Deep Reinforcement Learning for Dialogue Generation with Hierarchical Recurrent Encoder Decoder

Heejin Jeong, Xiao Ling

Hierarchical Recurrent Encoder-Decoder

A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware *Query Suggestion*

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob G. Simonsen, Jian-Yun Nie

Building *End-To-End Dialogue Systems* Using Generative Hierarchical Neural Network Models

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, Joelle Pineau

Motivation (Query Suggestion)

- Context aware query suggestion
- Long term, sequential dependence to narrow down current query (query N-gram).
- Explore the possibility of suggesting “long tailed queries” never seen in corpus

Motivation (End-To-End Dialogue Systems)

- Open domain, generative conversational dialogue system
- "Realistic, flexible interactions" in non-goal driven setting
- *Train "user simulator" for POMDP models*
- *Use features directly to represent state in POMDP models*

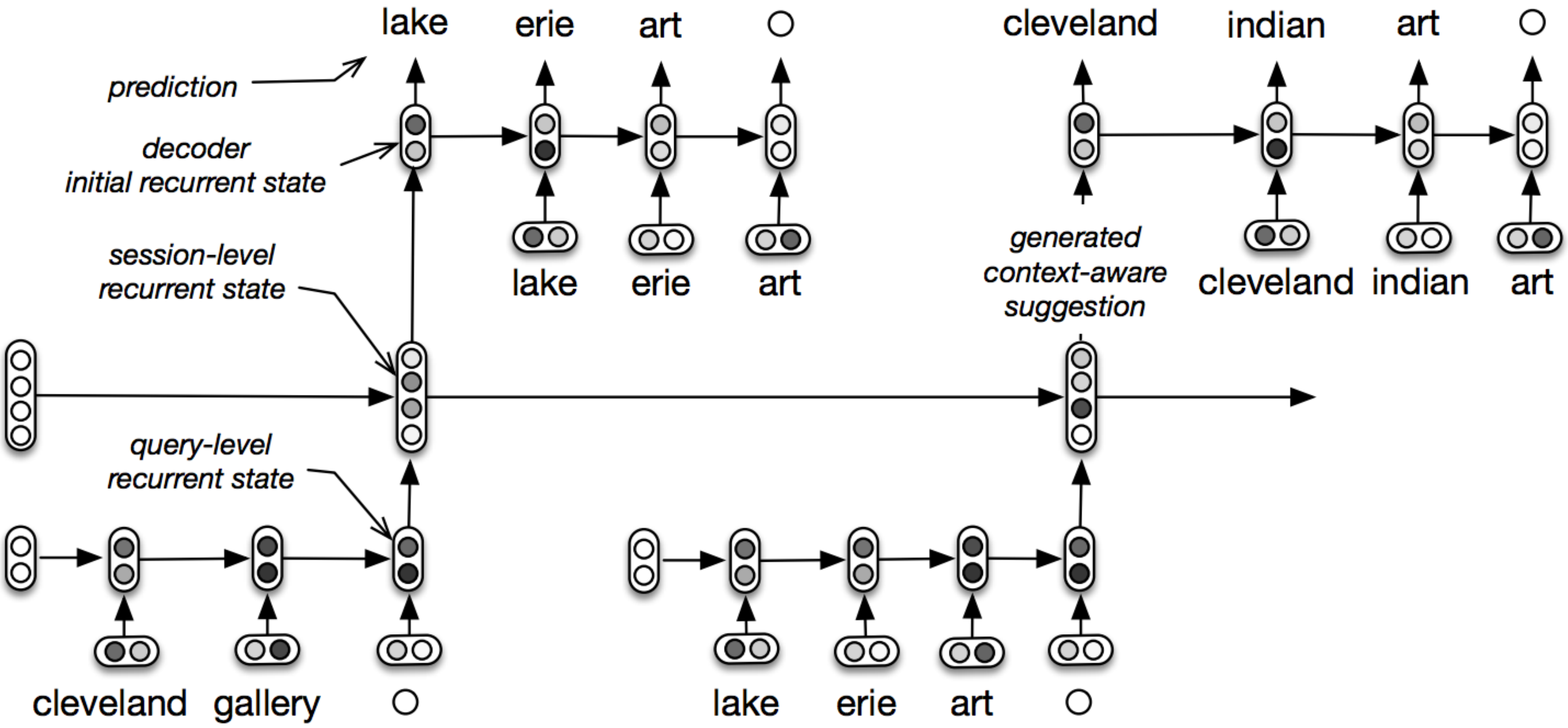
Network Assumption*

$$\Pr[w \text{ emitted at time } t \mid c_t] \propto \exp(c_t \cdot v_w)$$

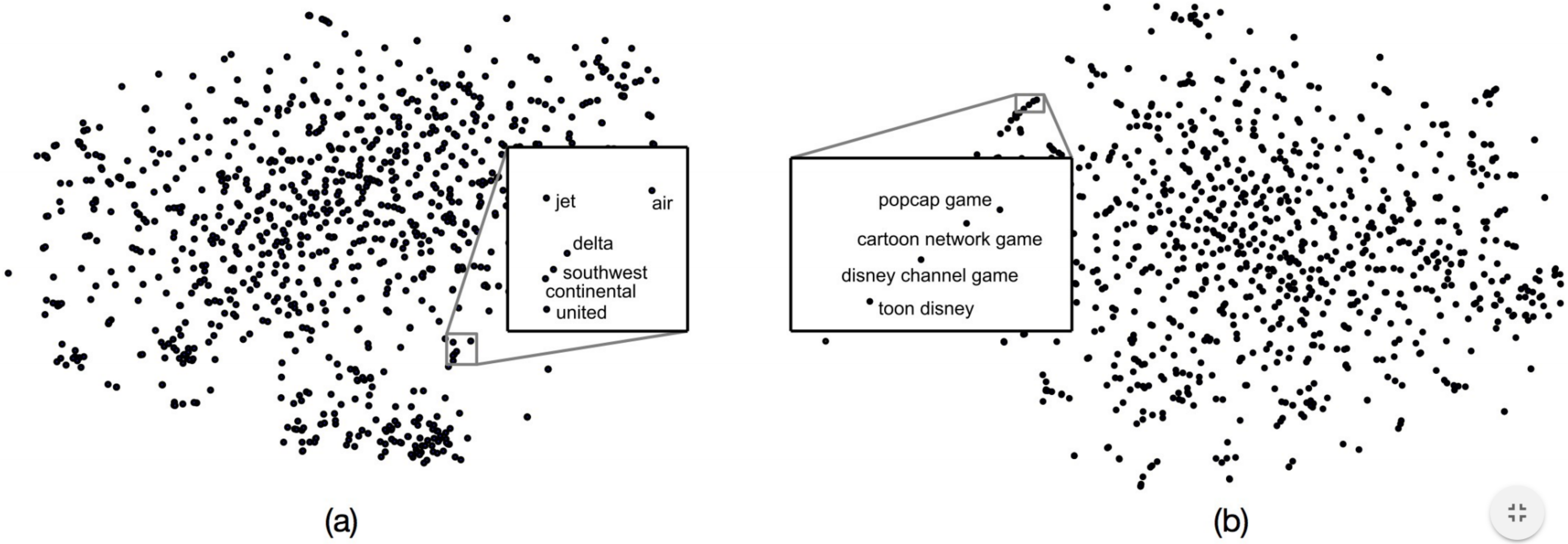
* RAND-WALK: A latent variable model approach to word embeddings

