

The User Simulator for Reinforcement Learning Powered Chatbot



Florian L. Kreyssig, Iñigo Casanueva

@Wenyi Tao

Speech Signal

Hypothesis

are there any action movies to see this weekend



Speech Recognition

Language Understanding (LU)

- Domain Identification
- User Intent Detection
- Slot Filling

Text Input

Are there any action movies to see this weekend?

Semantic Frame

request_movie

genre=action, date=this weekend

Dialogue Management (DM)

- Dialogue State Tracking (DST)
- Dialogue Policy

Natural Language Generation (NLG)

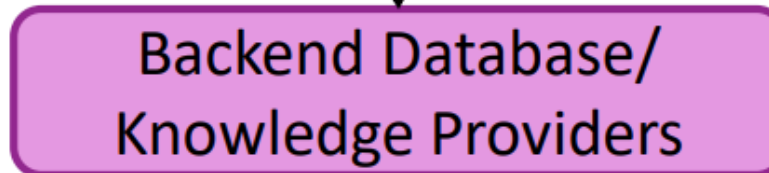
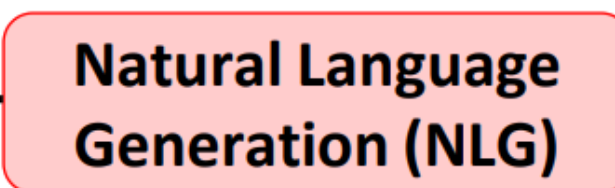
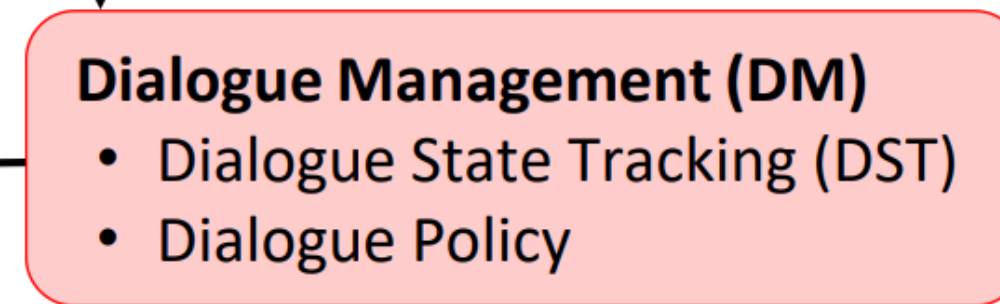
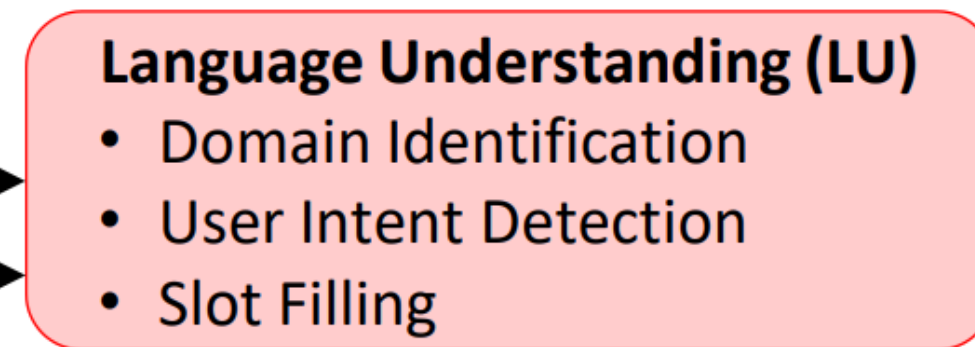
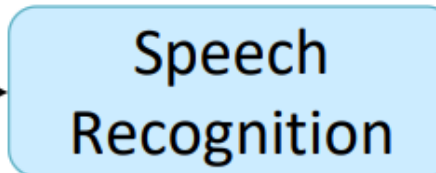
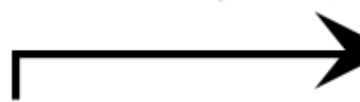
Text response

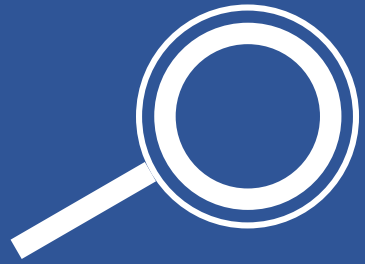
Where are you located?

