Controllable Text Generation: Types, Knowledge, and Logic

Dr. Minlie Huang (黄民烈)

Associate Professor

CS Department, Tsinghua University

aihuang@tsinghua.edu.cn

http://coai.cs.tsinghua.edu.cn/hml

Research Picture of the CoAI Group

Conversational AI

- Natural Language Understanding
- Deep Learning
- Reinforcement Learning

- Task-Oriented Dialogue
- Social Chatbot
- Question Answering
- Reading Comprehension
- Sentiment Analysis

Recent Papers on RL

- Learning Structured Representation with RL (**AAAI 2018**)
- Data Denoising in Relation Extraction (**AAAI 2018**)
  - 入选**PaperWeekly** 2017年度最值得读的10篇NLP论文
- Weakly Supervised Label Correction (**IJCAI-ECAI 2018**)
- Hierarchical Relation Extraction (submitted to **AAAI 2019**)
- Learning to Collaborate: Joint Ranking Optimization (**WWW 2018**)
  - Multi-agent reinforcement learning; deterministic policy; actor-critic
- Search Result Aggregation (in preparation)

Recent Papers on Dialogue

- Perceiving and Expressing **Emotions** (**AAAI 2018**)
- Proactive Behavior by **Asking Good Questions** (**ACL 2018**)
- Controlling **Sentence Function** (**ACL 2018**)
- **Topic** Change in Multi-turn Dialog Systems (**SIGIR 2018**)
- **Explicit Personality** Assignment (**IJCAI-ECAI 2018**)
- Better Understanding and Generation Using **Commonsense Knowledge** (**IJCAI-ECAI 2018 Distinguished Paper**)
- **Discourse parsing** in multi-party dialogues (**AAAI 2019**)
- **Memory** augmented dialog management (submitted to TOIS)
- **Multimodality** neural belief tracker (submitted to **WWW2019**)

Good Examples from LG Models

User：最幸福的人不是拿到了最好的东西，而是学会去珍惜手上已经拥有的人。

The happiest person is not the one who gets the best thing, instead who cherishes what she/he already has.

Machine: 是的，所以我也要珍惜自己的幸福

Agree, I will cherish my happiness too.

User: 时间啊，你这把杀猪刀。。。

Time is a killer…

Machine: 我是被杀的

I am the one who was killed.

User: 怕孤獨，又捨不下自己的小自由。

Afraid to be lonely, but wont' give up my little freedom.

Machine: 我想去流浪，但没有人陪我去。

I want to wander, but no one accompany me.

Bad Examples are More!

User: 其实，他没有那么爱你。

Ohh, he does not love you that much.

Machine: 幸福的一天开始了。

A happy day starts.

User: 回北京了！很冷很高兴！

Back to Beijing now. So cold so happy!

Machine: 什么时候来北京啊？

When will you come to Beijing?

User: 我来自上海。

I come from Shanghai.

Machine：真巧，我来自海南，老乡啊

So cool, I am from Hainan, the same province!

Controllability of Text Generation

- Three **fundamental problems** in current neural language generation models
  - **Semantics**
  - **Consistency** (long text generation)
  - **Logic** (reasonable and making sense)

Conditional Text Generation

- Free Text to Text
  - Dialogue Generation
  - Story Generation
  - Abstractive Summarization
- Structured Data to Text
- Category to Text
  - Emotion/Sentiment Generation
- Keywords to Text
  - Poetry Generation
  - Essay/Narrative/Story Generation
- Image/Video to Text
  - Captioning
  - Visual story-telling
- Generation from Scratch: random variable

In this talk

- **Types**: Question Generation in Conversational Systems (ACL 2018)
- **Knowledge**: Commonsense-aware Dialogue Generation (**IJCAI-ECAI 2018 Distinguished Paper**)
- **Logic**: Storing Ending Generation (AAAI 2019)

Typed Decoder

for Language Generation

Question Generation in Conversational Systems

我昨天晚上去聚餐了

I went to dinner yesterday night.

Yansen Wang, Chenyi Liu, Minlie Huang, Liqiang Nie.
Learning to ask questions in open-domain conversation systems. **ACL 2018**.

Question Generation in Conversational Systems

- Asking **good** questions requires **scene understanding**

Scene: Dining at a restaurant

我昨天晚上去聚餐了
I went to dinner yesterday night.

Friends?	Place?	Food?	…	Persons?	Bill?
WHO	WHERE	HOW-ABOUT		HOW-MANY	WHO

Yansen Wang, Chenyi Liu, Minlie Huang, Liqiang Nie.
Learning to ask questions in open-domain conversation systems. **ACL 2018**.

