Towards Self-Learning Chatbots

Pierre Pakey, CTO **Dimitri Lozeve**, Data Scientist

pierre@destygo.com
dimitri@destygo.com

March 27, 2019

Outline

1. Our vision for the future of conversational assistants
2. Towards self-learning chatbots
3. Putting RL into production: technical pipeline
4. The models
5. Conclusion

Our vision for the future of conversational assistants

Destygo

- conversational assistants for automating consumer relations
- SaaS platform, no technical knowledge required from the customer
- available for all languages
- travel and transport industries
  → Rail, Airlines, Airports, Travel Agencies...

Our model for chatbots

- customer defines some use-cases as intents
- customer uploads examples for each intent, used as the base data to train the models
- answers are generated based on the intent and on the results of API calls to retrieve the relevant information

Current challenges

- striving for language- and domain-agnostic models to improve generalisation
- state-of-the-art entity recognition and language understanding
- smaller intervention from customer
- verticalization: multiple agents sharing data among similar use cases

... and self-learning!

Towards self-learning chatbots

Reinforcement Learning

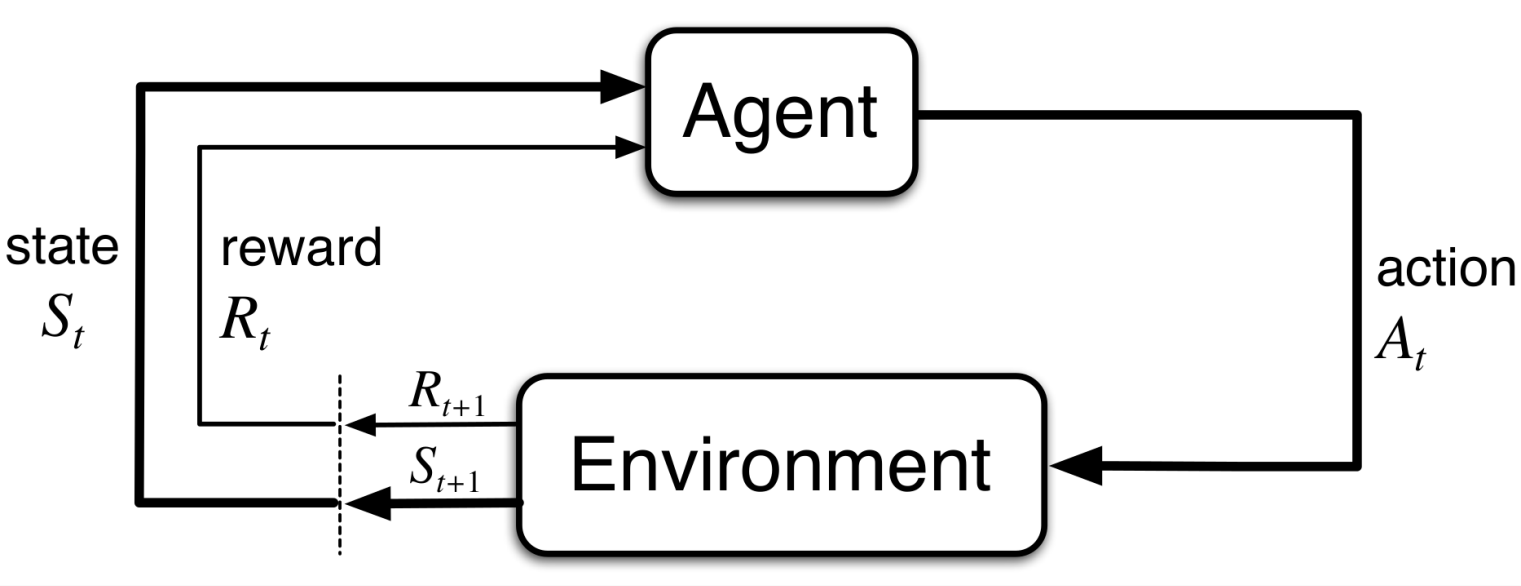


Figure 1: Agent-Environment interaction[1]

[1]Sutton and Barto 2018.

What is the link between RL theory and chatbots?

- Model of an agent
  → The chatbot agent
- Behaviour of an agent
  → How to compute a chatbot's optimal behaviour?

What is an agent?

Definition (Markov Decision Process)

- set of states $\mathcal{S}$
- $\mathcal{A}(s)$ of possible actions for state s
- set of rewards $\mathcal{R} \subset \mathbb{R}$
- function p representing the dynamics of the MDP

$$p(s', r \mid s, a) := \mathbb{P}(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a)$$

- notion of episodes

The chatbot as an RL agent

The chatbot MDP

state	user message
actions	user intent
dynamics	not known (model-free)
episodes	conversations

Agent behaviour

Policy:

$\pi(s, a)$ = probability of selecting action a when in state s.

Value function:

$$q_\pi(s, a) := \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$$

$$\text{where } G_t := \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad \text{(return)}.$$

Optimal behaviour:

$$q_*(s, a) = \sum_{s', r} p(s', r \mid s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right].$$

(Bellman optimality equation)

Challenges of RL for NLP and chatbots

- No possible model of the environment (the way people talk)
  → cannot possibly simulate conversations to evaluate and improve our policies
  → different from other applications (Atari games, Go), where it is possible to simulate the environment to generate large amounts of data
- Environment is constantly changing (changes in bot's configuration)