Sanjeev Arora, Yuanzhi Li ,Yingyu Liang, Tengyu Ma, Andrej Risteski

Network Architecture



Embedding maps topic similarity to spatial similarity



Objective Maximize Session log-likelihood

$$\mathcal{L}(S) = \sum_{m=1}^{M} \log P(Q_m | Q_{1:m-1})$$

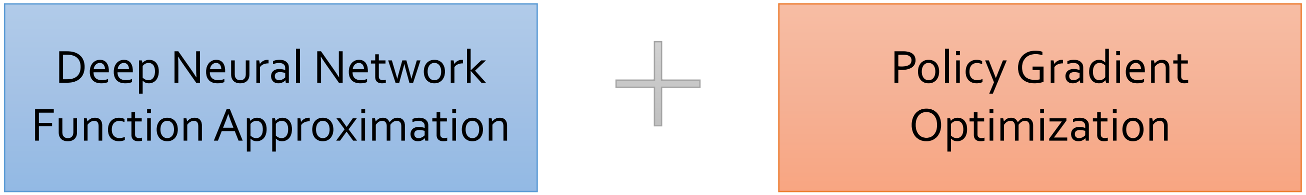
$$= \sum_{m=1}^{M} \sum_{n=1}^{N_m} \log P(w_{m,n} | w_{m,1:n-1}, Q_{1:m-1})$$

Deep Reinforcement Learning (DRL) for Dialogue Generation

Heejin Jeong, Xiao Ling

DRL-SEQ2SEQ

- Neural Reinforcement Learning generation method



Deep Neural Network Function Approximation

Policy Gradient Optimization

- The model's backbone - SEQ2SEQ (encoder-decoder architecture)
- The model's learning – simulating conversation between two virtual agents to explore its action space in order to maximize **its expected cumulative total future reward**.

Policy Gradient Methods

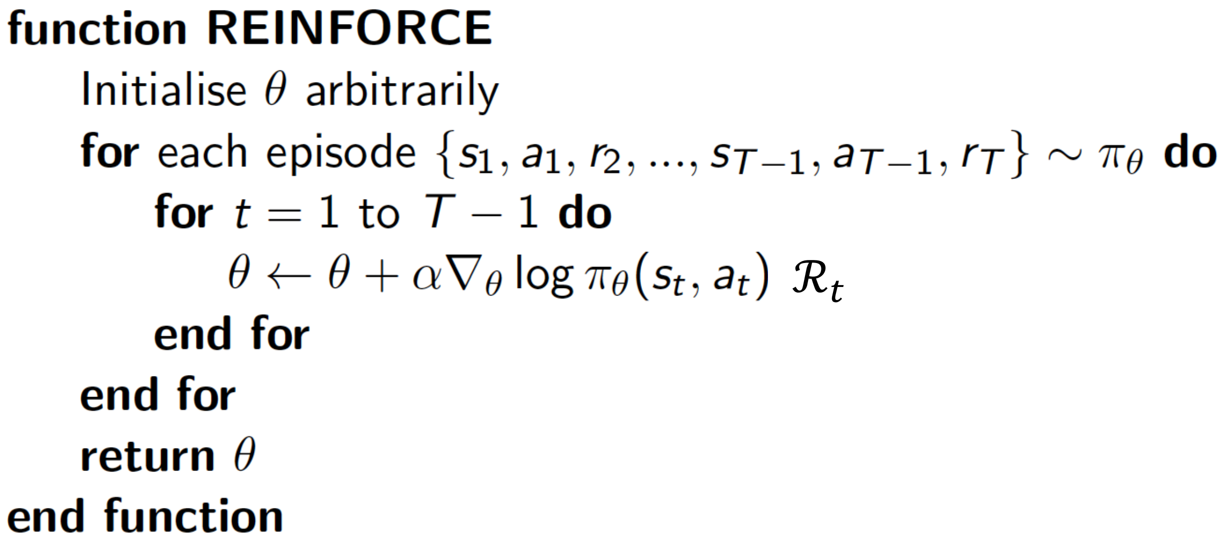
- Policy Objective Function, $J(\theta)$. SGA

$$\theta \longleftarrow \theta + \alpha \nabla_\theta J(\theta)$$

- Gradient w.r.t θ

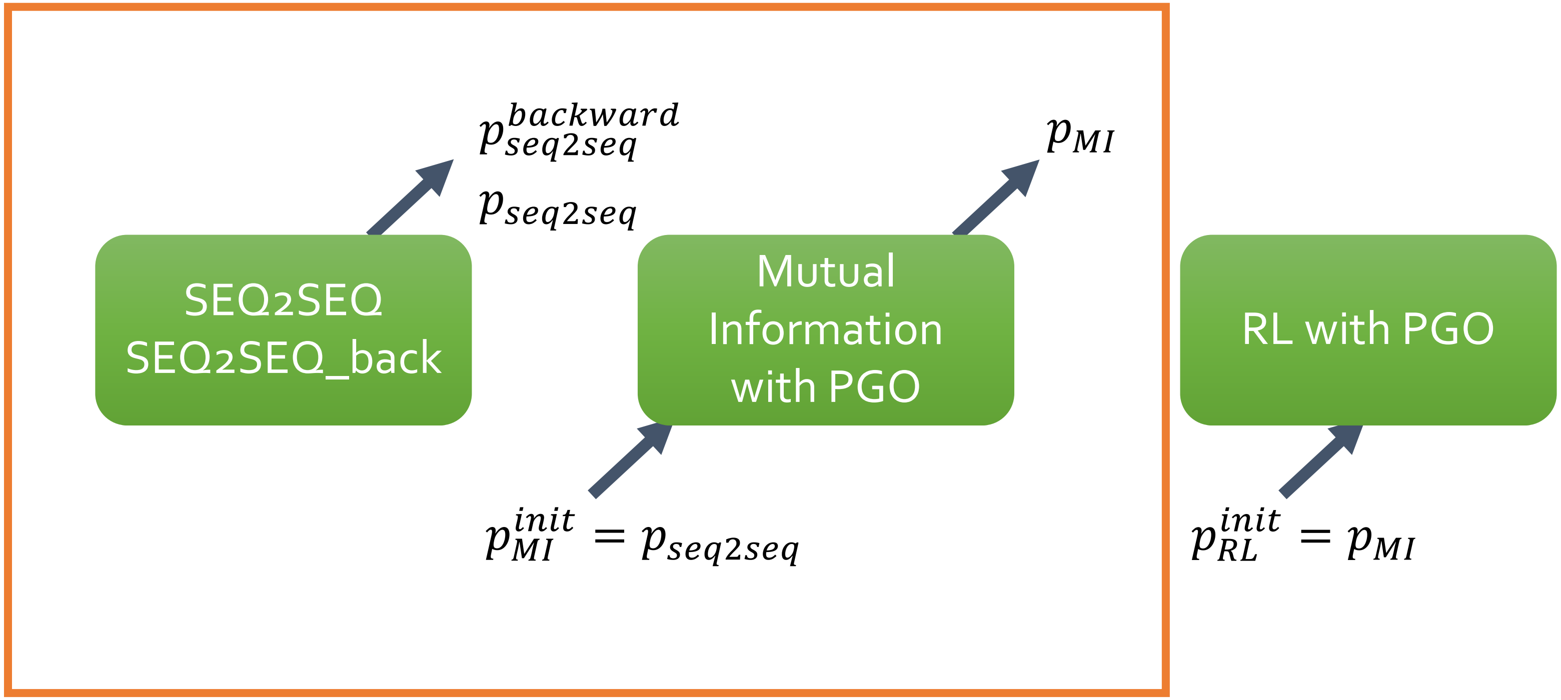
$$\nabla_\theta \mathrm{E}[\mathcal{R}_t | \pi_\theta] = \mathrm{E}[\mathcal{R}_t \nabla_\theta \log \pi_\theta(s_t, a_t)\,]$$