System Action/Policy

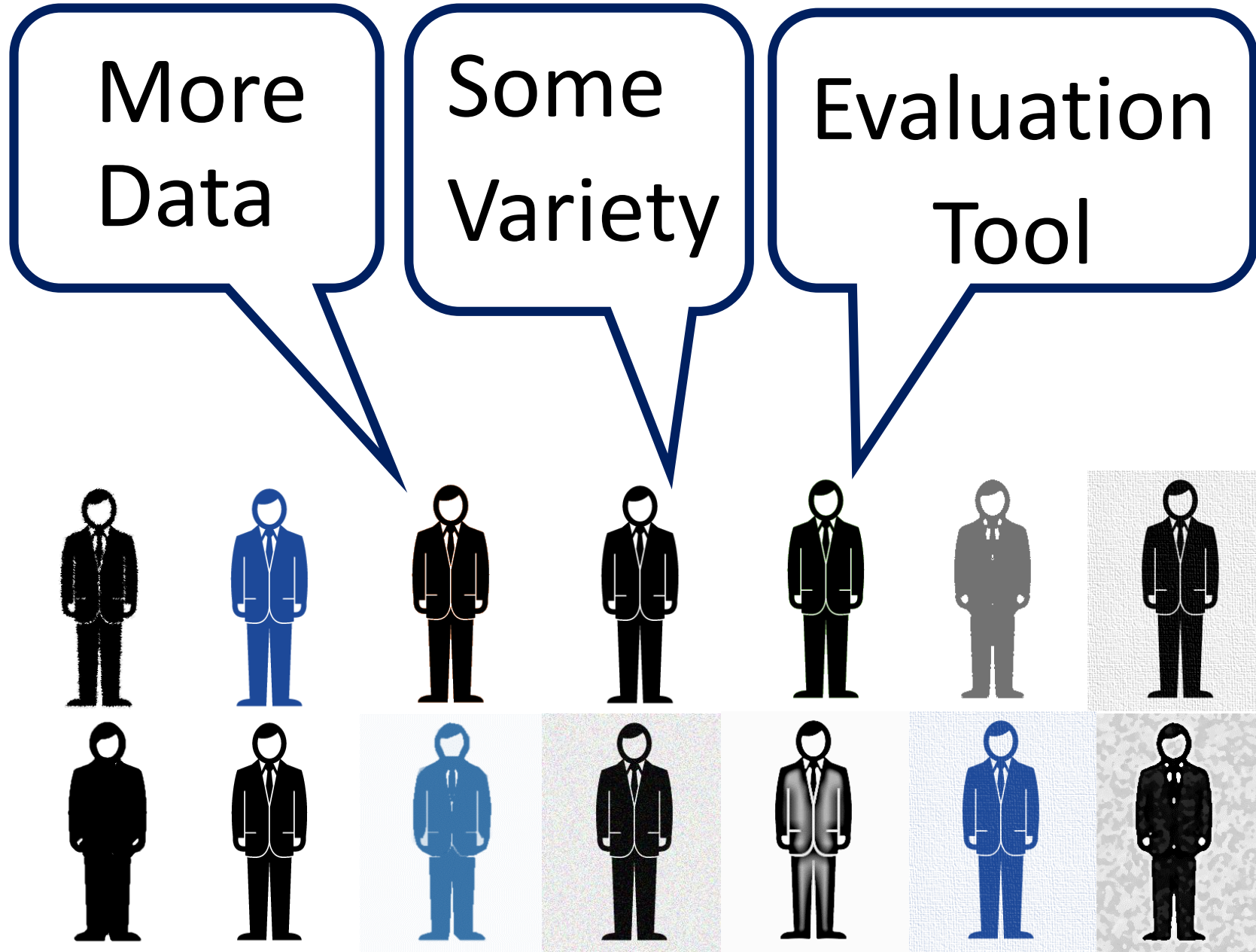
request_location

Backend Database/
Knowledge Providers



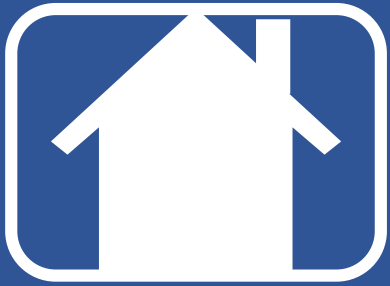