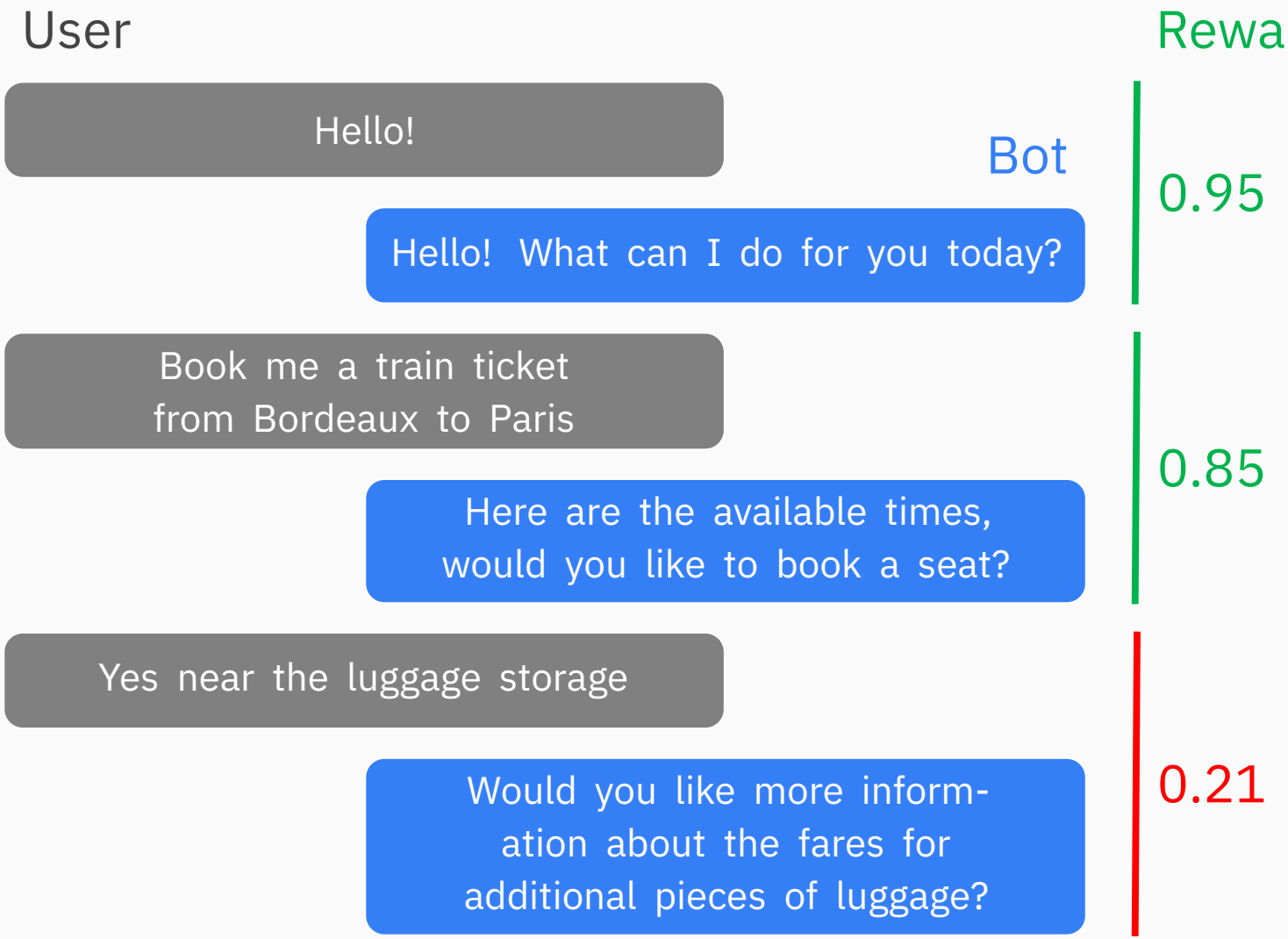
Putting RL into production: technical pipeline

Base hypotheses and prerequisites (1/2)

Base policy:

- Goal: for each user message, predict an intent
- set of possible intents determined by configuration
- Two steps:
  - Named Entity Recognition (“tomorrow at 5”, “Gare St-Lazare”, “Air France”)
  - Natural Language Understanding for Intent Classification

Base hypotheses and prerequisites (2/2): Reward function



Microservice architecture

Why microservices?[2]

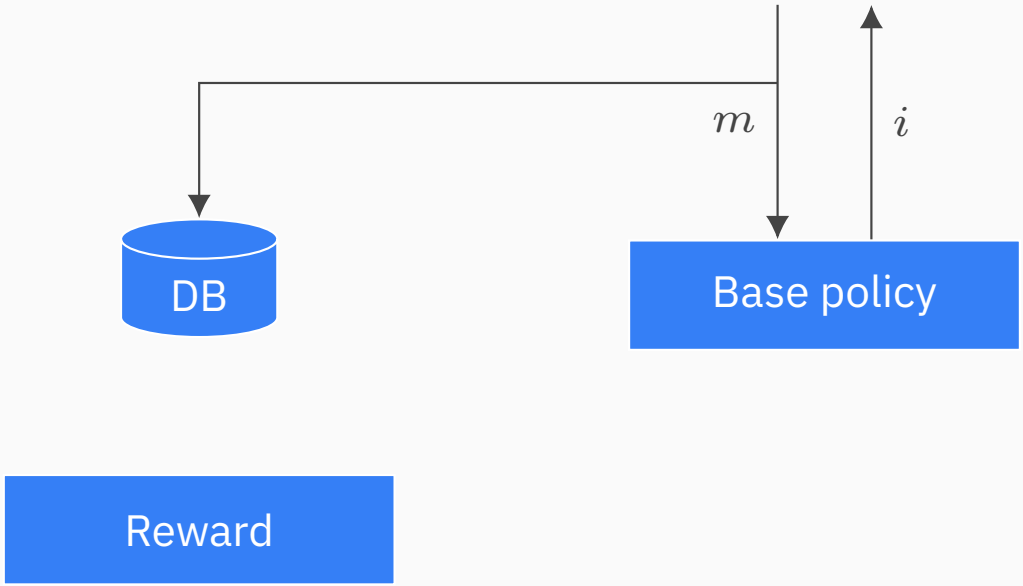
- general architectural policy at Destygo
- small, composable functionality
- increased robustness and scalability
- varied technologies: Ruby on Rails (web backend), Vue JS (frontend), and Python (ML backend)
- aligned with team organization