- REINFORCE (Monte-Carlo Policy Gradient)



```
function REINFORCE
    Initialise $\theta$ arbitrarily
    for each episode $\{s_1, a_1, r_2, ..., s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ do
        for $t = 1$ to $T - 1$ do
            $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t)\ \mathcal{R}_t$
        end for
    end for
    return $\theta$
end function
```

DRL-SEQ2SEQ



$p_{seq2seq}^{backward}$
$p_{seq2seq}$
p_{MI}
SEQ2SEQ
SEQ2SEQ_back
Mutual
Information
with PGO
RL with PGO
$p_{MI}^{init} = p_{seq2seq}$
$p_{RL}^{init} = p_{MI}$

RL for Open-Domain Dialogue

- Learning system – Two agents conversation
  $\xcancel{[p_1, q_1, p_2, q_2, \cdots, p_T, q_T]} \longrightarrow u_1, u_2, \cdots, u_{2T}$

- Action, $a \in A$: a dialogue utterance to generate, and $|A| = \infty$ $\longrightarrow$ $|A| = (bucket\ size) \times (vocab\ size)$

- State, s $\in S$: s_t = $[u_{t-1}, u_t]$

- Policy, $\pi(a|s) = p_{RL}\ (u_{t+1}|u_{t-1}, u_t)$ with the form of the SEQ2SEQ

- Reward, $r_t = r(a_t, s_t) = \lambda_1 r_{1,t} + \lambda_2 r_{2,t} + \lambda_3 r_{3,t}$: a weighted sum of three reward functions

- Mutual Information

RL Dialogue Simulation

- Initial Dialogue from a training set , m

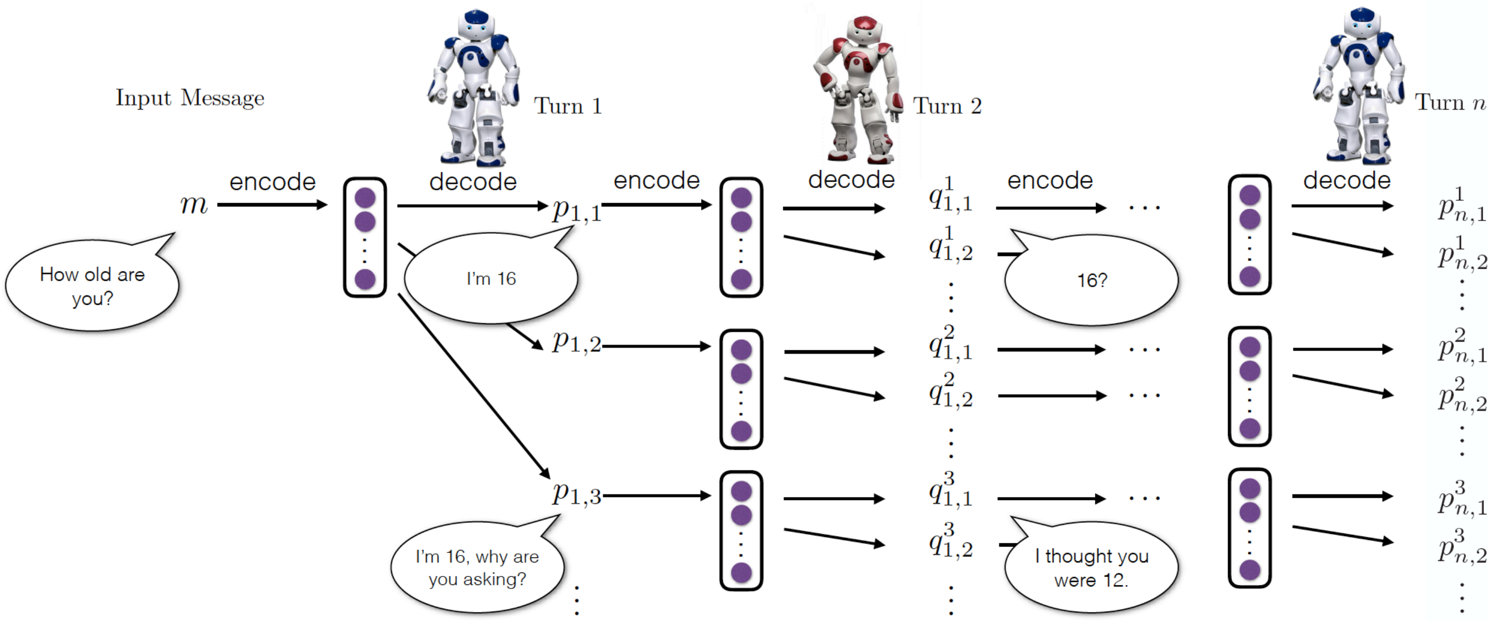


Figure 1: Dialogue simulation between the two agents.

Supervised Learning

- Pre-train a model using SEQ2SEQ with attention, $p_{seq2seq}$

  - Encoder input: $[u_{t-1}, u_t]$ (the concatenation of two previous turns)
  - Target: u_{t+1} (each turn)

- Pre-train a model using standard SEQ2SEQ, $p_{seq2seq}^{backward}$

- Initialize the RL policy p_{RL} for Mutual Information Learning with the pre-trained model $p_{seq2seq}$

Mutual Information

Semantic Coherence

: Mutual Information between a_t and $s_t = [u_{t-1}, u_t]$

$$m = \frac{1}{N_a} \log p_{seq2seq}(a_t | u_{t-1}, u_t) + \frac{1}{N_{\bar{s}_t}} \log p_{seq2seq}^{backward}(u_t | a_t)$$

$$m = \frac{1}{N_a} \sum_i \log p_{seq2seq}(w_{t,i} | s_t, w_{t,1}, \cdots, w_{t,i-1})$$

$$+ \frac{1}{N_{\bar{s}_t}} \sum_i \log p_{seq2seq}^{backward}(v_{t,i} | a_t, v_{t,1}, \cdots, v_{t,i-1})$$