Why User Simulator





and even
more



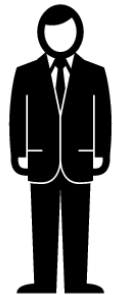
User Simulator

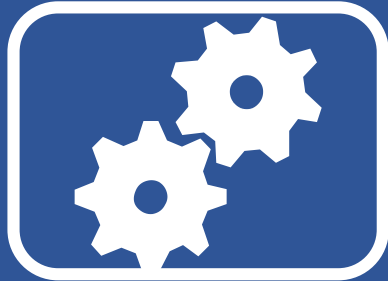
Keeps a list of its goals and actions

Randomly generates an agenda

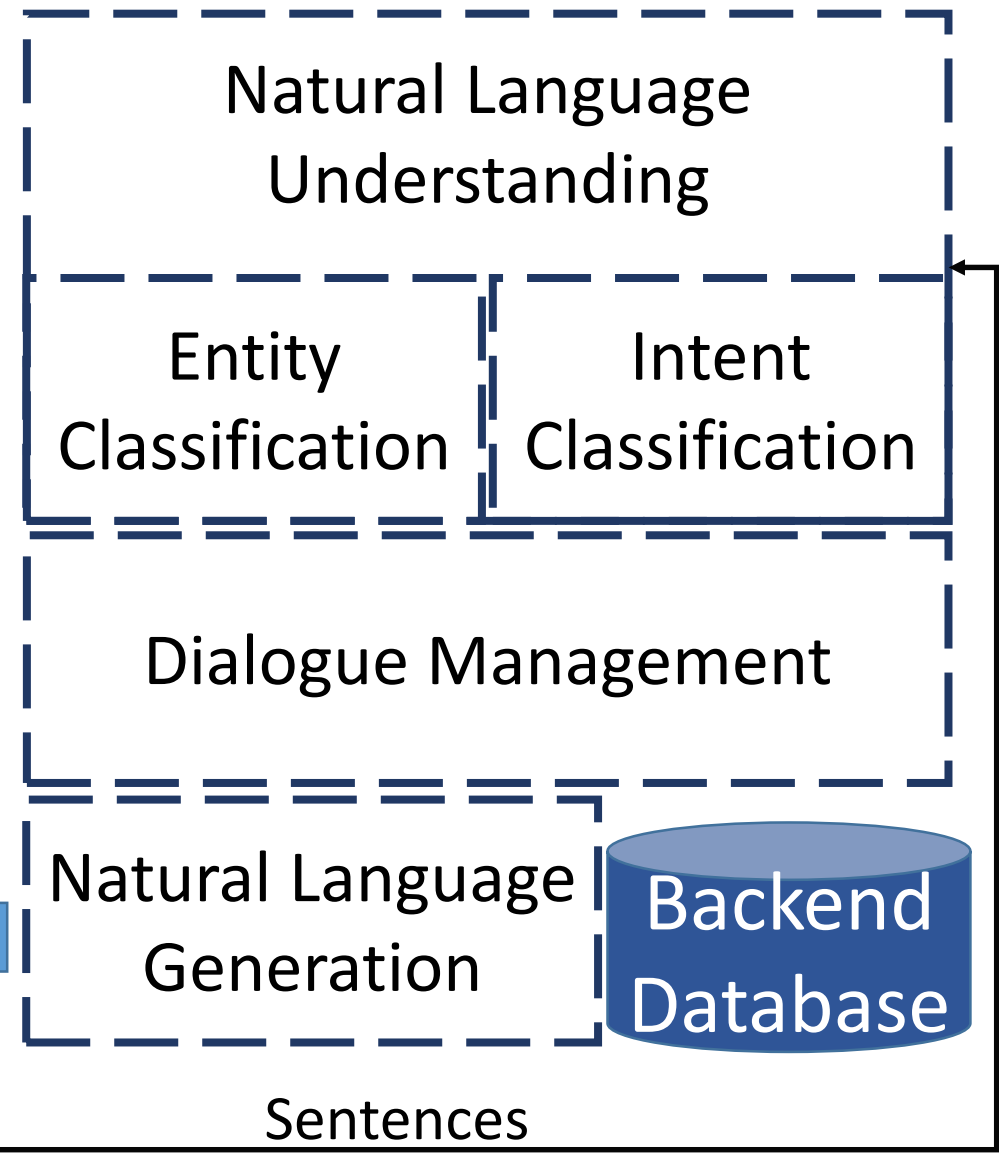
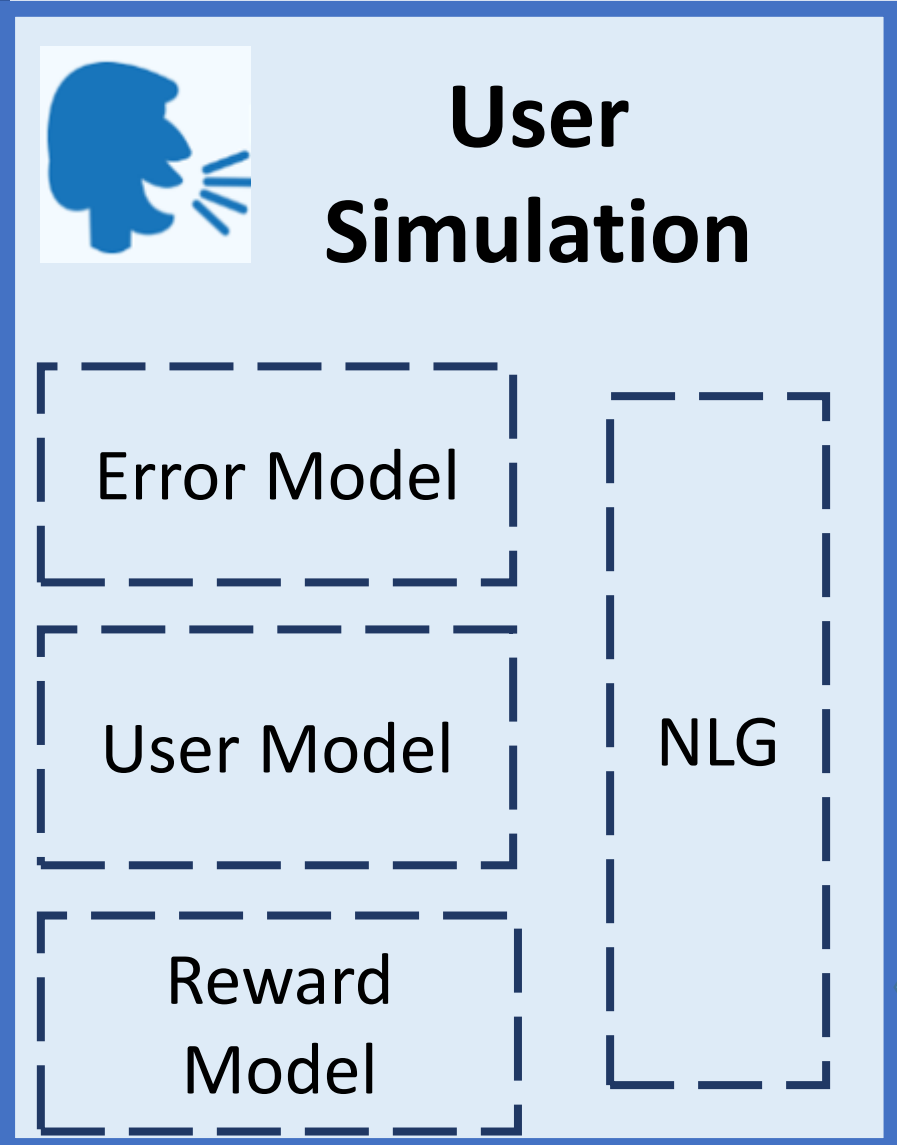
Update its list of goals and adds new ones