Communication between microservices

- REST API for configuration, message-passing
- queues for messages

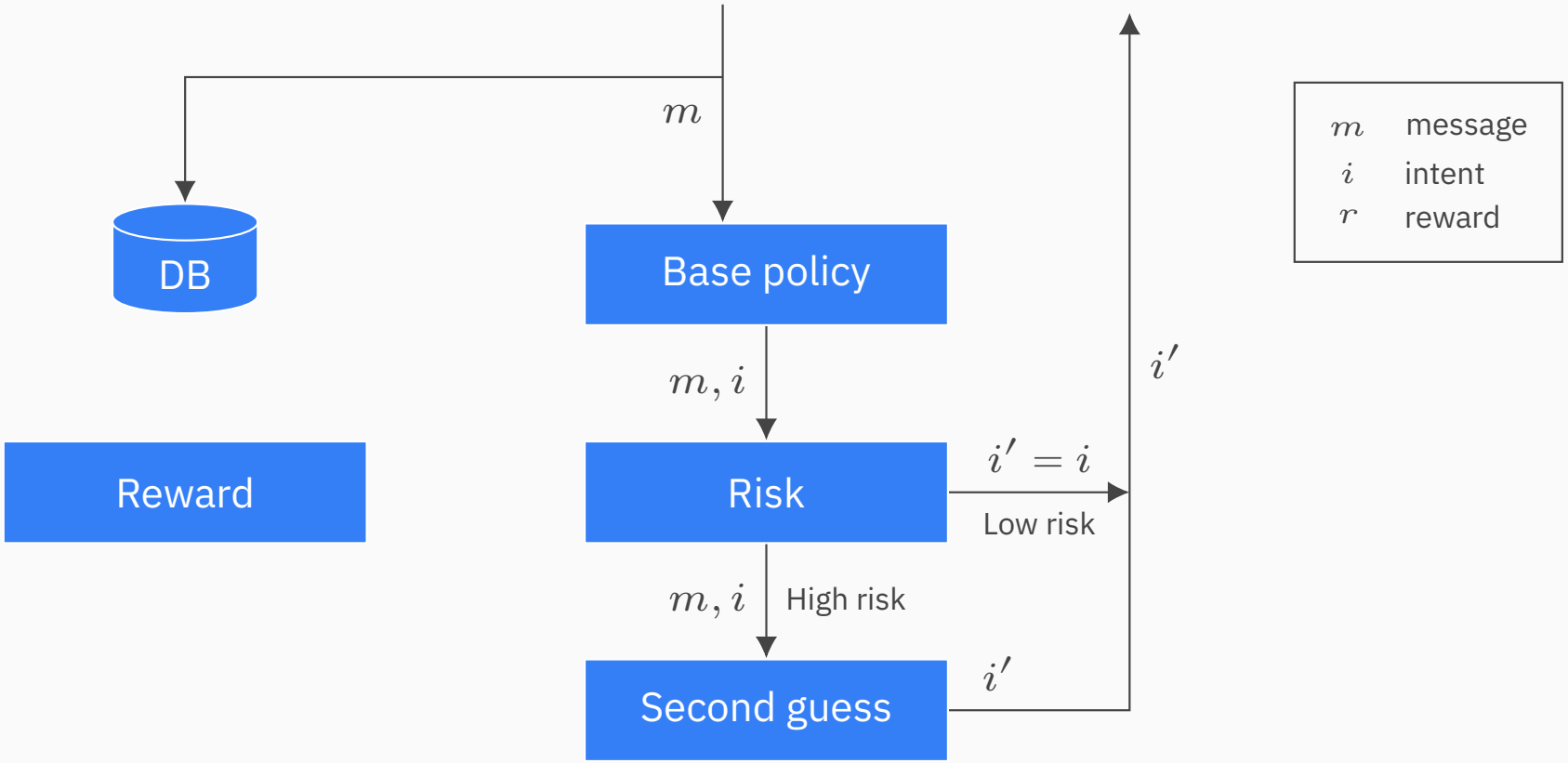
[2] Newman 2015.