Question Generation in Conversational Systems

- Responding + **asking** (Li et al., 2016)
- **Key proactive** behaviors (Yu et al., 2016)
- Asking good questions are indication of **machine understanding**
- Key differences to **traditional** question generation (eg., reading comprehension):
  - **Different goals**: Information seeking vs. Enhancing interactiveness and persistence of human-machine interactions
  - **Various patterns**: YES-NO, WH-, HOW-ABOUT, etc.
  - **Topic transition**: from topics in post to topics in response

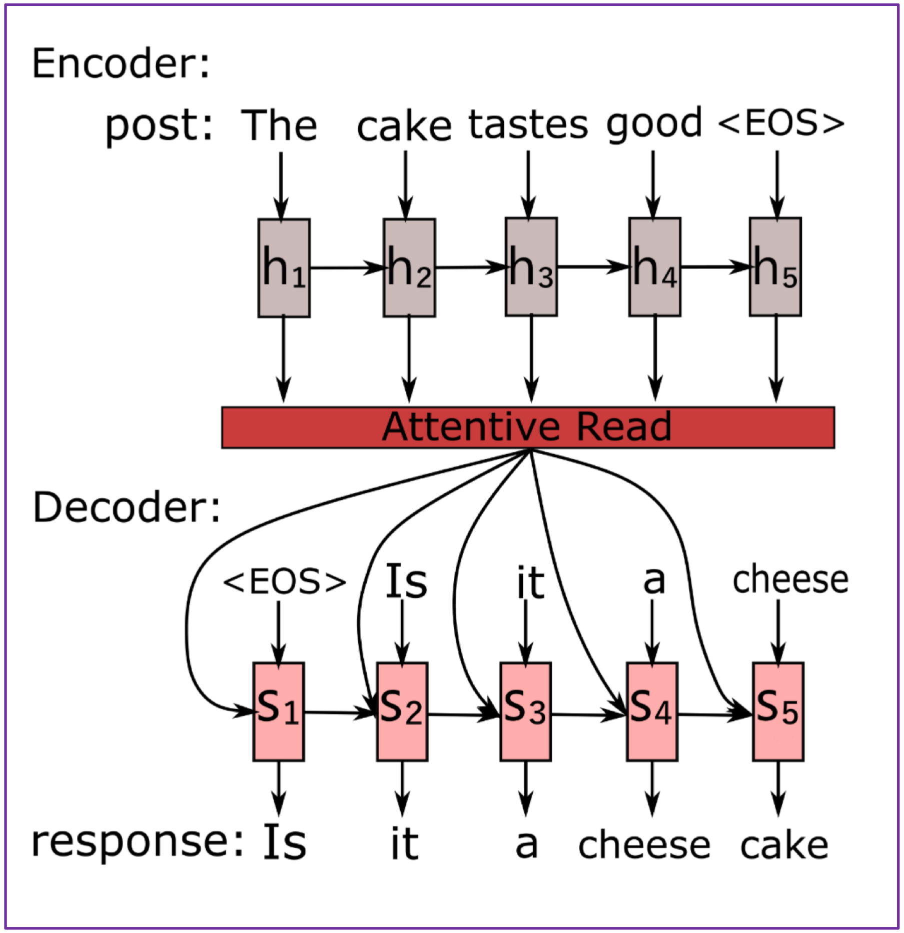
Question Generation in Conversational Systems

- A good question is a natural composition of
  - **Interrogatives** for using various questioning patterns
  - **Topic words** for addressing interesting yet novel topics
  - **Ordinary words** for playing grammar or syntactic roles

Example 1:
User: I am too fat ...
Machine: **How about** climbing this weekend?

Example 2:
User: Last night, I stayed in KTV with friends.
Machine: **Are** you happy with your singing?

