

Outline

Morning program

- Preliminaries

- Modeling user behavior

- Semantic matching

- Learning to rank

Afternoon program

- Entities

- Generating responses

- Recommender systems

- Industry insights

- Q & A

Deep Learning in industry

- ▶ Companies have endless amounts of data!
Or do they?
- ▶ Performance
Is .9 accuracy/ F_1 /etc. good enough?
No? Would 0.95 be?
- ▶ Business logic/constraints
 - *Your model is doing great in general, but not in case X, Y and Z.*
Can you keep it exactly as it is now, and fix just these cases?
- ▶ Explicit domain knowledge
E.g.: recommending product X for user Y is not applicable, as it is not available where user Y lives.

Deep Learning in industry

- ▶ Hybrid Code Networks
Combining RNNs with domain-specific knowledge
- ▶ Smart Reply
Automated response suggestion for email

Hybrid Code Networks

Task

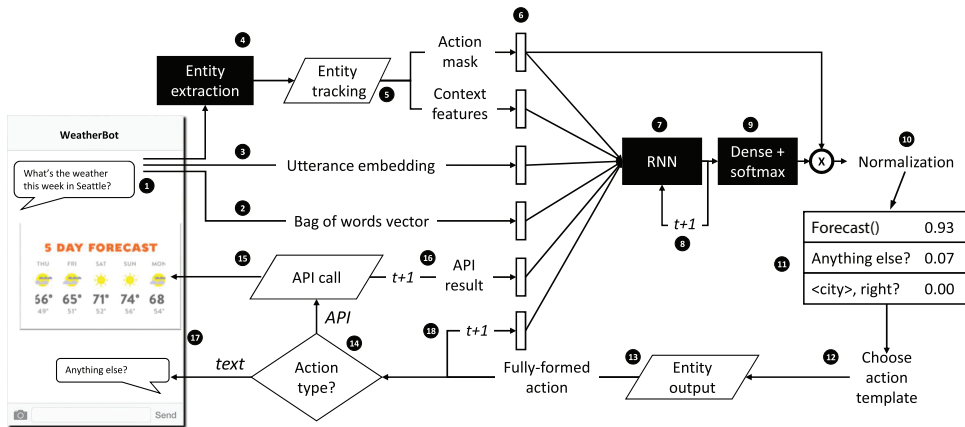
Dialogue system. User can converse with a system that can interact with APIs.

Combining RNNs with domain-specific knowledge

- ▶ Incorporate business logic by including modules in the system that can be programmed
- ▶ Explicitly condition actions on external knowledge

[Williams et al., 2017]

Hybrid Code Networks



Trapezoids refer to programmatic code provided by the software developer.
Shaded boxes are trainable components.

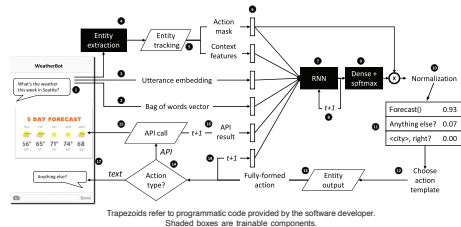
[Williams et al., 2017]

Hybrid Code Networks

Training of the RNN

Supervised setting

Every step: update weights, according to entropy loss on correct prediction of actions.



Reinforcement learning

At the end of the dialogue: update weights, according to:

$$\mathbf{w} \leftarrow \mathbf{w} + \left(\sum_t \nabla_{\mathbf{w}} \log \pi(a_t | \mathbf{h}_t; \mathbf{w}) \right) (G - b)$$

Annotations for the equation:

- LSTM**: Points to the hidden state \mathbf{h}_t in the policy function.
- return of the dialog: $G = 0.95^{T-1}$** : Points to the return G .
- Jacobian**: Points to the gradient term $\nabla_{\mathbf{w}} \log \pi(a_t | \mathbf{h}_t; \mathbf{w})$.
- estimate of the average return of the current policy, estimated on the last 100 dialogs**: Points to the baseline b .

Smart Reply

Automated response suggestion for email

Use an RNN to generate responses for any given input message.

Additional constraints

- ▶ **Response quality**

Ensure that the individual response options are always high quality in language and content.

- ▶ **Utility**

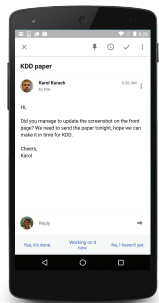
Select multiple options to show a user so as to maximize the likelihood that one is chosen.

- ▶ **Scalability**

Process millions of messages per day while remaining within the latency requirements.

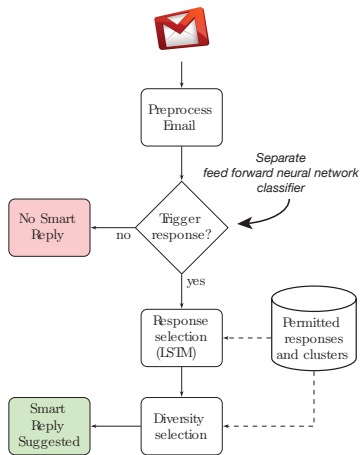
- ▶ **Privacy**

Develop this system without ever inspecting the data except aggregate statistics.



[Kannan et al., 2016]

Smart Reply



[Kannan et al., 2016]

Response selection

- ▶ Construct a set of allowed responses R .
- ▶ Organise the elements of R into a trie.
- ▶ Conduct a left-to-right beam search, and only retain hypotheses that appear in the trie.

Complexity: $O(\text{beam size} \times \text{response length})$.

Utility/diversity

Goal: present user with diverse responses

Instead of “No”, “No, thanks”, and “Thanks!”, we’d rather produce “No, thanks”, “Yes, please”, “Let me come back to it”.

- ▶ Manually label a couple of messages per response intent.
- ▶ Use a state-of-the-art label propagation algorithm to label all other messages in R .

What do we learn?

- ▶ Deep learning component is a (small) part of a much larger system.
- ▶ Getting the right training data can be hard.
- ▶ The machine learned part is guided/corrected/prevented from predicting undesired output.