Prediction and training pipeline

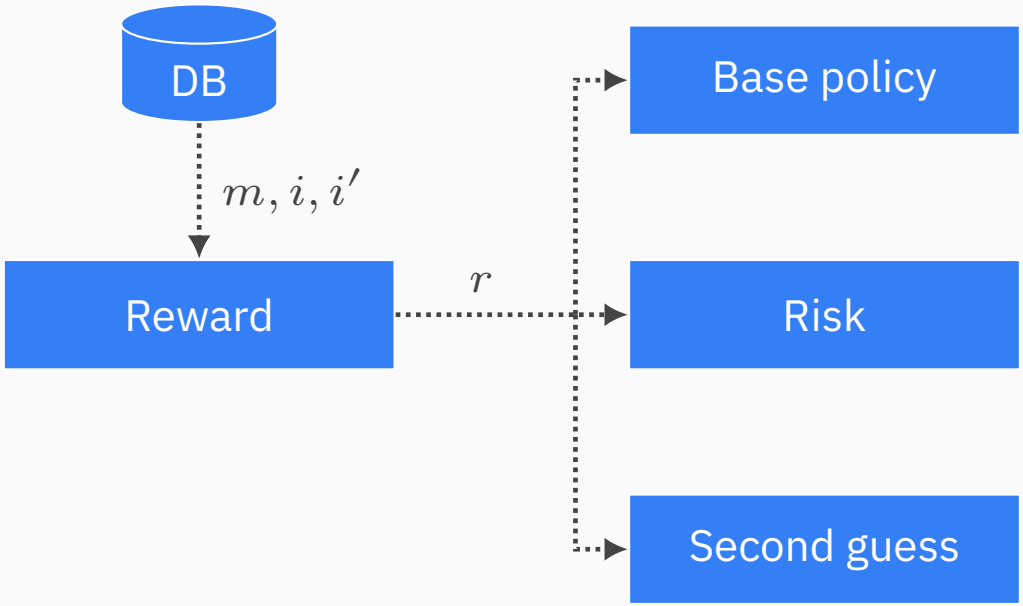


m	message
i	intent
r	reward

Prediction and training pipeline

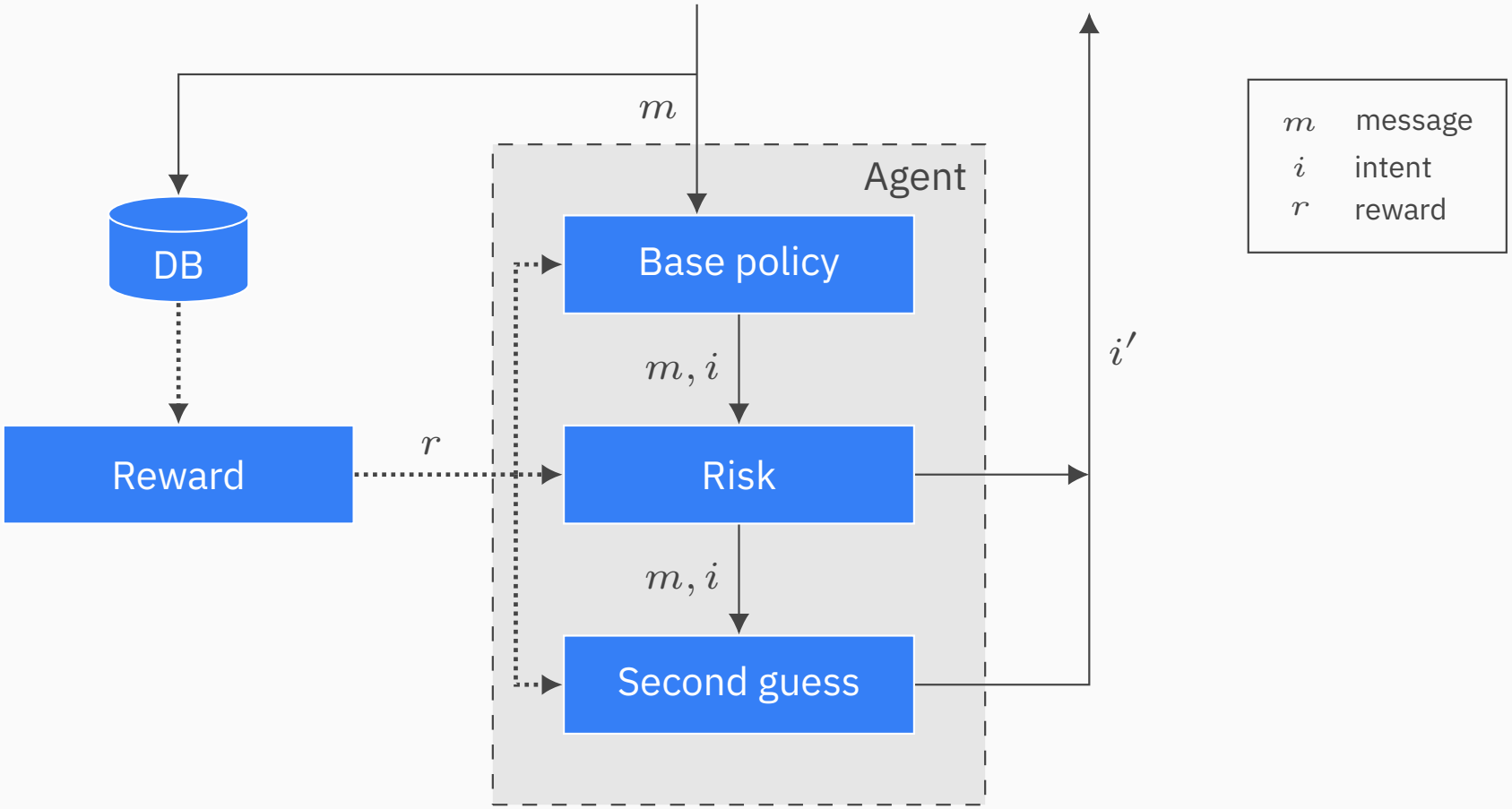


Prediction and training pipeline



m	message
i	intent
r	reward

Prediction and training pipeline



Why do we split the agent in 3?

