FoodBot: A Chatbot Knowing all about Restaurants

FoodBot Team 1
National Taiwan University
國立臺灣大學 National Taiwan University

Link ☞ http://140.112.49.151:8081/index3.html

Features:

✔ 2 **talking styles**: gentle🙂 & hilarious😆

✔ **Speech API** supported

✔ Functionalities including: **recommending restaurants** and **providing various info** of a certain restaurant (phone number, rating, reviews, has wifi or not, etc.

Scan me!

Ontology

Data Source: yelp (using the API and crawler)
Data Size: 1000 restaurants information from New York.
Number of tables: 3
Restaurant Table: (about 1000 rows)

Column title	name	rating	categories	Phone	city	displayAddress	postalCode	countryCode	latitude	longitude
Example	21 Green House	4	Italian	123456	NY	#134 Swer Rd.	13211	U.S.	13.3342	245.78

Comment Table: (about 300000 rows)

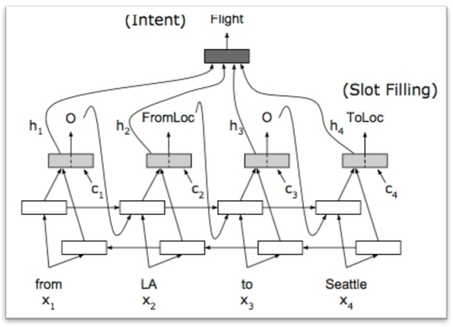
Column title	name	rating	user	area	review	reviewid
Example	21 Green House	3.5	24	Bronx	This one is not bad. The food is tasty, but it's a little bit noisy.	100

Other info of Restaurant Table:(about 1000 rows)

Column title	Name	Hours	price_range	Takes_Reservations	(…15 extra information items from the Yelp website)
Example	21 Green House	Monday to Friday, 9:00-17:00	$50	Yes	(all the information normally is Yes/No)

Language Understanding

Model architecture:



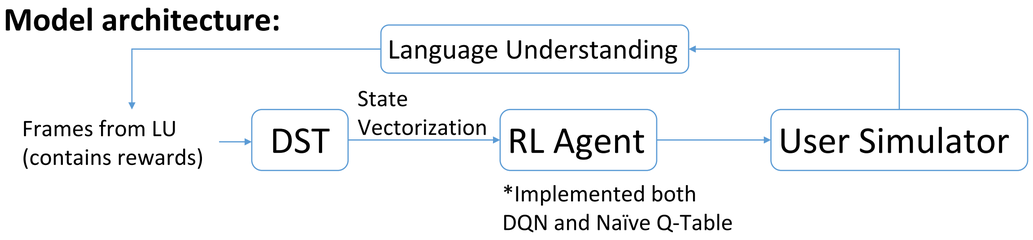
Descriptions of the *Attention-based RNN NLU model
- Each input word is embedded into a 128-dim word vector
- The two bottom layers are bidirectional RNN where each cell is an LSTM unit.
- Adopted Encoder-Decoder structure

Data collection: 30,000 sentences from 41 templates,6 user dialogue records
Training size: 10 intents - for each intent: 3000 sentences
Testing size: 40 different sentences
Performance on Testing Set: ~67.144% (2201/3278 turns)

* Liu, Bing, and Ian Lane. "Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling." *arXiv preprint arXiv:1609.01454* (2016).

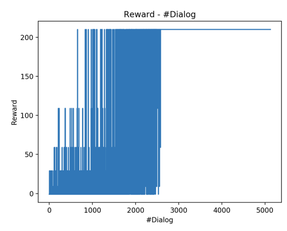
Dialogue Management

Model architecture:



User simulator:
(1) Generate sentences according to randomly chosen pre-defined semantic frames
(2) Policy checker: check the action agent made and response accordingly
(3) Memory mechanism: when asked to confirm details in history, it can check it with the memory and give the right answer

Learning curve: see right ==========>
Success rate:
(1) Learning based: 83.3% (Fully-exploited)
(2) Rule based: ~67.1% (Mainly caused by the LU error)



Natural Language Generation

Model architecture:
(1) An adapted version of a *semantically conditioned LSTM
(2) Template-based (using this one!)

Data collection:
(1) LSTM: sentences with ~10 intents written by human
(2) Template-based: sentences with 22 intents written by human

Training size: 400 sentences and semantic frames from ~10 intents
Testing size: 20 semantic frames
Performance:
(1) BLEU score: 0.5518
(2) Naturalness: see the example below

inform(name = 33 greenwich; address = 7593 kirkland lane rockaway)

=> *"33 greenwich is good good in 7593 kirkland lane rockaway address is in."*

* Wen, T. H., Gasic, M., Mrksic, N., Su, P. H., Vandyke, D., & Young, S. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.