Encoder-decoder Framework



$$X = x_1 x_2 \cdots x_m$$

$$Y = y_1 y_2 \cdots y_n$$

$$Y^* = \underset{Y}{argmax}\, \mathcal{P}(Y|X).$$

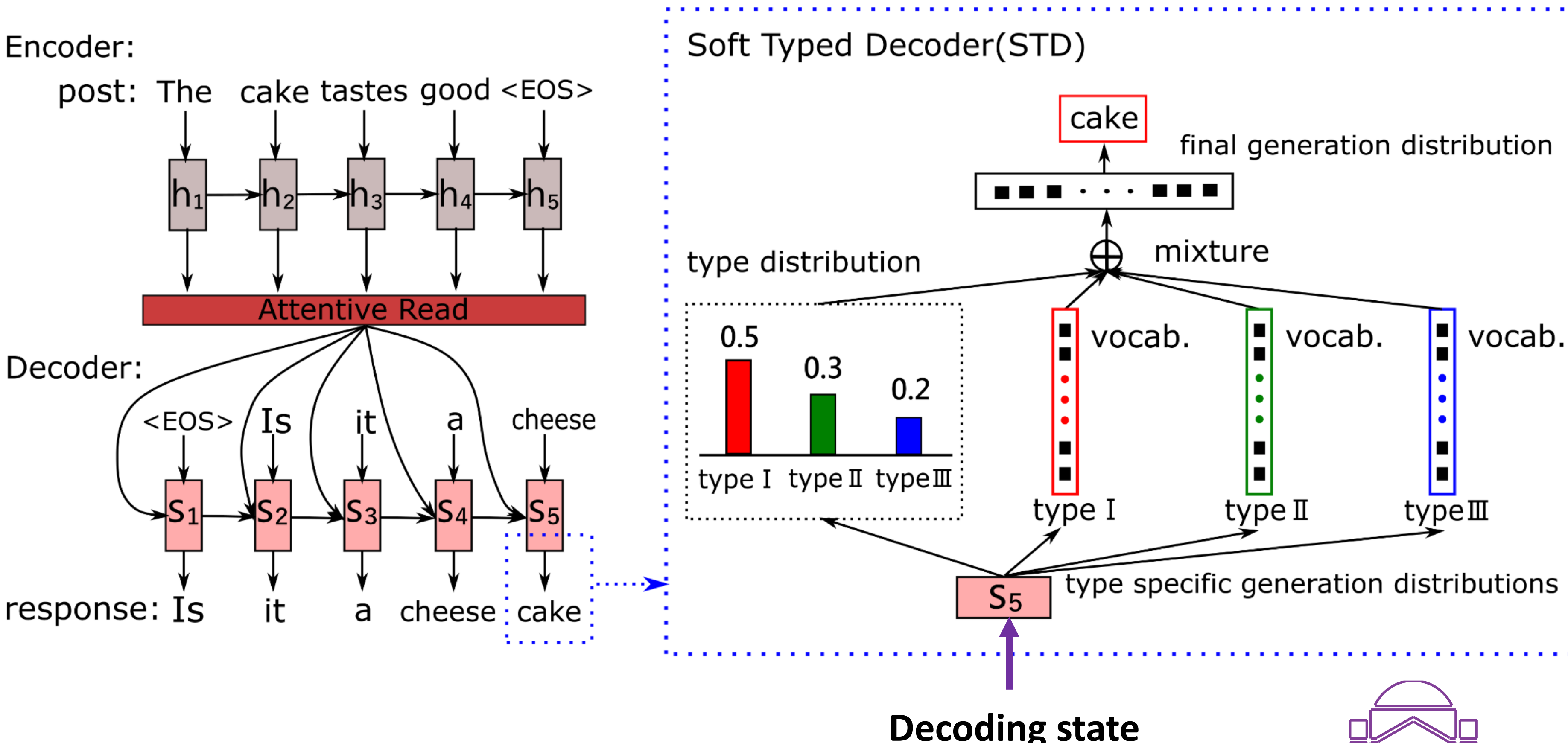
$$\mathcal{P}(y_t|y_{<t}, X) = \mathbf{MLP}(\mathbf{s}_t, \boldsymbol{e}(y_{t-1}), \mathbf{c}_t),$$

$$\mathbf{s}_t = \mathbf{GRU}(\mathbf{s}_{t-1}, \boldsymbol{e}(y_{t-1}), \mathbf{c}_t),$$

$$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i} \mathbf{h}_i$$

$$\mathbf{h}_t = \mathbf{GRU}(\mathbf{h}_{t-1}, \boldsymbol{e}(x_t)),$$

Soft Typed Decoder (STD)



Soft Typed Decoder (STD)

- Applying **multiple type-specific generation distributions** over the same vocabulary

- Each word has a **latent** distribution among the set type(w)∈{***interrogative***, ***topic word***, ***ordinary word***}

- STD is a very simple **mixture** model

$$\mathcal{P}(y_t|y_{<t},X)=\sum_{i=1}^{k}\underbrace{\mathcal{P}(y_t|ty_t=c_i,y_{<t},X)}_{\text{type-specific generation distribution}}\cdot\underbrace{\mathcal{P}(ty_t=c_i|y_{<t},X)}_{\text{word type distribution}},$$

Soft Typed Decoder (STD)

- Estimate the **type distribution** of each word:

$$\mathcal{P}(ty_t|y_{<t}, X) = softmax(\mathbf{W}_0\mathbf{s}_t + \mathbf{b}_0),$$

- Estimate the **type-specific generation distribution** of each word:

$$\mathcal{P}(y_t|\boxed{ty_t = c_i}, y_{<t}, X) = softmax(\mathbf{W}_{\boxed{c_i}}\mathbf{s}_t + \mathbf{b}_{c_i}),$$

- The final generation distribution is a **mixture** of the three type-specific generation distribution.

