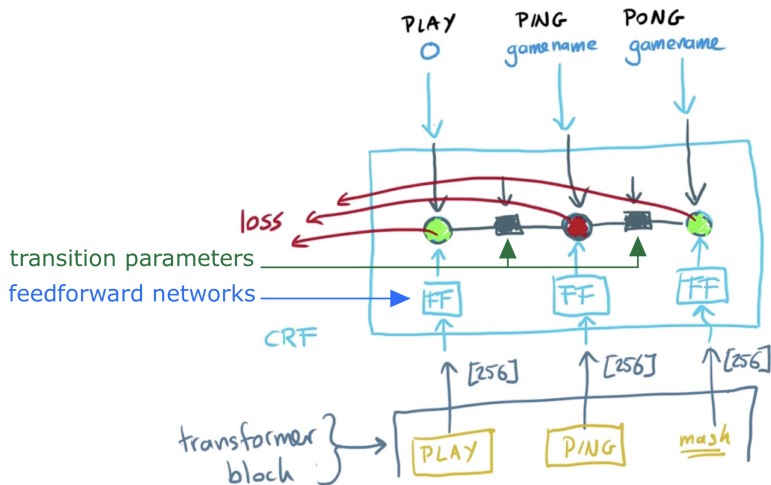
Bunk et al. (2020)

DIET Entity Recognition



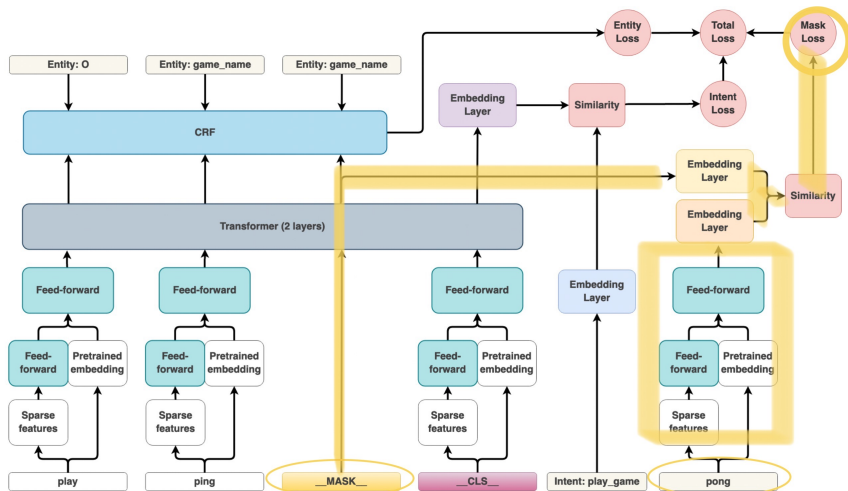
Bunk et al. (2020)

DIET CRF



Rasa YouTube [2]

DIET Masked Language Modeling



Bunk et al. (2020)

Dialogue Management (prev. Rasa Core)

- Stand-alone component
- Easier adaptation to e.g. other languages
- Possible to use with other NLU systems

- Maintains the dialogue **state**
 - slot information, previous action(s), etc.
- One per dialogue session
- Passes on information to **policies**
- Gets feedback from executed actions

- Makes decisions about next actions
- **Featurizers** → encode the dialogue state
 - current status of slots
 - intents/entities
 - previous action(s)
- Policies can be rule-based or ML-based
- Multiple policies are ranked by confidence and priorities

- Rule Policy
 - makes decisions based on user-defined rules
- Memoization Policy
 - compares last `max_history` turns with training stories
 - outputs exact next action if match is found
- TED Policy³
 - encodes state(s) and one-hot action embeddings
 - calculates similarities

³Vlasov et al. (2020)

Example: Rules

- Always have to follow the same path
- Lack the ability to generalize

```
rules:  
- rule: Greet whenever the user greets  
  steps:  
  - intent: greet  
  - action: utter_greet
```