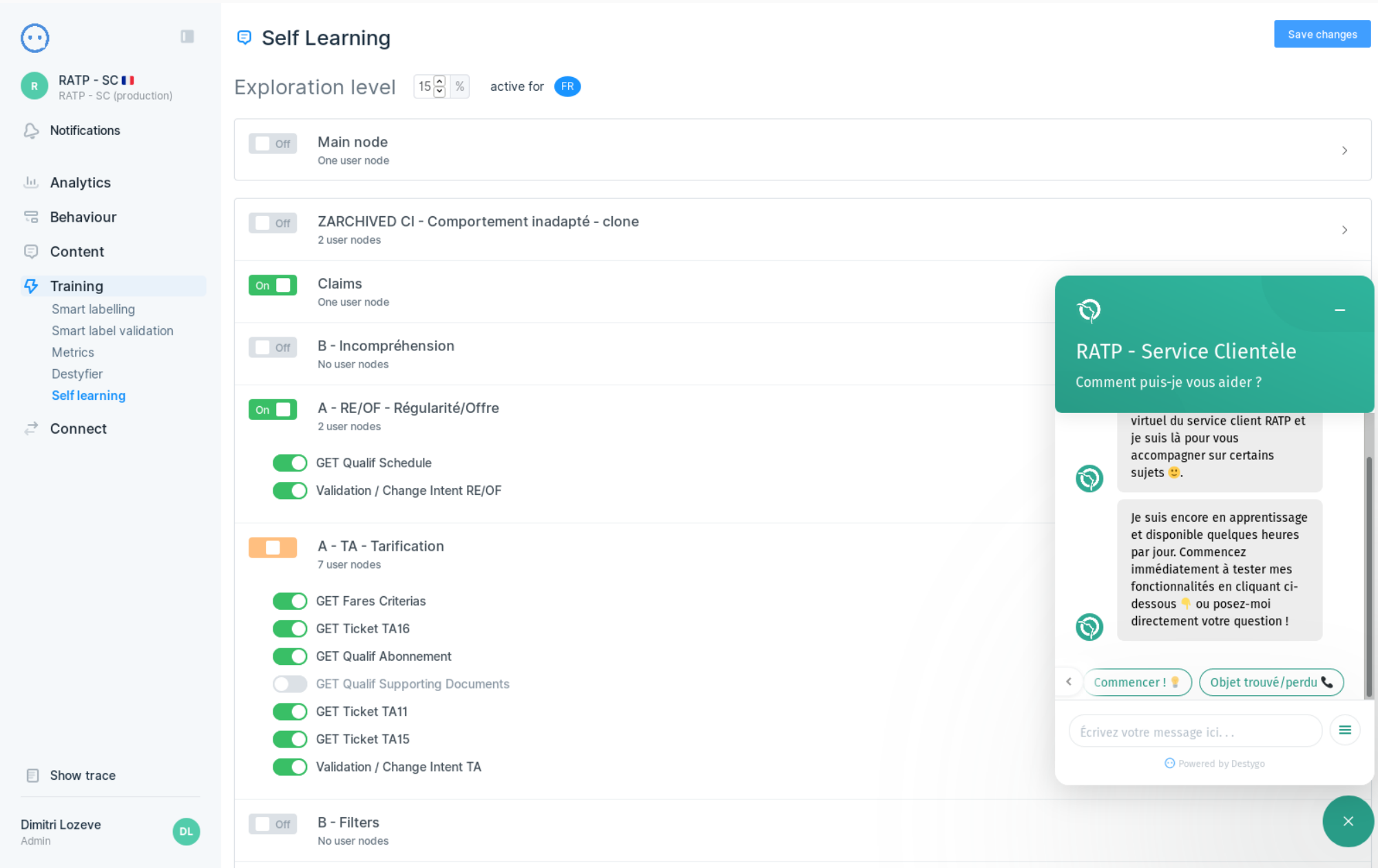
Data requirements

- Second guess based on heuristics
  → fewer learnable elements
- Less data required for convergence

Maintenance

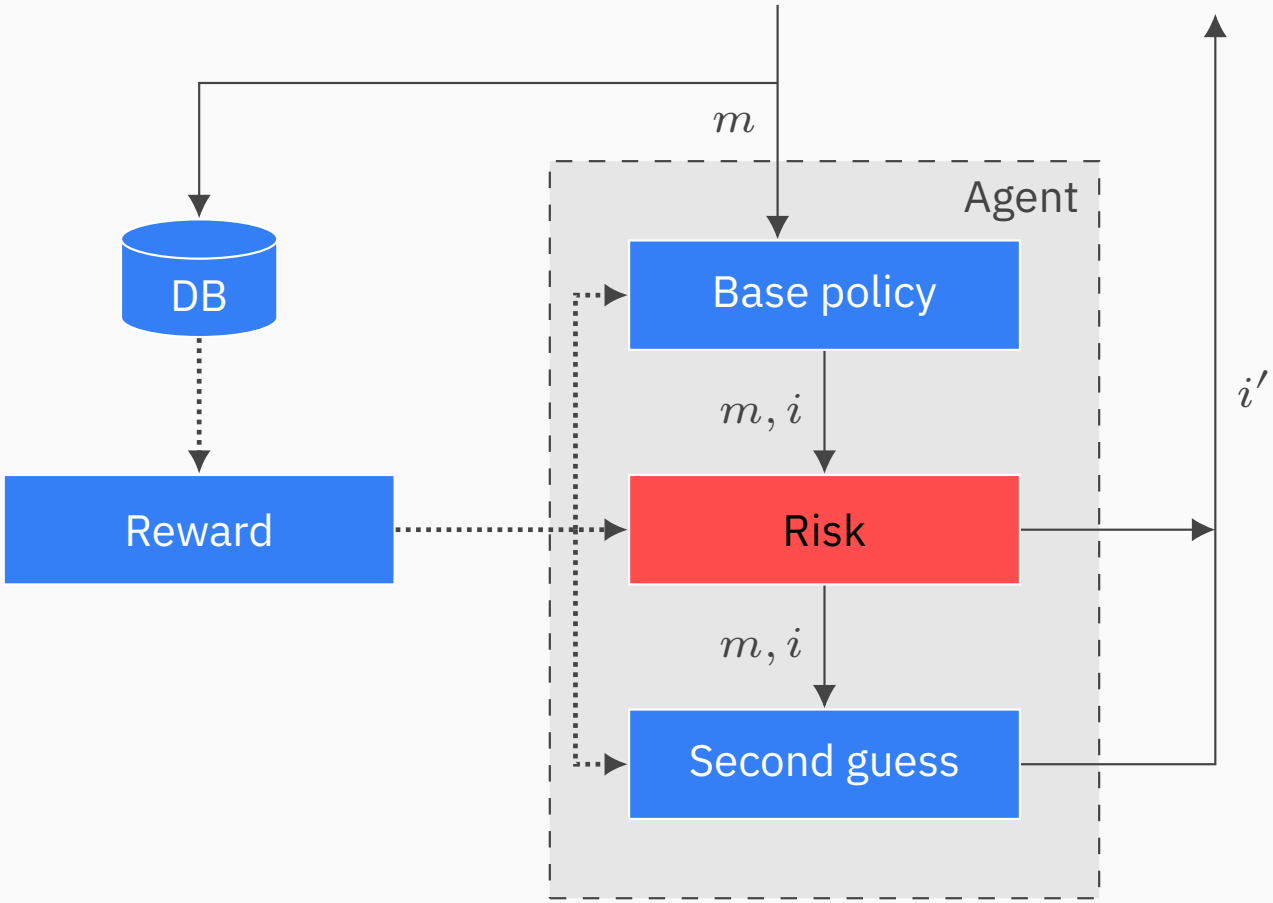
- Easier to reason about the role of each service
- Prediction pipeline is more resilient (fallback to base model if problem occurs downstream)
- Models can evolve independently