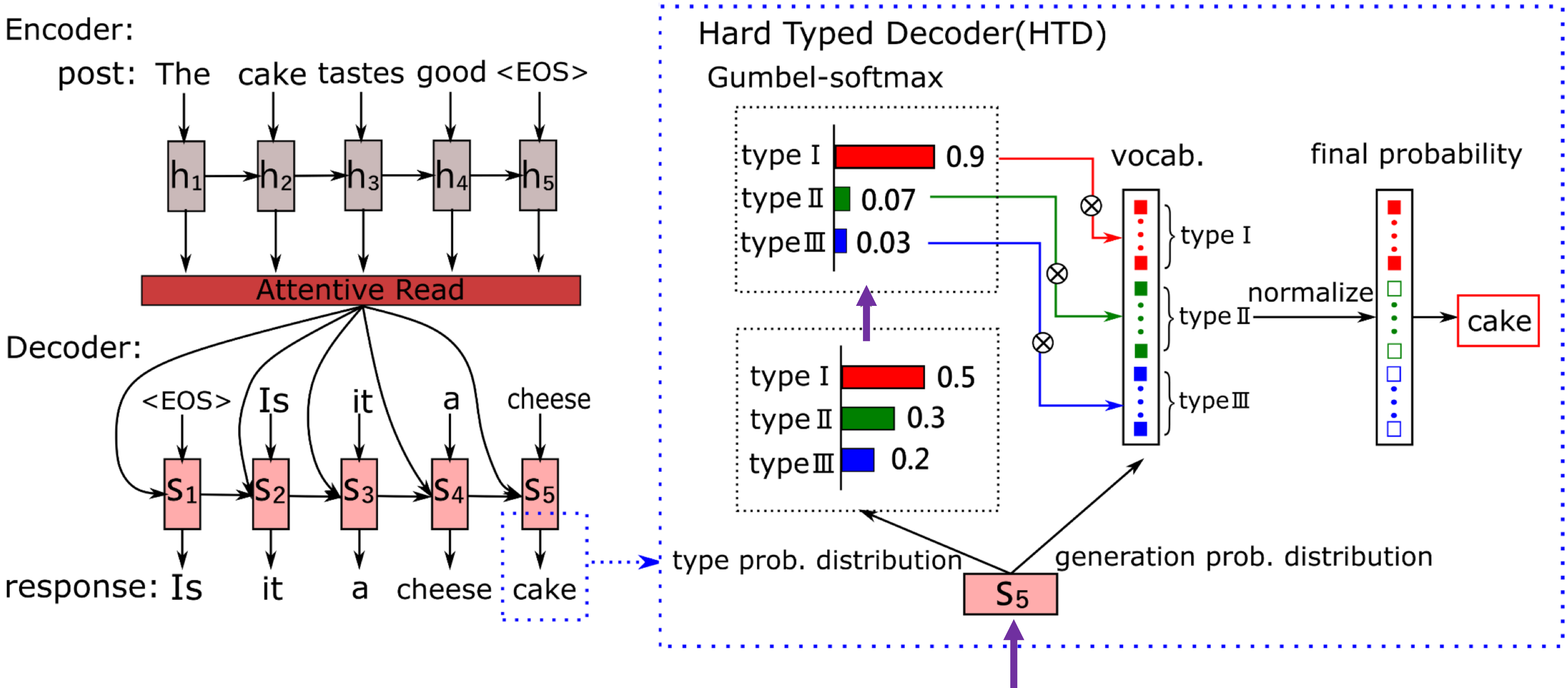
$$\mathcal{P}(y_t|y_{<t}, X) = \sum_{i=1}^{k} \mathcal{P}(y_t|ty_t = c_i, y_{<t}, X) \cdot \mathcal{P}(ty_t = c_i|y_{<t}, X),$$

Hard Typed Decoder (HTD)

- In soft typed decoder, word types are modeled in a **latent, implicit** way

- Can we control the word type more **explicitly** in generation?
    - Stronger control

Hard Typed Decoder (HTD)