- Generate natural and reasonable conversations to enable reinforcement learning for exploring the policy space
- Type:
 - Graph-based
 - Bi-gram model $P(a_u | a_m)$
 - Agenda-based
 - Neural based





Elements of User Simulator



The error model enables the system to maintain the robustness during training

Agenda-Based

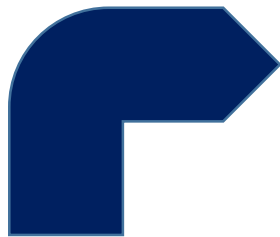
Goal generator

Constraints:

Type = bar
drinks = beer
area = central

Request:

name = ?
addr = ?
phone = ?

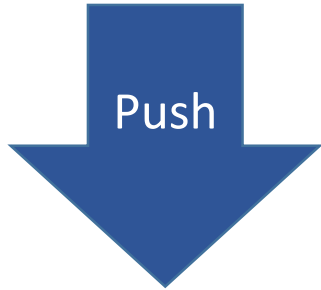


I am looking for a nice bar serving beer.

```
inform(Type = bar)
Inform(drinks=beer)
Inform (area =
central)
Request(name)
Request(addr)
Request(phone)
Bye()
```

Agenda
Stack-like

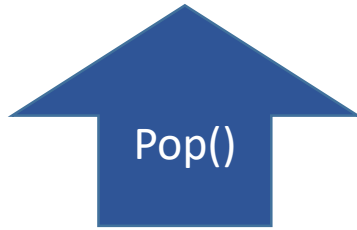
Sample from corpus



```
Negate(drinks=beer)
Inform(price= cheap)
Inform (area = central)
Request(name)
Request(addr)
Request(phone)
Bye()
```

Okay, a wine bar
What price range?





Inform(price= cheap)
Inform (area = central)
Request(name)
Request(addr)
Request(phone)
Bye()





Push

```
Affirm()  
Inform(price= cheap)  
Inform (area = central)  
Request(name)  
Request(addr)  
Request(phone)  
Bye()
```

A bar serving beer
Correct?



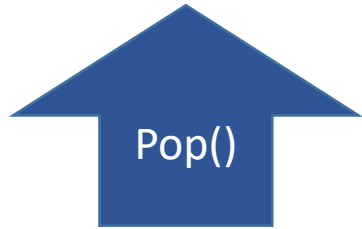


yes, Something
cheep in town
center

Pop()

Request(name)
Request(addr)
Request(phone)
Bye()





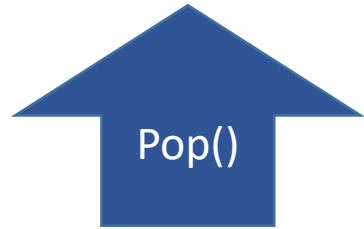
Request(phone)
Bye()

I got Murphy on
Main Square serves
Cheap beer



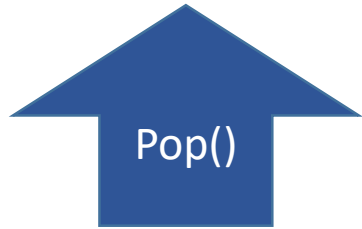


What is the phone
Number?