Mutual Information and PGO

- Objective Function:

$$J(\theta) = \mathrm{E}[m(\hat{a},[u_{t-1},u_t])] \qquad\qquad \hat{a} \sim p_{RL}$$

$$\nabla J(\theta) = m(\hat{a},[u_{t-1},u_t])\nabla \log p_{RL}(\,\hat{a}|[u_{t-1},u_t])$$

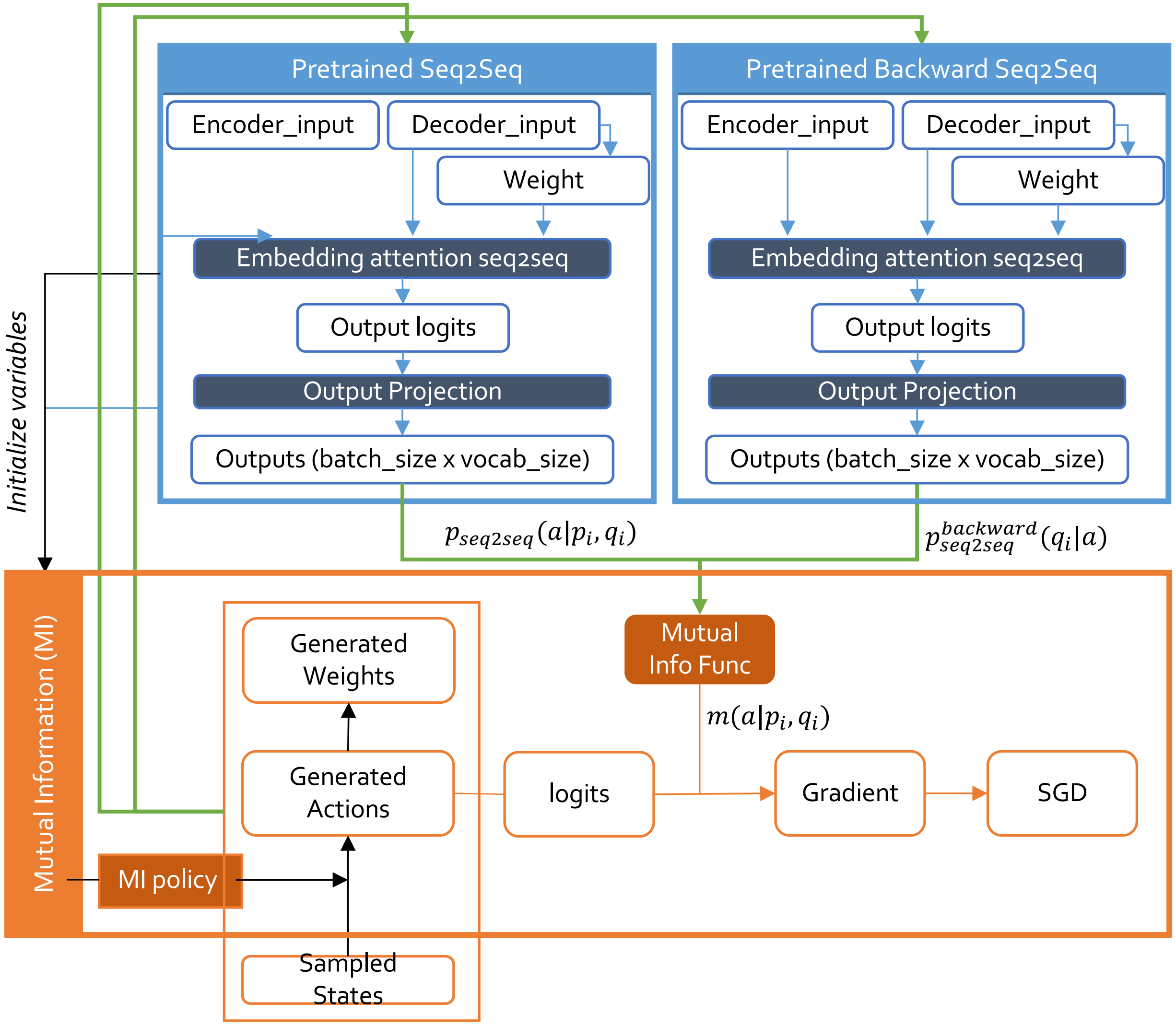
- Batch Setting:

$$\nabla J(\theta) = \sum_j m\big(\hat{a}_j,[u_{t-1},u_t]_j\big)\nabla \log p_{RL}(\,\hat{a}_j|[u_{t-1},u_t]_j)$$

$$= \nabla \sum_j m\big(\hat{a}_j,[u_{t-1},u_t]_j\big) \log p_{RL}(\,\hat{a}_j|[u_{t-1},u_t]_j)$$

- ~~Curriculum Learning Strategy, Baseline Strategy~~

Implementation



Datasets

Datasets

- CALLHOME American English Speech (LDC97S42)
- Open Subtitles Corpus (http://www.opensubtitles.org/)

CALLHOME Dataset

- 120 unscripted 30-minute telephone conversations between native speakers of English.
- "All calls originated in North America; 90 of the 120 calls were placed to various locations outisde of North America, while the remaining 30 calls were made within North America. Most participants called family members or close friends."*
- Transcripts of the conversations were obtained through ADR

*https://catalog.ldc.upenn.edu/ldc97s42

Open Subtitles Corpus

- Previously used for machine translation (30 languages)
- total number of files: 20,400
- total number of tokens: 149.44M
- total number of sentence fragments: 22.27M
- We only used 28 documents that translated english to another language

Examples (CALLHOME) - Raw

69.95 74.68 B: and i'm, i'm thinking about the @SEPTA, the transit workers.

74.77 75.29 A: **mhm**.

74.91 76.37 B: they have a very strong union.

76.38 78.00 B: **((now))** i work for the federal government

78.30 78.97 A: okay.

78.50 80.72 **B**: and **%um**, we can't

80.74 82.39 **B**: you know, we can not strike.

82.81 84.04 **B**: we're represented

84.18 86.93 **B**: **%um**, whether we belong to the union or not.

87.36 87.91 A: **mhm.**

87.43 89.74 B: **%um**, the union isn't

89.73 91.25 B: it is very powerful.

91.25 93.47 B: nationwide, it is very powerful

93.73 94.40 A: **mhm**.

Examples (Open Subtitles) - Raw

In the last century before the birth... of the new faith called Christianity... which was destined to overthrow the pagan tyranny of Rome... and bring about a new society... the Roman republic stood at the very centre of the civilized world.

"Of all things fairest." sang the poet...

"first among cities and home of the gods is golden Rome."

Yet even at the zenith of her pride and power... the Republic lay fatally stricken with a disease called... human slavery.

The age of the dictator was at hand... waiting in the shadows for the event to bring it forth.

In that same century... in the conquered Greek province of Thrace... an illiterate slave woman added to her master' s wealth... by giving birth to a son whom she named Spartacus.

A proud. rebellious son... who was sold to living death in the mines of Libya... before his thirteenth birthday.

There. under whip and chain and sun... he lived out his youth and his young manhood... dreaming the death of slavery... 2. 000 years before it finally would die.

Back to work!