Decoding state

Hard Typed Decoder (HTD)

- Estimate the generation probability distribution

$$\mathcal{P}(y_t|y_{<t},X) = softmax(\mathbf{W}_0\mathbf{s}_t + \mathbf{b}_0).$$

- Estimate the type probability distribution

$$\mathcal{P}(ty_t|y_{<t},X) = softmax(\mathbf{W}_1\mathbf{s}_t + \mathbf{b}_1).$$

- Modulate words' probability by its corresponding type probability:

$$\mathcal{P}'(y_t|y_{<t},X) = \mathcal{P}(y_t|y_{<t},X)\boxed{\cdot\boldsymbol{m}(y_t)},$$

m(*y*$_t$) is related to the type probability of word *y*$_t$

Hard Typed Decoder (HTD)

Generation distr.			Type distr.			Modulated distr.	
what	*0.3*		$T_{interrogative}$	*0.7*		*what*	*0.8*
food	*0.2*	X	T_{topic}	*0.1*	→	*food*	*0.05*
is	*0.4*		$T_{ordinary}$	*0.2*		*is*	*0.09*
....						....	

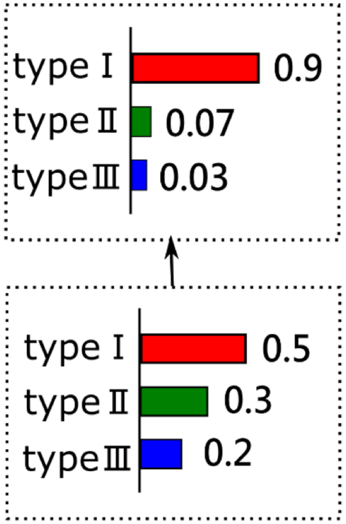
- **Argmax?** **(**firstly select largest type prob. then sample word from generation dist.**)**
  - Indifferentiable
  - Serious grammar errors if word type is wrongly selected

Hard Typed Decoder (HTD)

- **Gumble-Softmax**:
  - A differentiable surrogate to the **argmax** function.

$$\boldsymbol{m}(y_t) = \mathbf{GS}(\mathcal{P}(ty_t = c(y_t)|y_{<t}, X)),$$

$$\mathbf{GS}(\pi_i) = \frac{e^{(log(\pi_i)+g_i)/\tau}}{\sum_{j=1}^{k} e^{(log(\pi_j)+g_j)/\tau}},$$



Hard Typed Decoder (HTD)

- In HTD, the types of words **are given in advance**.
    - *How to determine the word types?*

Hard Typed Decoder (HTD)

- **Interrogatives**:
  - A list of about 20 interrogatives are given by hand.
- **Topic words:**
  - Training: all nouns and verbs in response are topic words.
  - Test: 20 words are predicted by PMI.

$$PMI(w_x, w_y) = log \frac{p(w_x, w_y)}{p_1(w_x) * p_2(w_y)},$$

$$Rel(k_i, X) = \sum_{w_x \in X} e^{PMI(w_x, k_i)},$$

- **Ordinary words:**
  - All other words, for grammar or syntactic roles

Loss Function

- Cross entropy
- Supervisions are on both final probability and the type distribution:

$$\Phi_1 = \sum_t -\log \mathcal{P}(y_t = \tilde{y}_t | y_{<t}, X),$$

$$\Phi_2 = \sum_t -\log \mathcal{P}(ty_t = \widetilde{ty}_t | y_{<t}, X),$$

$$\Phi = \Phi_1 + \lambda \Phi_2,$$

- λ is a term to balance the two kinds of losses.

Dataset

- PMI estimation: calculated from **9 million post-response** pairs from Weibo.

- Dialogue Question Generation Dataset(DQG), about **491,000 pairs**：
  - Distilled questioning responses using about 20 hand-draft templates
  - Removed universal questions
  - Available at http://coai.cs.tsinghua.edu.cn/hml/dataset/

Baselines

- **Seq2Seq:** A simple encoder-decoder model (Luong et al., 2015)
- **Mechanism-Aware (MA):** Multiple responding mechanisms represented by real-valued vectors (Zhou et al., 2017)
- **Topic-Aware (TA):** Topic Aware Model by incorporating topic words (Xing et al., 2017)
- **Elastic Responding Machine (ERM)**: Enhanced MA using reinforcement learning (Zhou et al., 2018)

Automatic Evaluation

Model	Perplexity	Distinct-1	Distinct-2	TRR
Seq2Seq	63.71	0.0573	0.0836	6.6%
MA	**54.26**	0.0576	0.0644	4.5%
TA	58.89	0.1292	0.1781	8.7%
ERM	67.62	0.0355	0.0710	4.5%
STD	56.77	0.1325	0.2509	12.1%
HTD	56.10	**0.1875**	**0.3576**	**43.6%**

Table 1: Results of automatic evaluation.

