INTELLIGENT CONVERSATIONAL BOT 2017

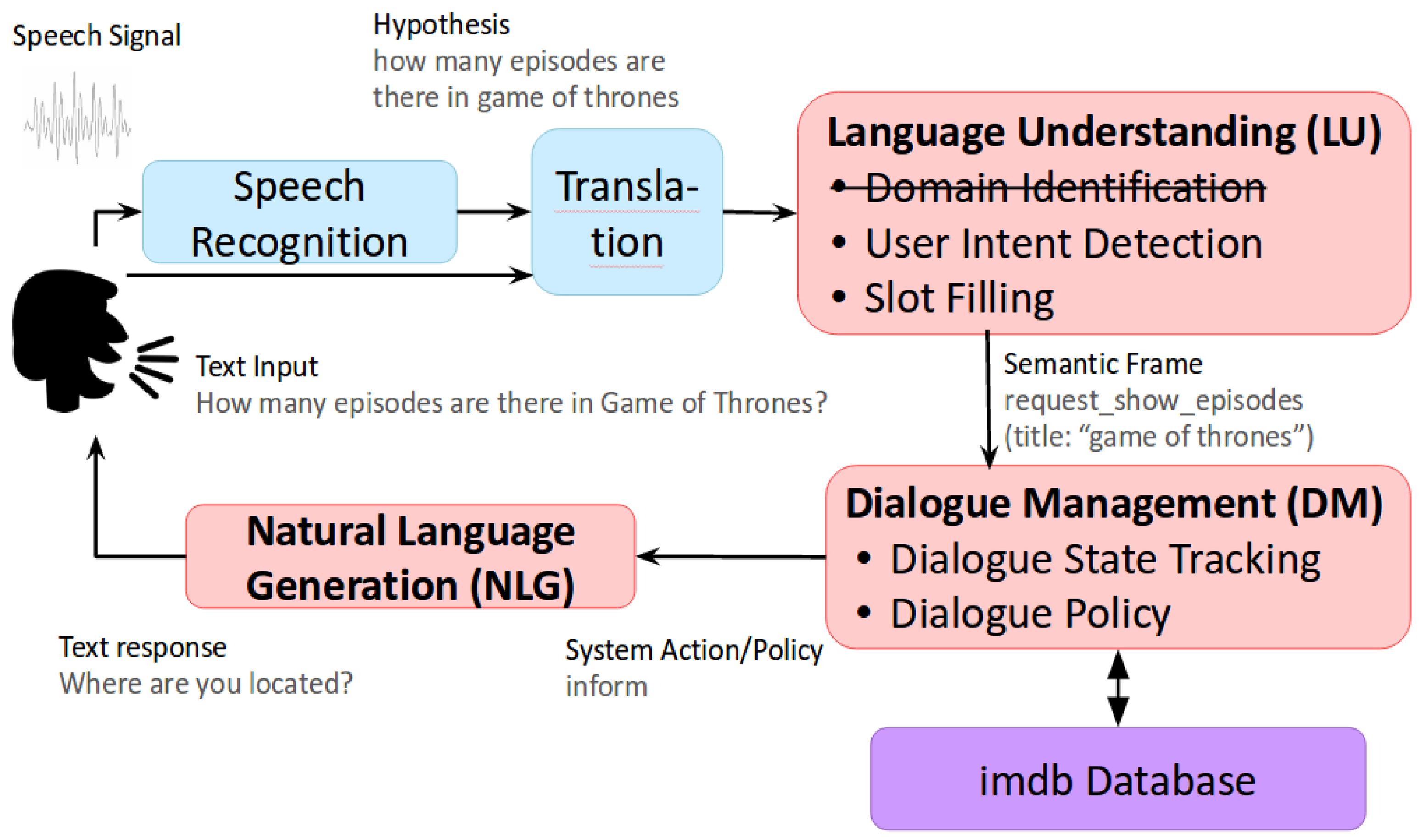
TVBot

C.-S. Chen, K.-L. Lo, F.-Y. Sun, H.-T. Yeh & Y.-T. Yeh

https://facebook.com/theTVBot

國立臺灣大學 National Taiwan University

Overview



We are able to:

- find show info given other info.
- find character name given person's real name, or the opposite.
- find the person related to a show given her/his job.
- accept English and Chinese.
- do voice recognition.

Ontology and Database

Source

Trakt.tv: it provides a unified interface to popular movie / tv show sources (IMdb, etc.).