Get up, Spartacus, you Thracian dog!

Come on, get up!

My ankle, my ankle!

My ankle!

Spartacus again?

This time he dies.

Back to work, all of you!

- Welcome, Lentulus Batiatus.

- Welcome, indeed, my dear captain.

Preprocessing (CALLHOME)

- Extract all words delimited by special symbols
- **((now)) , {{ok}}, [um]**
- Fold consecutive speaker turns

Preprocessing (Open Subtitles)

- Removed symbols
- **<i>, </ i>, \xc2\xa4**

”<i>They say your whole life flashes before</ i> <i>your eyes when you die. </ i>”

- Added consecutive speaker turns
- Divide into sessions, each one with four turns

Normalization

- Python back-port of CMU's Twoknizes library
- Twitter and web aware tokenizer for English
- Lower case
- White space stripping
- Numbers were not converted to ####
- Proper nouns were not folded

Example (CALLHOME) - Post Processing

B: do you think they accomplish anything </s>

A: i think there comes a lot for as far as the employee is concerned because um there's a lot of jobs around today in today's society that um they're not uh what you would call equal opportunity and a lot of times you don't have a standing chance against management unless you have some type of um backing with you um in certain instances like um where you have um discrimination to employees as far as raises are concerned or as far as um employment opportunities and getting better positions in certain establishments </s>

B: well i live in philadelphia and i'm i'm thinking about the septa the transit workers </s>

A: **mhm** </s>

B: they have a very strong union now i work for the federal government </s>

A: okay </s>

B: and um we can't you know we can not strike we're represented um whether we belong to the union or not </s>

A: **mhm** </s>

B: **um** the union isn't it is very powerful nationwide it is very powerful </s>

A: **mhm** </s>

Example (Open Subtitles) - Post Processing

A: in the last century before the birth of the new faith called christianity which was destined to overthrow the pagan tyranny of rome and bring about a new society the roman republic stood at the very centre of the civilized world
B: of all things fairest sang the poet
A: first among cities and home of the gods is golden rome
B: yet even at the zenith of her pride and power the republic lay fatally stricken with a disease called human slavery

A: the age of the dictator was at hand waiting in the shadows for the event to bring it forth
B: in that same century in the conquered greek province of thrace an illiterate slave woman added to her master s wealth by giving birth to a son whom she named spartacus
A: a proud rebellious son who was sold to living death in the mines of libya before his thirteenth birthday
B: there under whip and chain and sun he lived out his youth and his young manhood dreaming the death of slavery 2 000 years before it finally would die

A: back to work
B: get up spartacus you thracian dog
A: come on get up
B: my ankle my ankle
A: my ankle

HRED Training Details

- Divided into 104,057 sessions of four turns each
- Average sentence length is 7 words
- Maximum sentence length is 500 words
- 80% Train, 20% Validation set
- Minimum length is one word
- 10,000,000 max iterations
- 50,005 vocabularies

Results

Result

- SEQ2SEQ with Two Previous Sentences

  3 layers of 512 units , after 109000 Global Steps, Perplexity at 30.28

human: this is bad
bot: this is a joke
human: what joke
bot: i do not know
human: what are you doing
bot: nothing

human: you are a loser
bot: you are a real joke you know
human: what is a real joke
bot: i do not know

human: how old are you
bot: eleven
human: you are so young boy
bot: i am sorry
human: it's totally okay
bot: i am sorry
human: what are you up to this weekend
bot: i am sorry

human: where are you from
bot: where are you going
human: i am going home
bot: i am going to get you out of here
human: thank you but where
bot: what is this
human: nothing
bot: i am sorry

Result

- SEQ2SEQ Backward

3 layers of 512 units , after 107000 Global Steps, Perplexity at 28.29

GOOD EXAMPLES

human: i do not know man
bot: what do you think
human: that s amazing
bot: wow
human: that works best for me
bot: i will be right back
human: I love you too
bot: i love you
human: i know I am hot
bot: i am a little nervous
human: I am done with you
bot: i am a fucking bitch

BAD EXAMPLES

human: i am going to go to the bathroom
bot: i am going to go to the bathroom
human: this is weird
bot: i am going to be a great man
Bad human: okay
bot: i am going to go to the bathroom
human: you look gorgeous
bot: i am going to go see the kitchen
human: fine
bot: i am fine