Evaluation metrics

- **Perplexity & Distinct**
- **TRR (Topical Response Ratio)**:
  - 20 topic words are predicted with PMI for each post.
  - TRR is the proportion of the responses containing at least one topic word.

Manual Evaluation

- Pair-wise comparison: win, loss, tie
- Three evaluation criteria:
  - **Appropriateness:** whether a question is reasonable in logic and content, and has key info.
  - **Richness:** containing topic words or not
  - **Willingness** to respond to a generated question

Manual Evaluation(Pairwise)

Models	Appropriateness			Richness			Willingness		
	Win (%)	Lose (%)	Tie (%)	Win (%)	Lose (%)	Tie (%)	Win (%)	Lose (%)	Tie (%)
STD vs. Seq2Seq	42.0	38.6	19.4	37.2**	15.2	47.6	45.4*	38.6	16.0
STD vs. MA	39.6*	31.2	29.2	32.6**	16.8	50.6	49.4**	27.0	23.6
STD vs. TA	42.2	40.0	17.8	49.0**	5.4	45.6	47.6*	40.2	12.2
STD vs. ERM	43.4*	34.4	22.2	60.6**	13.2	26.2	43.2*	36.8	20.0
HTD vs. Seq2Seq	50.6**	30.6	18.8	46.0**	10.2	43.8	58.4**	33.2	8.4
HTD vs. MA	54.8**	24.4	20.8	45.0**	17.0	38.0	67.0**	18.0	15.0
HTD vs. TA	52.0**	38.2	9.8	55.0**	5.4	39.6	62.6**	31.0	6.4
HTD vs. ERM	64.8**	23.2	12.0	72.2**	8.4	19.4	56.6**	36.6	6.8
HTD vs. STD	52.0**	33.0	15.0	38.0**	26.2	35.8	61.8**	30.6	7.6

Table 2: Annotation results. Win for "A vs. B" means A is better than B. Significance tests with Z-test were conducted. Values marked with * means *p-value* < 0.05, and ** for *p-value* < 0.01.

Score: the probability of win/lose/tie of our model vs. baseline

Examples

Post:	我们昨天去唱歌了 We went to karaoke yesterday.
Seq2Seq:	什么时候? **When?**
MA:	你怎么知道我是在问 **How** did you know I am questioning you?
TA:	什么? **What?**
ERM:	什么情况? **What** happened ?
STD:	去哪里唱歌? **Where** did you sing karaoke?
HTD:	你们几个人唱歌? **How many** people were singing with you?

Table 4: Generated question examples. Interrogative words are bolded and topic words underlined.

More Examples

- Different questioning patterns and topic transition:

Post-3:	我在体育中心上班。 I work at the sports center.
Human:	你每天从家过去要多久? **How** long does it take you to go there from home**?**
STD:	你是**什么**部门的啊**?** **Which** department are you from**?**
HTD:	那你是体育学院的学生**吗?** **Are** you a student from the College of Physical Education**?**
Post-4:	今天的寿司真好吃。 The sushi today is so delicious!
Human:	我记得你不喜欢吃日式料理的不是**吗?** I remember you didn't like eating Japanese food, **did** you**?**
STD:	你什么时候来无锡，我请你。 **When** will you come to Wuxi**?** It is my treat.
HTD:	是**吗?**改天尝尝! Really**?** I will try sometime!

Work→Department
Sports→ College…

Suchi→Treat
Suchi→Try

Visualization

- Type prediction at each decoding position

Post: 我喜欢小动物(I like little animals)