Configuration and monitoring pipeline



The models

Focus on risk



Models

First approach (approximation of true RL)

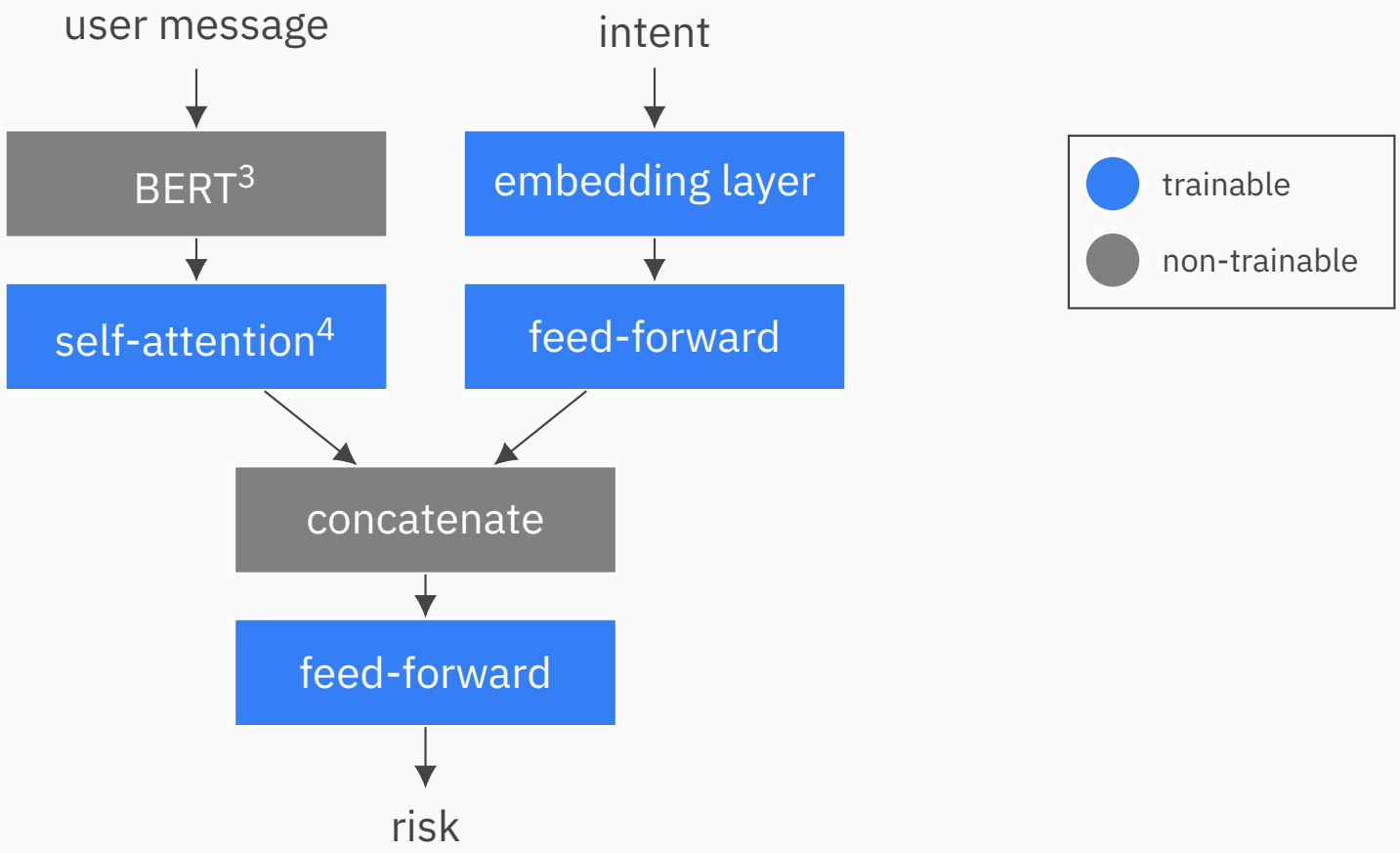
Message-Intent Risk (MIR)

- input: user message text + intent ID
- output: probability of “failure”