Example: Stories

```
stories:  
- story: book a ticket  
  steps:  
  - intent: greet  
  - action: utter_greet  
  - intent: book_ticket  
  - action: ...  
  ...  
  - intent: goodbye  
  - action: utter_goodbye
```

TED Features

"I'd like a pizza."

intent

• $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

order

slot

• $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

address

entity

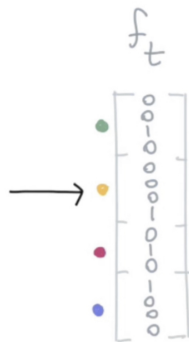
• $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

pizza

prev
action

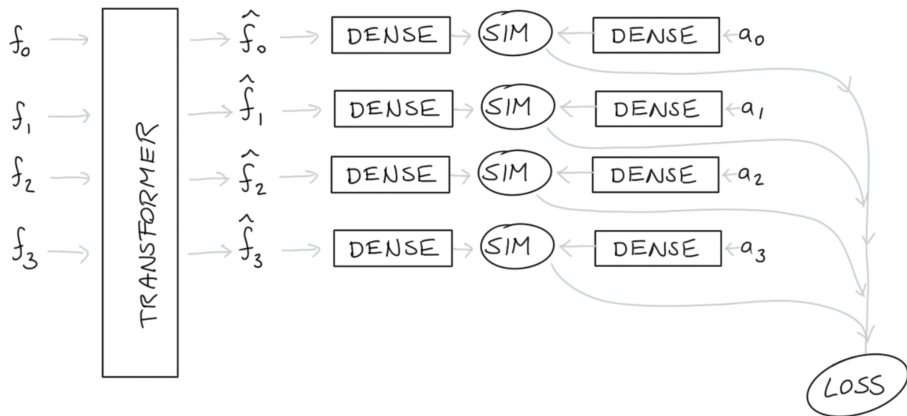
• $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

listen



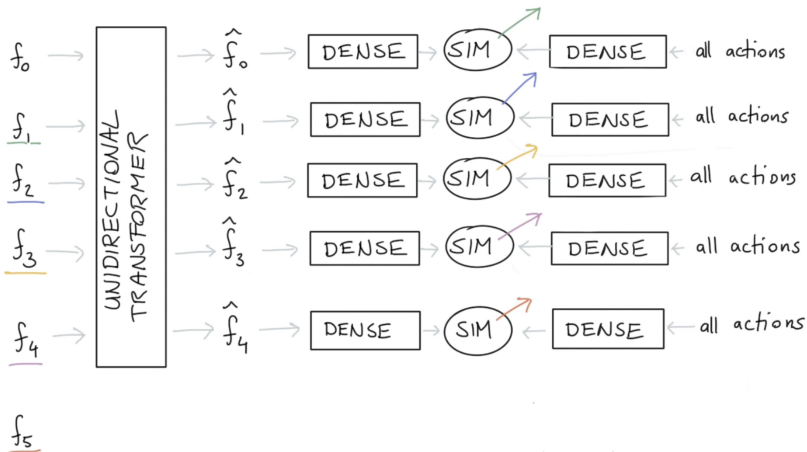
Rasa YouTube [1]

TED Training Loss



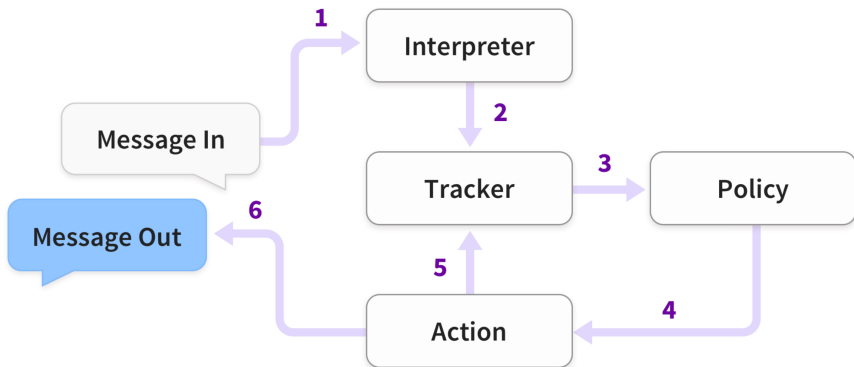
Rasa YouTube [1]