Request(phone)
Bye()





Bye()

You can call at
__**





Thanks , Bye

Pop()

Bye()





Agenda-Based User Simulator

- Pros:
 - Very elegant statistical bootstrapping system
 - Very goal-consistence
 - Capture randomness
 - No training data needed
- Cons:
 - Limited diversity
 - In-ablitiy to interface a text level belief tracker

Neural-Based

State of the art

Goal
Generator

Feature
Extractor

Seq2Seq Model

(C,R)

Constraint

Request

Food = Spanish
Area = north

loc = ?
Phone = ?

Initial:

1. Sampled with a probability from ontology
2. If no constraint is sample, goal is resample

Update:

3. Update turn specific constraint

Goal
Generator

Feature
Extractor

Seq2Seq Model

$$v_t = [a_t \ r_t \ i_t \ c_t]$$

Dialogue act

a1

Machine-act
vector encodes
the dialogue acts.
all-binary

Slot
mentioned

a2

Requst_not
fullfiled

Rt

Binary
representation of
requests not
fulfilled

Inconsistency
slot

I

Inconsistency
between system's
response and
Constraint

Constraint

C

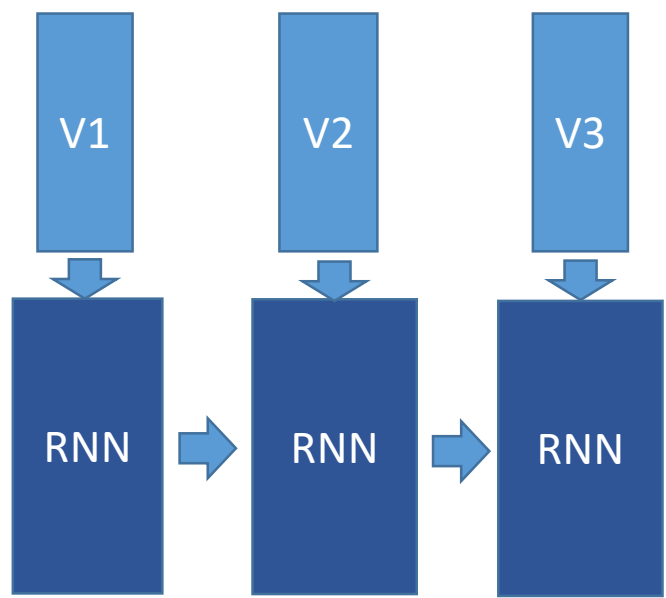
Current
constraint
informed slots

Goal
Generator

Feature
Extractor

Seq2Seq Model

$$(h_t, s_t) = \text{RNN}(X_t, s_{t-1})$$