Second Guess

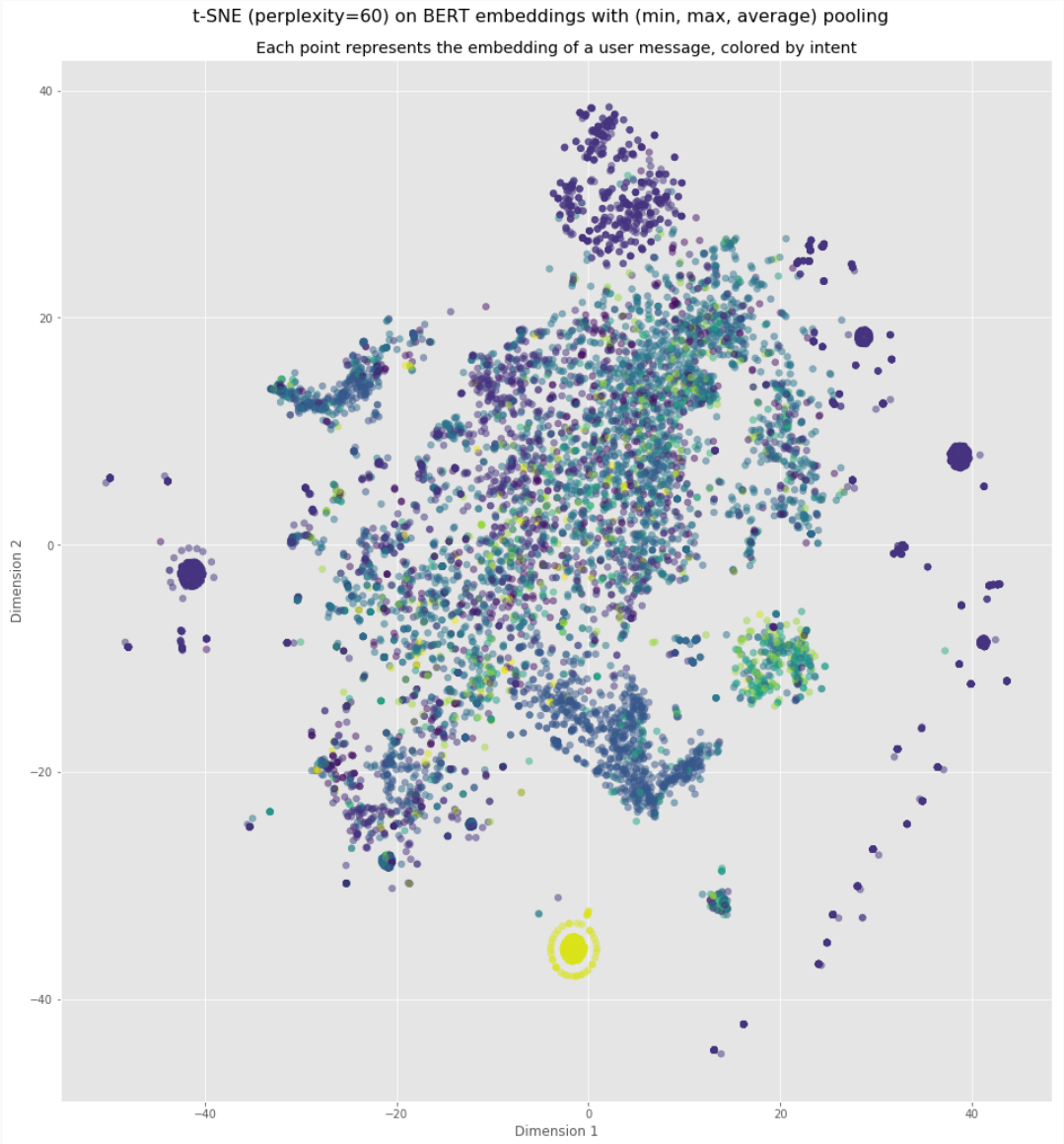
- given a high-risk message, choose the “next-best” intent

Risk model implementation



[3] Devlin et al. 2018.
[4] Parikh et al. 2016.

BERT embeddings



destygo

Risk model training and performance

Training

- TensorFlow 1.12 + Keras, on Python 3.6
- GPU for training and inference, on a `p3.2xlarge` instance
- Training is ~3 hours on an NVIDIA Tesla V100!

Performance on data labelled with the reward

Accuracy	0.857
F_1 score	0.855
AUC	0.812

Limitations of the current approach

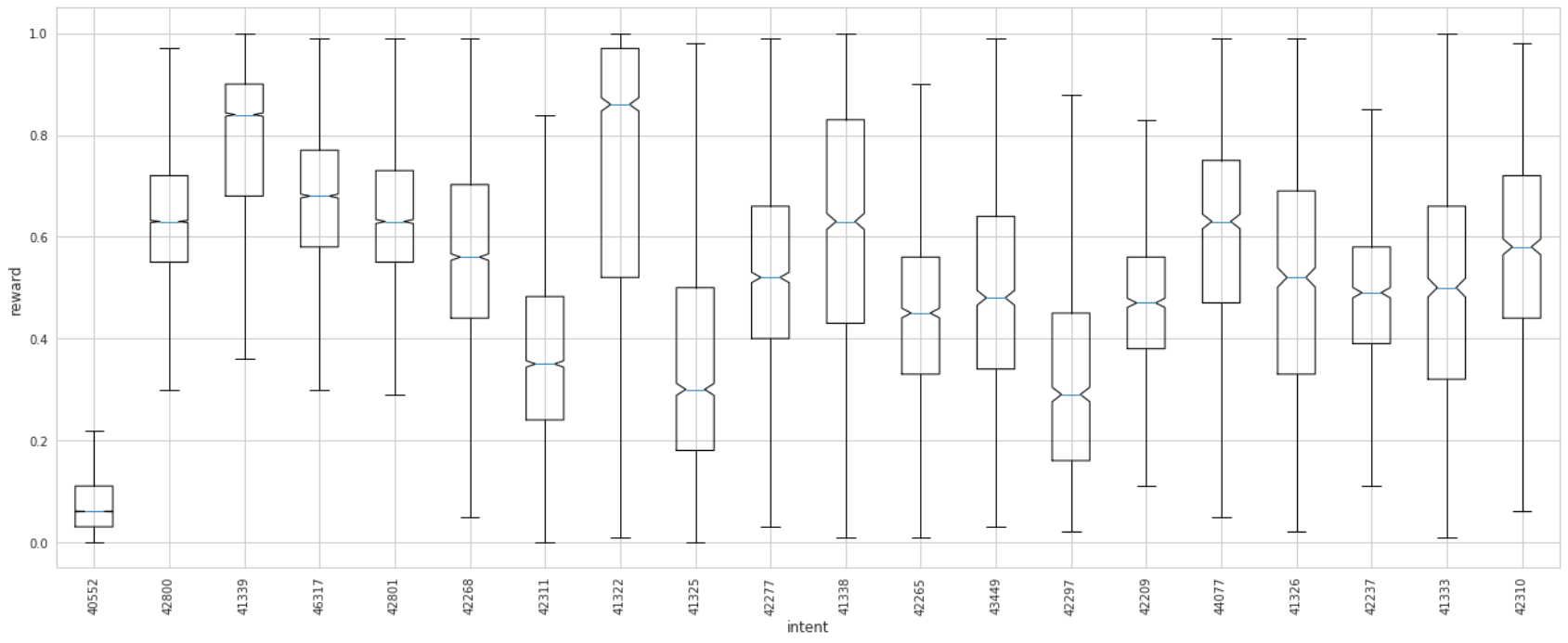


Figure 2: Distribution of the reward for some frequent intents

Model will be biased on some intents!