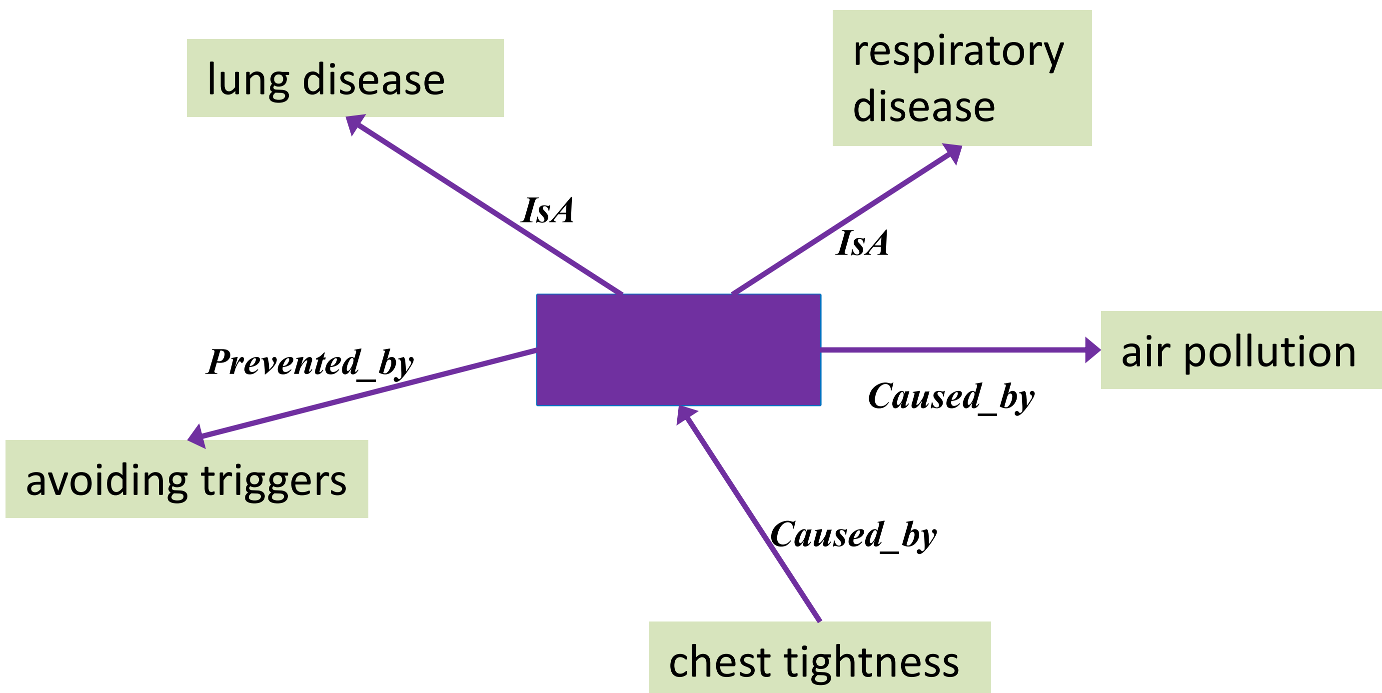
Response:	你(you)	喜欢(like)	兔子(rabbit)	吗(*particle*)	?	_EOS
Interrogative	0.09	0.02	0.01	0.85	1.00	0.01
Topic word	0.26	0.35	0.71	0.14	0.00	0.02
Ordinary word	0.65	0.63	0.28	0.01	0.00	0.97
Decoding steps	1	2	3	4	5	6