Intent List

Fine-grained intents over vague intents. *Why?* Reduce possibility for wrong slot filling.

e.g. *Who directed House of cards?* v.s. *How many episodes are there in Game of Thrones?*

• request_show_title	• request_cast_character	• request_crew_lighting
• request_show_first_aired	• request_person_name	• request_crew_editing
• request_show_air_info	• request_person_birthday	• request_crew_art
• request_show_runtime	• request_pershon_birthplace	• request_crew_costume
• request_show_network	• request_person_death	• inform
• request_show_country	• request_crew_sound	• confirm_yes
• request_show_status	• request_crew_camera	• confirm_no
• request_show_rating	• request_crew_directing	• deny
• request_show_genres	• request_crew_writing	• thanks
• request_show_aired_episodes	• request_crew_visual_effects	
• request_show_season_count	• request_crew_production	

Value Mapping

The slot value from user NL might not match actual slot.

- country: Japan, Japanese → jp
- year: 1970, the 70s → 1970
- rating: nine, 9 → 9.0
- time: 16:00, 4p.m., 4 oclock → 16:00

Language Understanding

RNN-NLU

RNN-NLU is able to classify intent and fill slots at the same time, *jointly optimized*.

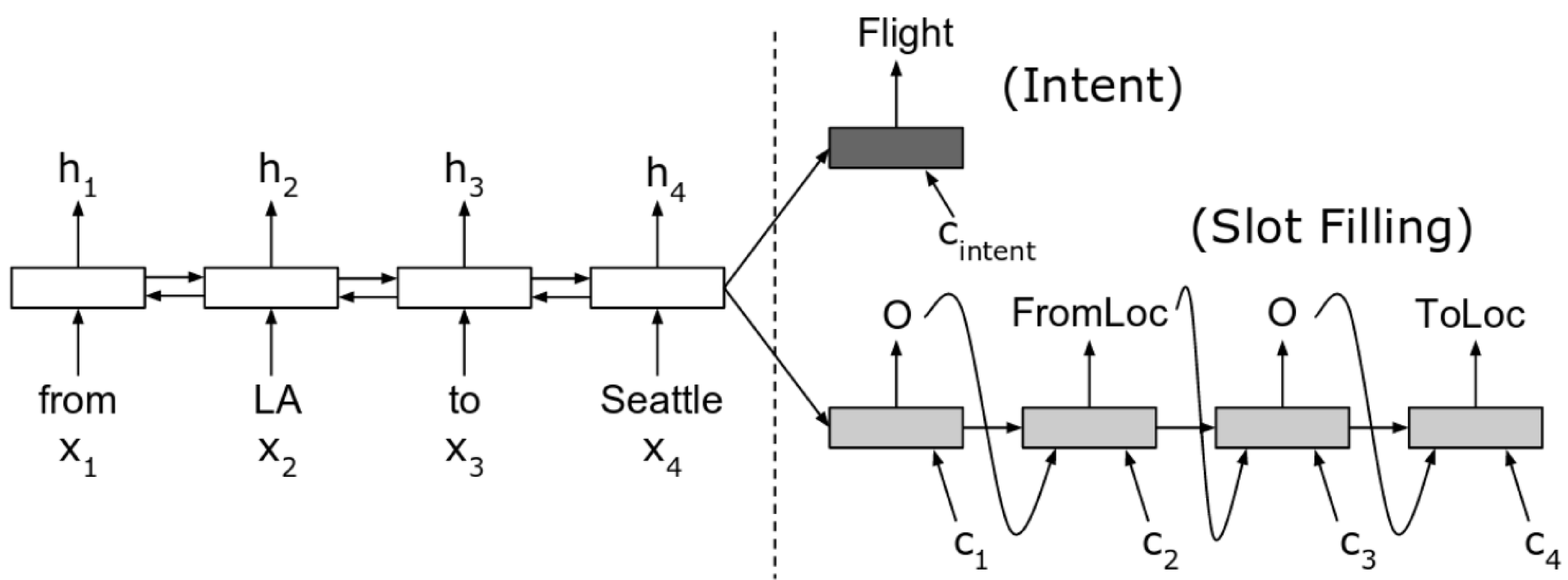


Figure 1: The framework of RNN-NLU [1]

Tend to *over-fill*. e.g. *I would like to watch a show that airs on Friday.* → `request_show_title(show.air_day: 'Friday', show.country: 'to')`

api.ai

Provided by an external service. Model unknown.
Tend to *under-fill*. e.g. *I would like to find a show produced by Jamie.* → `request_show_title()`

Comparison

	RNN-NLU	api.ai
Good	New model, powerful if fine tuned.	Online training.
Bad	Hard to train (days every time).	Black box, can't tune or apply tricks.

Table 1: NLU comparison

	RNN-NLU	api.ai
Accuracy	0.80	N/A
F1	0.89	N/A

Table 2: Performance for NLU

Data

Mostly generated by hands. We generate almost 1000 templates, and fill in the db values as the training sentences.