The real problem: state space representation

- Natural language: infinite dimension (depends on the embedding)
- Lots of implicit knowledge
- Choice of the embedding is crucial, plenty of options
  - Other embeddings: ElMo,[5] OpenAI GPT (2?)
  - Sentence-level embeddings: min-max-average pooling,[6] Universal Sentence Encoder[7]

[5] Peters et al. 2018.
[6] Arora, Liang, and Ma 2016.
[7] Cer et al. 2018.

Solution: change the representation?

The whole conversation as state

- state = all previous messages in the conversation
- represent more accurately the “stage” of the conversation: meaningful representation of current vs. next state
- captures more diversity
- but: more difficult to fit (limited amount of data)

Change action space representation

Embed intents with examples of bot messages and BERT embeddings

Towards real Q-learning

Only an approximation!

- Theory: Risk estimation can estimate the value of any given state
- Practice: Risk model cannot be used directly to determine the optimal action (not enough data)
- Splitting the agent in 3 is a good short-term approximation, as it requires less data

Problem: instantaneous reward maximization

- Instantaneous reward (on pairs of message) cannot capture information about whether the user's goal is achieved

```
I am tired
            Why do you say you are tired?
Because I did not sleep well
            Is it because you did not sleep well
            that you came to me?[8]
```

- Need to add long-term reward, at the conversation level
- Less data required to decide whether we helped the user

[8] ELIZA (`M-x doctor`)

Future evolutions for risk model training

- Build target function iteratively on (s, a, r, s') data:

$$Y_k^Q = r + \gamma \max_i \left(1 - \text{Risk}(s', i)\right).$$

  → Risk model trained on this new target, approximating the next state's value

- Optimizations well-studied in an abundant Deep RL literature[9]
  → go closer to the current state-of-the-art

[9] Arulkumaran et al. 2017; Francois-Lavet et al. 2018; Li 2018.

Conclusion

Conclusion: relevance of RL for chatbots at Destygo

- RL should be an essential part of the development of any conversational agent[10]
- Very few companies working on self-learning today
- State-of-the-art advances in many different applications (NLP, RL)

[10] Hancock et al. 2019.

We’re hiring!

- Research Scientist: experienced academic researchers in NLP and/or RL, lead ML research at Destygo
- Software Engineers: experienced with microservice architectures and interested in ML
- Product Managers: platform and R&D
- Site Reliability Engineer
- Sales Representatives

`www.destygo.com/jobs`

Thank you!

References

- Arora, Sanjeev, Yingyu Liang, and Tengyu Ma (Nov. 4, 2016). “A Simple but Tough-to-Beat Baseline for Sentence Embeddings”. In: URL: https://openreview.net/forum?id=SyK00v5xx (visited on 01/30/2019).
- Arulkumaran, Kai et al. (Nov. 2017). “A Brief Survey of Deep Reinforcement Learning”. In: *IEEE Signal Processing Magazine* 34.6, pp. 26–38. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2743240. arXiv: 1708.05866. URL: http://arxiv.org/abs/1708.05866 (visited on 11/23/2017).
- Cer, Daniel et al. (Mar. 29, 2018). “Universal Sentence Encoder”. In: *arXiv:1803.11175 [cs]*. arXiv: 1803.11175. URL: http://arxiv.org/abs/1803.11175 (visited on 02/24/2019).
- Devlin, Jacob et al. (Oct. 10, 2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805 [cs]*. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 02/21/2019).
- Francois-Lavet, Vincent et al. (2018). “An Introduction to Deep Reinforcement Learning”. In: *Foundations and Trends® in Machine Learning* 11.3, pp. 219–354. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000071. arXiv: 1811.12560. URL: http://arxiv.org/abs/1811.12560 (visited on 01/05/2019).
- Hancock, Braden et al. (Jan. 16, 2019). “Learning from Dialogue after Deployment: Feed Yourself, Chatbot!” In: *arXiv:1901.05415 [cs, stat]*. arXiv: 1901.05415. URL: http://arxiv.org/abs/1901.05415 (visited on 01/18/2019).
- Li, Yuxi (Oct. 15, 2018). “Deep Reinforcement Learning”. In: *arXiv:1810.06339 [cs, stat]*. arXiv: 1810.06339. URL: http://arxiv.org/abs/1810.06339 (visited on 01/16/2019).
- Newman, Sam (2015). *Building microservices: designing fine-grained systems*. First Edition. OCLC: ocn881657228. Beijing Sebastopol, CA: O’Reilly Media. 259 pp. ISBN: 978-1-4919-5035-7.
- Parikh, Ankur P. et al. (June 6, 2016). “A Decomposable Attention Model for Natural Language Inference”. In: *arXiv:1606.01933 [cs]*. arXiv: 1606.01933. URL: http://arxiv.org/abs/1606.01933 (visited on 11/06/2018).
- Peters, Matthew E. et al. (Feb. 15, 2018). “Deep contextualized word representations”. In: URL: https://arxiv.org/abs/1802.05365 (visited on 11/01/2018).
- Schmitt, Alexander and Stefan Ultes (Nov. 2015). “Interaction Quality: Assessing the quality of ongoing spoken dialog interaction by experts—And how it relates to user satisfaction”. In: *Speech Communication* 74, pp. 12–36. ISSN: 01676393. DOI: 10.1016/j.specom.2015.06.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167639315000679 (visited on 03/07/2019).
- Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement learning: an introduction*. Second edition. Adaptive computation and machine learning series. Cambridge, MA: The MIT Press. ISBN: 978-0-262-03924-6.
- Ultes, Stefan et al. (Aug. 20, 2017). “Domain-Independent User Satisfaction Reward Estimation for Dialogue Policy Learning”. In: *Interspeech 2017*. Interspeech 2017. ISCA, pp. 1721–1725. DOI: 10.21437/Interspeech.2017-1032. URL: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1032.html (visited on 03/07/2019).