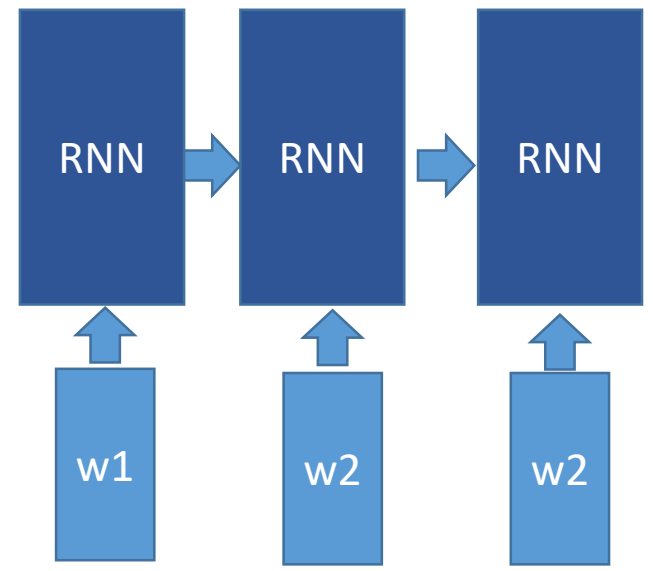


Encoder

$$P_t = W_p H_T^E + b_p$$

Last output

Decoder



One-hot embedding

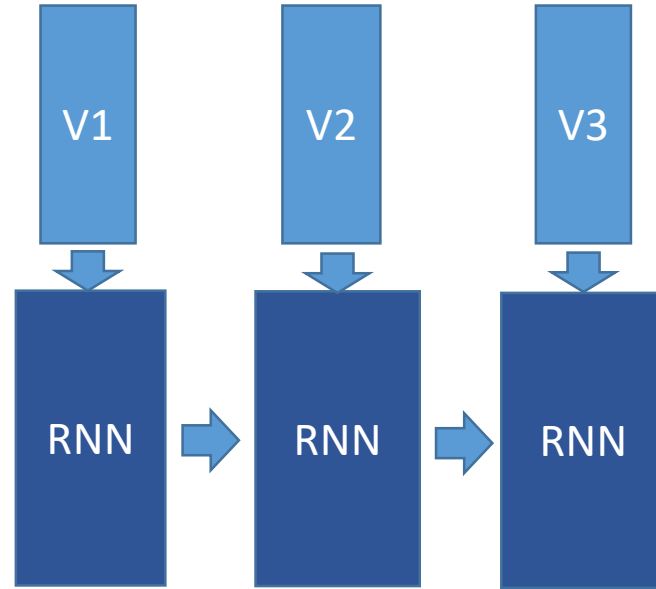
Objective function

$$P(u|p) = P(w_0|P) \prod_{t=1}^L P(w_t|w_{t-1} | w_{t-1} \dots w_0, p)$$

Goal
Generator

Feature
Extractor

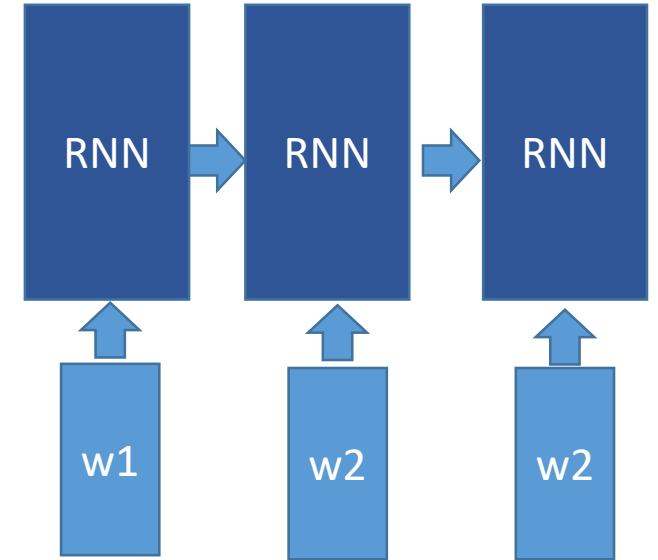
Seq2Seq Model



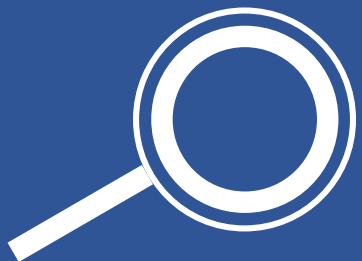
Encoder

Training: the true words from the dataset are used. (Teachers Forcing)

Decoder



Testing: beam search or sample from probability distribution



Evaluation metric

Train. Sim.	Eval. Sim.			
	NUS		ABUS	
	Rew.	Suc.	Rew.	Suc.
NUS-best	13.0	98.0 ^{N₁}	13.3	99.8
ABUS-best	1.53	71.5 ^{A₁}	13.8	99.9 ^{A₂}
NUS-avg	12.4	96.6	11.2	94.0
ABUS-avg	-7.6	45.5	13.5	99.5

- **Statistical methods:** similarity between the outputs of the US and a real user on a test set.(blue or other)
- **Cross-model Evaluation:** based on success rate

success rate	NUS-eval	ABUS-eval
NUS-train	98	99.8
ABUS-train	71.5	99.5

Conclusion

User simulator are necessary for providing

Ambient Training Data

For deep reinforcement learning training



Necessity cumbersome

Thanks

@Wenyi Tao
2019.02.02