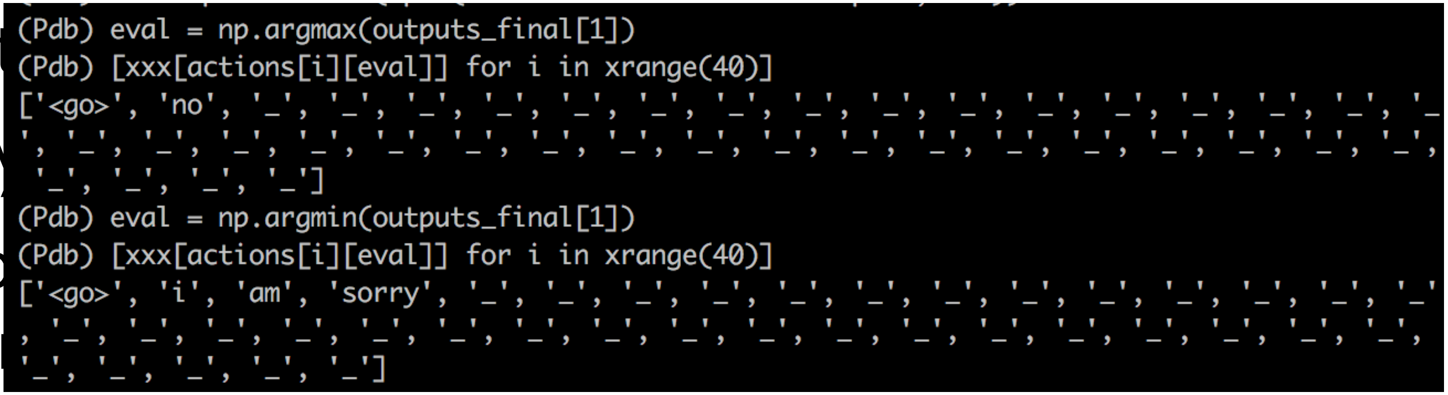
Result

- Mut

  3 la

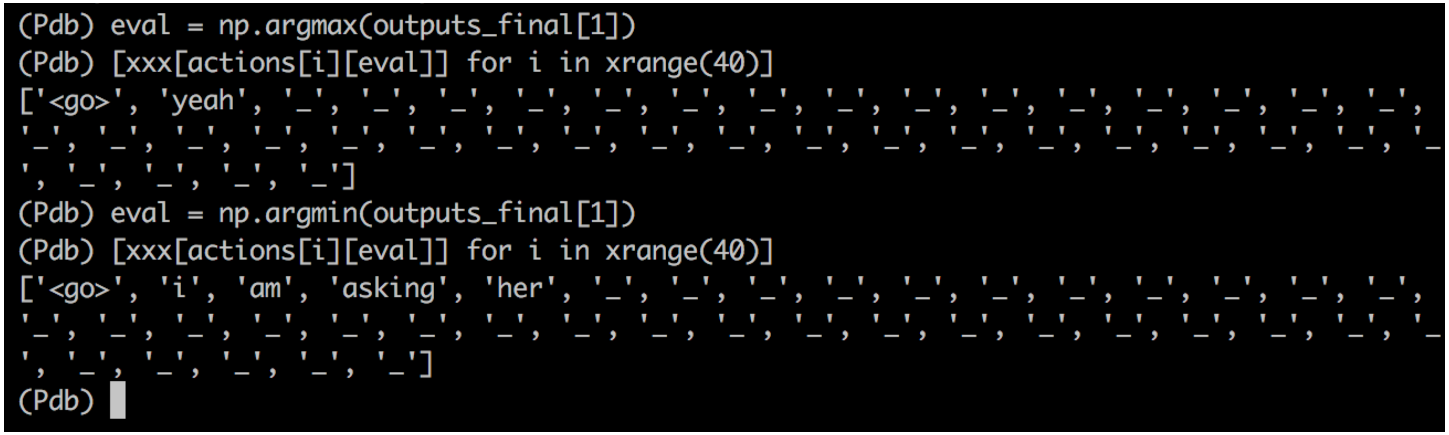
- Sho

  sco



```
(Pdb) eval = np.argmax(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'no', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
 '_', '_', '_', '_']
(Pdb) eval = np.argmin(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'am', 'sorry', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_'
, '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
'_', '_', '_', '_', '_']
```

10 steps later

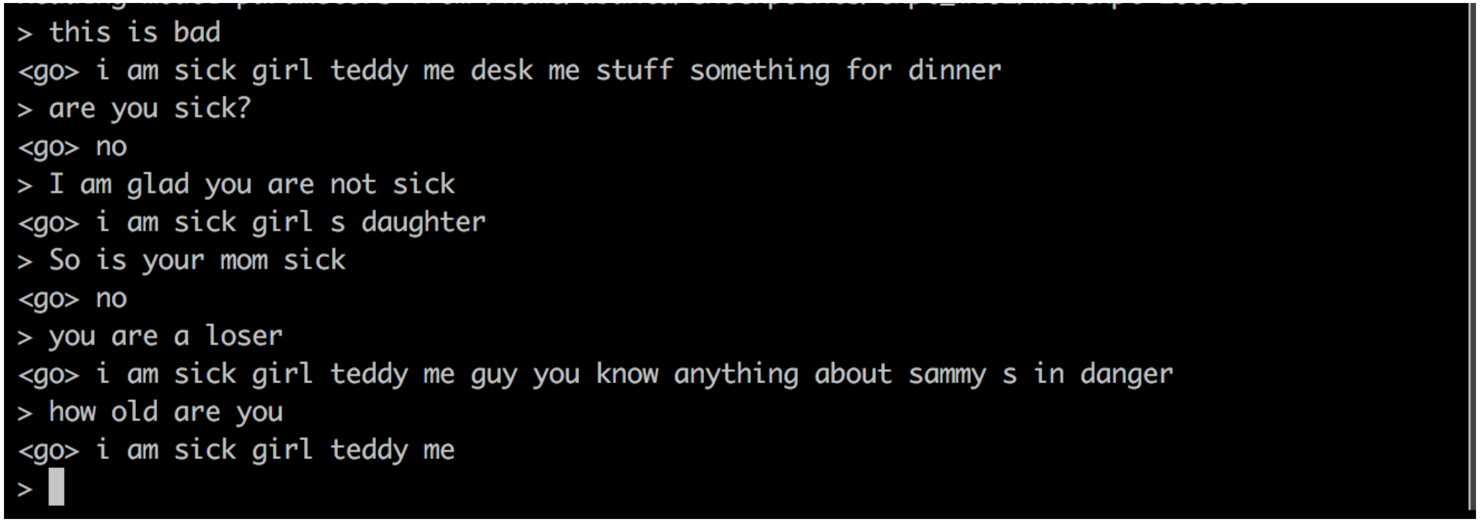


```
(Pdb) eval = np.argmax(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'yeah', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
'_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_']
(Pdb) eval = np.argmin(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'am', 'asking', 'her', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
'_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_', '_']
(Pdb) 
```

Result

- Mutual Information

3 layers of 512 units , after 20 steps

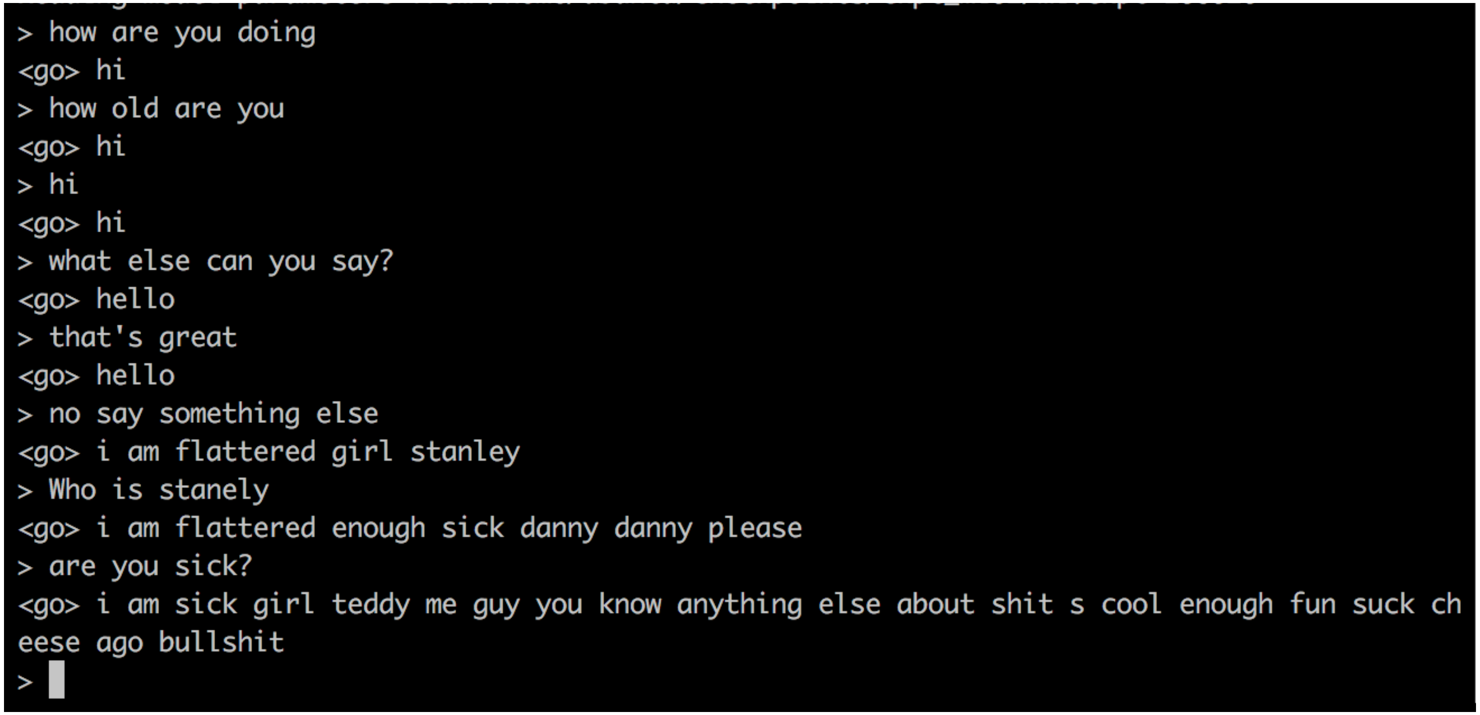


```
> this is bad
<go> i am sick girl teddy me desk me stuff something for dinner
> are you sick?
<go> no
> I am glad you are not sick
<go> i am sick girl s daughter
> So is your mom sick
<go> no
> you are a loser
<go> i am sick girl teddy me guy you know anything about sammy s in danger
> how old are you
<go> i am sick girl teddy me
> 
```