TED Prediction



Rasa YouTube [1]

Design Overview



Bocklisch et al. (2017)

Template-Based NLG

- Using Rasa's default response templates
- Types of responses defined in the domain file (`utter_`)

- Also possible to:
 - allow variations in responses
 - use slot values in responses

```
responses:  
  utter_greet:  
    - text: "Hi there!"  
    - text: "Hey, {name}. How are you?"  
  utter_bye:  
    - text: "See you!"
```

NN-Based Response Selection

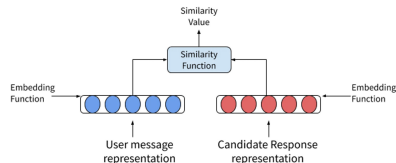
- **Response selectors**

delegate inputs with specific intents to different response models

- Potentially more consistent and precise responses
- End-to-end feature

pipeline:

```
...  
- name: "ResponseSelector"  
  retrieval_intent: chitchat  
- name: "ResponseSelector"  
  retrieval_intent: faq
```



Rasa Blog [4]

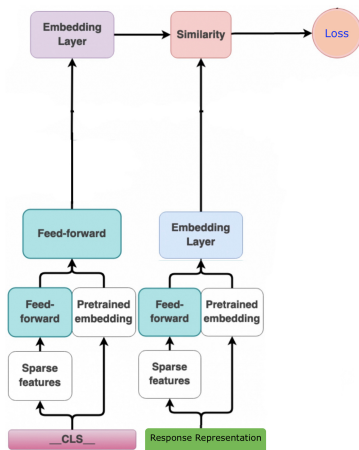
Response Selector

```
nlu:
- intent: ask_faq/sub_intent1
  examples: |
    - ...
- intent: ask_faq/sub_intent2
  examples: |
    - ...

responses:
  utter_faq/sub_intent1:
    - text: ...
  utter_faq/sub_intent2:
    - text: ...

rules:
- rule: respond to FAQs
  steps:
    - intent: ask_faq
    - action: utter_faq
```

NN-Based Response Selection



- CLS as input representation
- Compared with candidate response representation

- **SC-LSTM**⁴

- *Semantically Controlled LSTM*
- introduces **DA cell** with a **reading gate**
- cell regulates info needed for later time steps (DAs)

- **SC-GPT**⁵

- GPT-2⁶ style LM pre-training
- (general) dialogue act-response pair pre-training
- (domain-specific) DA-response fine-tuning

- Advantages

- natural, varying responses
- generalizable (new domains, new users)

⁴Wen et al. (2015)

⁵Peng et al. (2020)

⁶Radford et al. (2019)

Summary

- Pipeline
- DIET
 - intent classification
 - entity recognition
 - masked language modeling
- Tracker and policies
- Response generation

- Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management.
- Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A. (2020). Diet: Lightweight language understanding for dialogue systems.
- Peng, B., Zhu, C., Li, C., Li, X., Li, J., Zeng, M., and Gao, J. (2020). Few-shot natural language generation for task-oriented dialog.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Vlasov, V., Mosig, J. E. M., and Nichol, A. (2020). Dialogue transformers.
- Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems.

Rasa Blog

- 1 <https://rasa.com/blog/its-about-time-we-get-rid-of-intents>
- 2 <https://rasa.com/blog/were-a-step-closer-to-getting-rid-of-intents>
- 3 <https://rasa.com/blog/bending-the-ml-pipeline-in-rasa-3-0>
- 4 <https://rasa.com/blog/response-retrieval-models>

Rasa Youtube

- 1 <https://youtu.be/j90NvurJI4I>
- 2 <https://youtu.be/vWStcJDuOUk>