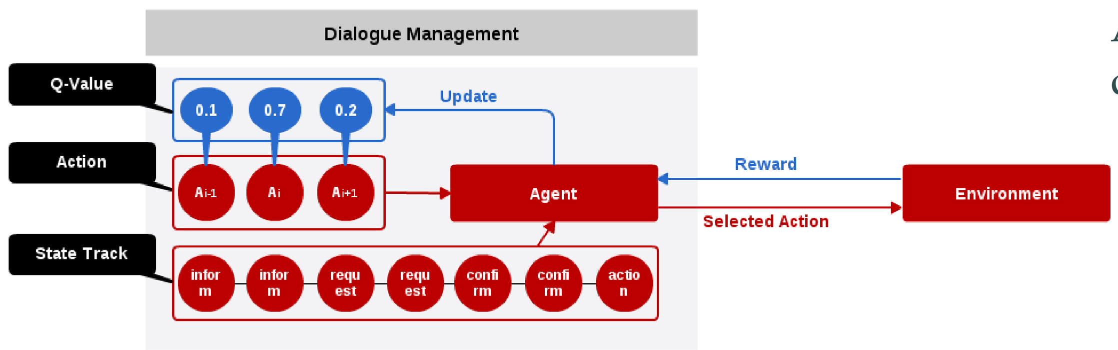
Dialogue Management

State Tracker

For both `user` and `agent`, the state tracker tracks the following information in vector form $\langle a_1, a_2, \ldots, a_n \rangle, a_i \in \{0, 1\}, n = |Slots|$:

- (latest_)inform_slots
- (latest_)inform_values
- (latest_)request_slots
- (latest_)confirm_slots
- (latest_)confirm_values
- latest_action

RL Agent (DQN[2])



Ruled-based Agent

Ask user by a pre-defined sequence.

User Simulator

- Goal: contains *target slot* and *information* known by user.
- Success: after episode ends, check if agent reply correct slot.
- Error model: to simulate NLU error, a parameter controlls the possibility of wrong NLU.

	Ruled-based	DQN
w/ error	0.66	0.53
w/o error	1.0	0.93

Table 3: Performance for agent

Reward function

Our reward function definition is

$$\begin{cases} -1 \times max\_turn, & \text{when dialog fails} \\ 2 \times max\_turn, & \text{when dialog success} \\ -1, & \text{otherwise} \end{cases}$$

Performance

Table 4: Succ. rate of warmup

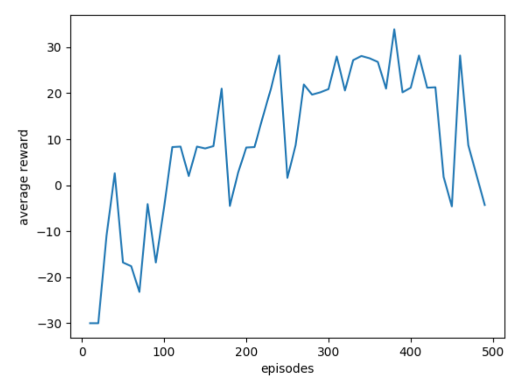


Table 5: Avg. reward against user-sim w/o error

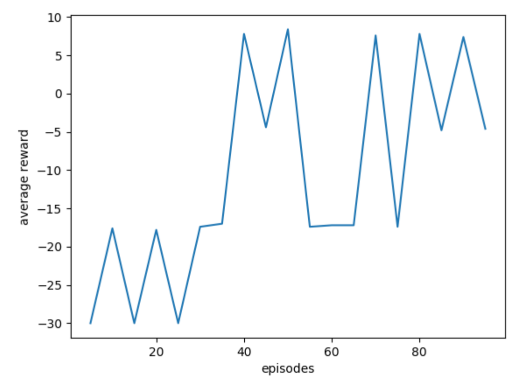


Table 6: Avg. reward against user-sim w/ error

Natural Language Generation

We separated the sentence into 7 categories:

- thanks
- inform
- request
- confirm_again
- confirm_conflict
- confirm_no_conflict
- multiple_choice

	BLEU
Training	1.0
Testing	0.4425

Table 7: Performance for NN-NLG

Rule-based NLG

For each categories, we designed a template to fill the info.

NN-NLG

The model is a seq2seq. Trained on several templates for each categories.

Miscellaneous

Speech API

Integrated with Bing Speech-to-text API. If *audio* gotten, call the api.

Translation API

The db is built in *English*. To support other language, sent to Bing Translation API if other language detected.

References

[1] Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016*, pages 685–689, 2016.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.