Result

- Mutual Information

  3 layers of 512 units , after 20 steps



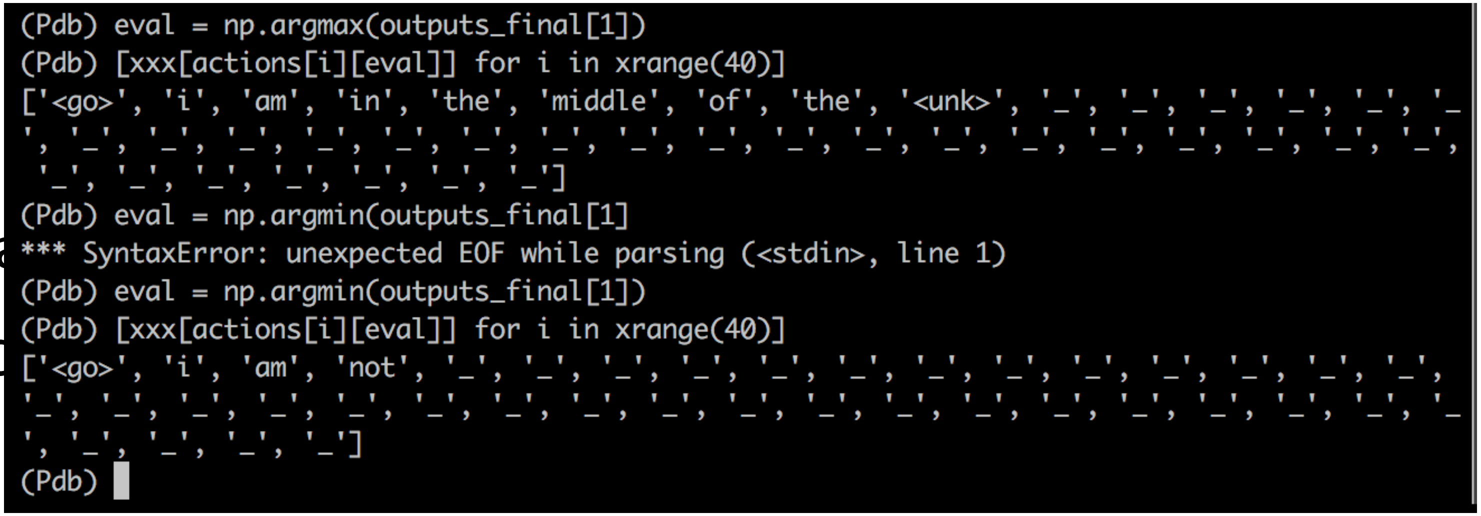
```
> how are you doing
<go> hi
> how old are you
<go> hi
> hi
<go> hi
> what else can you say?
<go> hello
> that's great
<go> hello
> no say something else
<go> i am flattered girl stanley
> Who is stanely
<go> i am flattered enough sick danny danny please
> are you sick?
<go> i am sick girl teddy me guy you know anything else about shit s cool enough fun suck ch
eese ago bullshit
> 
```

Result

- M

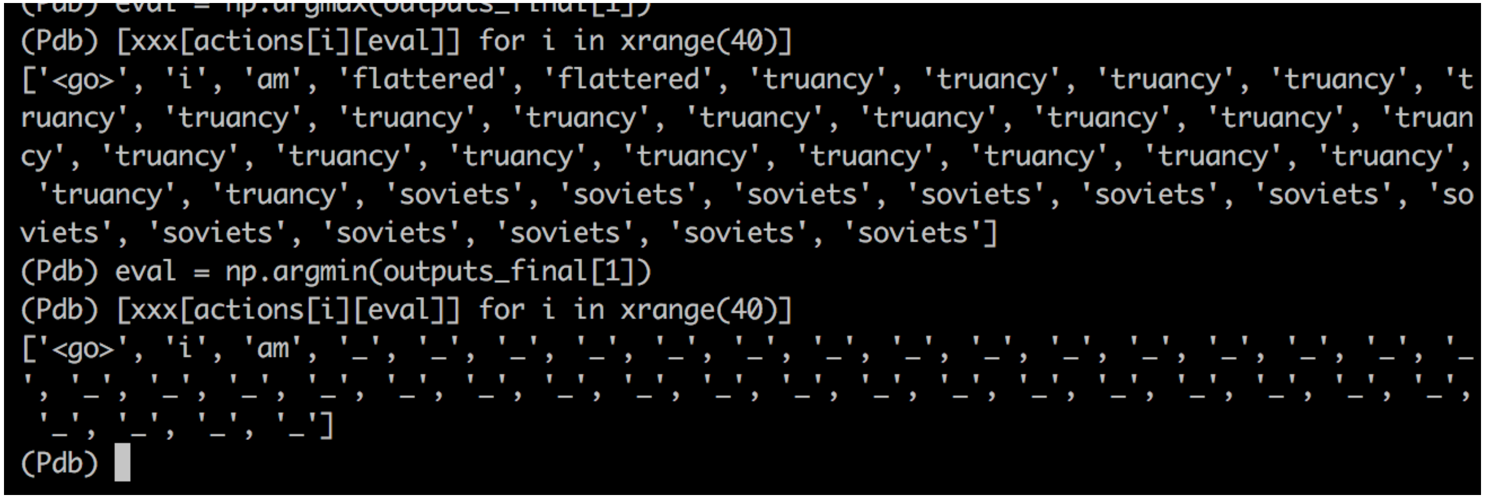
3 la

Ac



```
(Pdb) eval = np.argmax(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'am', 'in', 'the', 'middle', 'of', 'the', '<unk>', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
 '_', '_', '_', '_', '_', '_', '_']
(Pdb) eval = np.argmin(outputs_final[1]
*** SyntaxError: unexpected EOF while parsing (<stdin>, line 1)
(Pdb) eval = np.argmin(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'am', 'not', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
'_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_']
(Pdb) █
```

10 steps later



```
(Pdb) eval = np.argmax(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'am', 'flattered', 'flattered', 'truancy', 'truancy', 'truancy', 'truancy', 't
ruancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truan
cy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy', 'truancy',
 'truancy', 'truancy', 'soviets', 'soviets', 'soviets', 'soviets', 'soviets', 'soviets', 'so
viets', 'soviets', 'soviets', 'soviets', 'soviets', 'soviets']
(Pdb) eval = np.argmin(outputs_final[1])
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'am', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
 '_', '_', '_', '_']
(Pdb) █
```

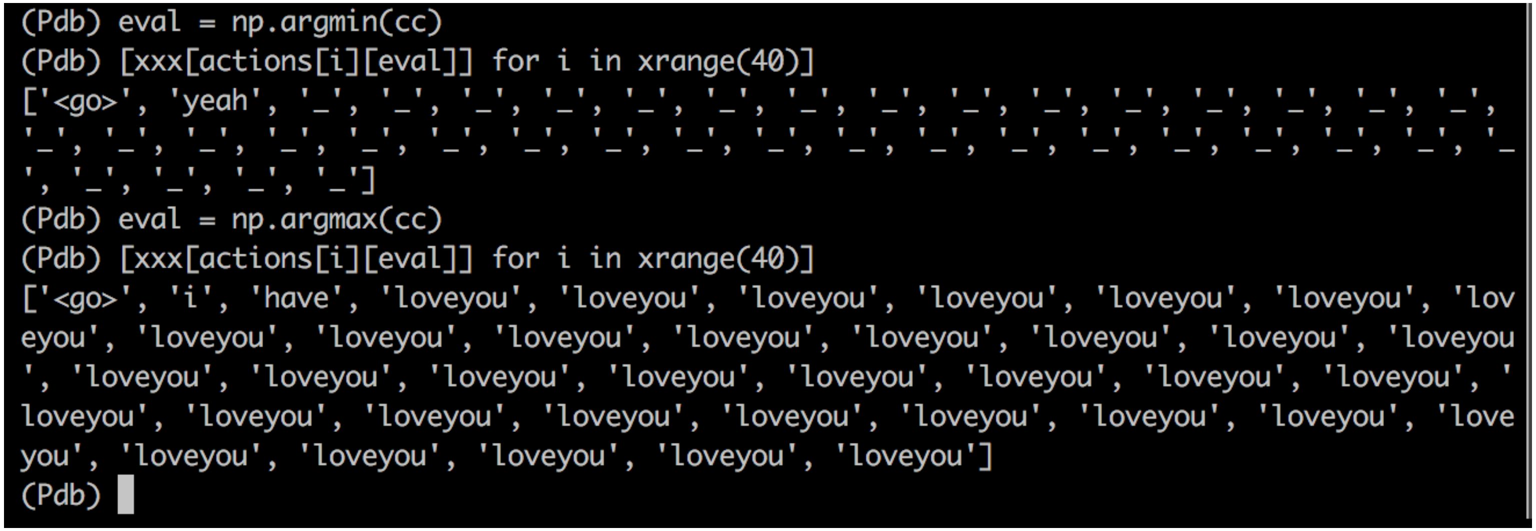
Result

- Mutual Information - Negative

3 layers of 512 units , after 20 steps

Additional Reward for length

& Penalty for Repeated Words

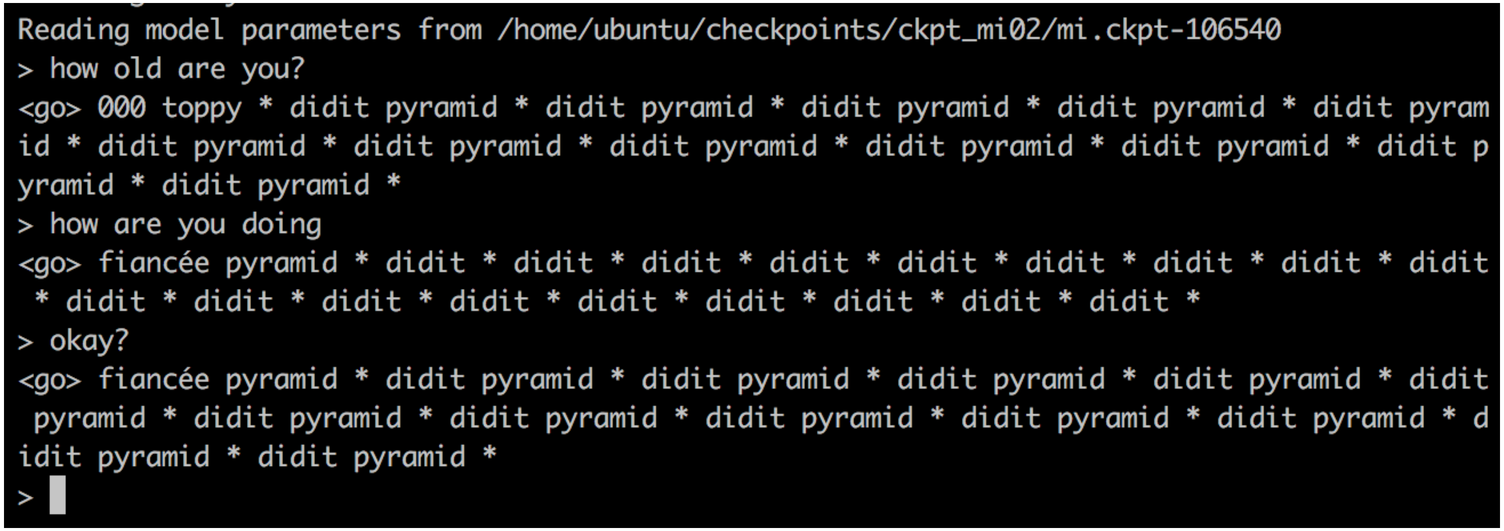


```
(Pdb) eval = np.argmin(cc)
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'yeah', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_',
'_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_
', '_', '_', '_', '_']
(Pdb) eval = np.argmax(cc)
(Pdb) [xxx[actions[i][eval]] for i in xrange(40)]
['<go>', 'i', 'have', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'lov
eyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou
', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', '
loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'love
you', 'loveyou', 'loveyou', 'loveyou', 'loveyou', 'loveyou']
(Pdb) 
```

Result

- Additional Reward for Longer Sentence
- Additional Penalty for Repeated Word, “no”, and ”I am sorry”

  (after 540 Global steps)



```
Reading model parameters from /home/ubuntu/checkpoints/ckpt_mi02/mi.ckpt-106540
> how old are you?
<go> 000 toppy * didit pyramid * didit pyramid * didit pyramid * didit pyramid * didit pyram
id * didit pyramid * didit pyramid * didit pyramid * didit pyramid * didit pyramid * didit p
yramid * didit pyramid *
> how are you doing
<go> fiancée pyramid * didit * didit * didit * didit * didit * didit * didit * didit * didit
 * didit * didit * didit * didit * didit * didit * didit * didit * didit *
> okay?
<go> fiancée pyramid * didit pyramid * didit pyramid * didit pyramid * didit pyramid * didit
 pyramid * didit pyramid * didit pyramid * didit pyramid * didit pyramid * didit pyramid * d
idit pyramid * didit pyramid *
> 
```

Discussion

- Movie Dataset
- Curriculum Learning Strategy
- Trade-off of different reward functions – weights

Demo

SEQ2SEQ
SEQ2SEQ Backward