Knowledge
in
Language Generation

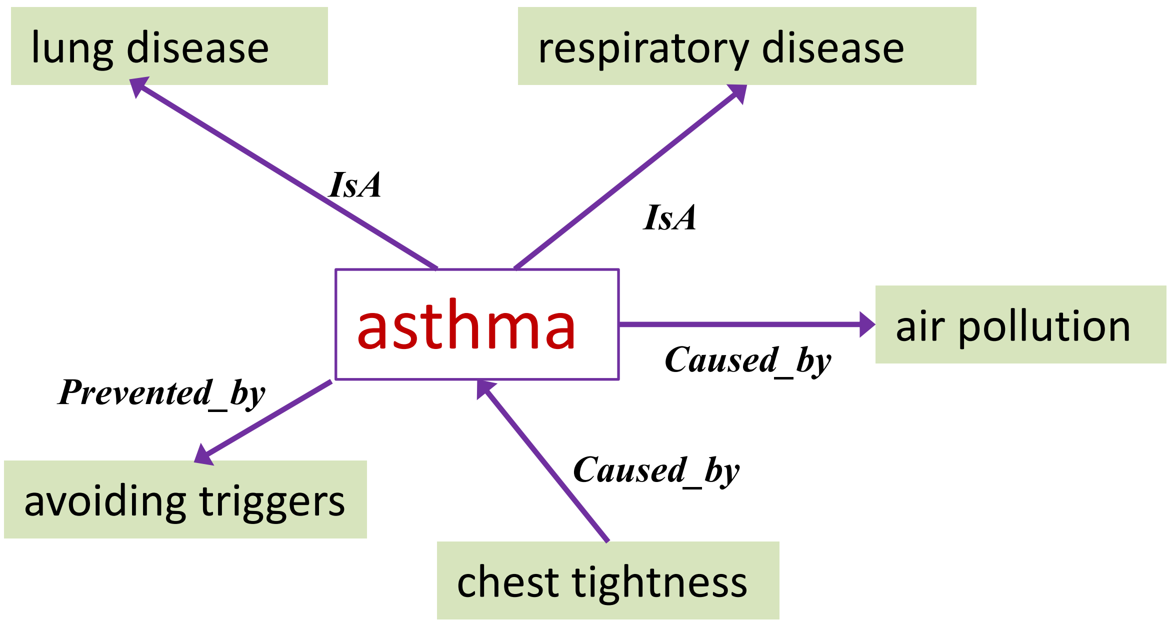
Commonsense Knowledge

- **Commonsense knowledge** consists of facts about the everyday world, that all humans are expected to know. (Wikipedia)
  - Lemons are sour
  - Tree has leafs
  - Dog has four legs
- Commonsense Reasoning ~ **Winograd Schema Challenge**:
  - The trophy would not fit in the brown suitcase because it was too **big**. What was too **big**?
  - The trophy would not fit in the brown suitcase because it was too ***small***. What was too ***small***?

Commonsense Knowledge



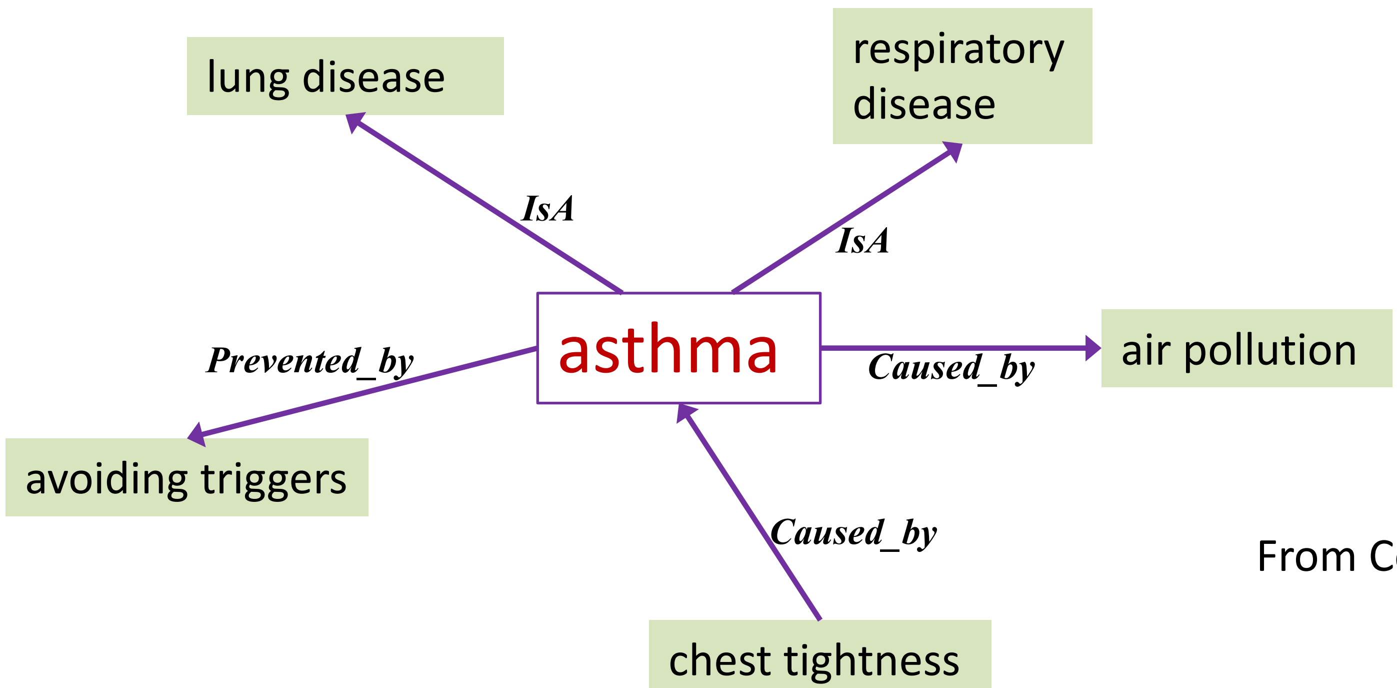
Commonsense Knowledge



Commonsense Knowledge

Post: I have an asthma since three years old.

Triples in knowledge graph:
(lung disease, IsA, asthma)
(asthma, Prevented_by, avoiding triggers)



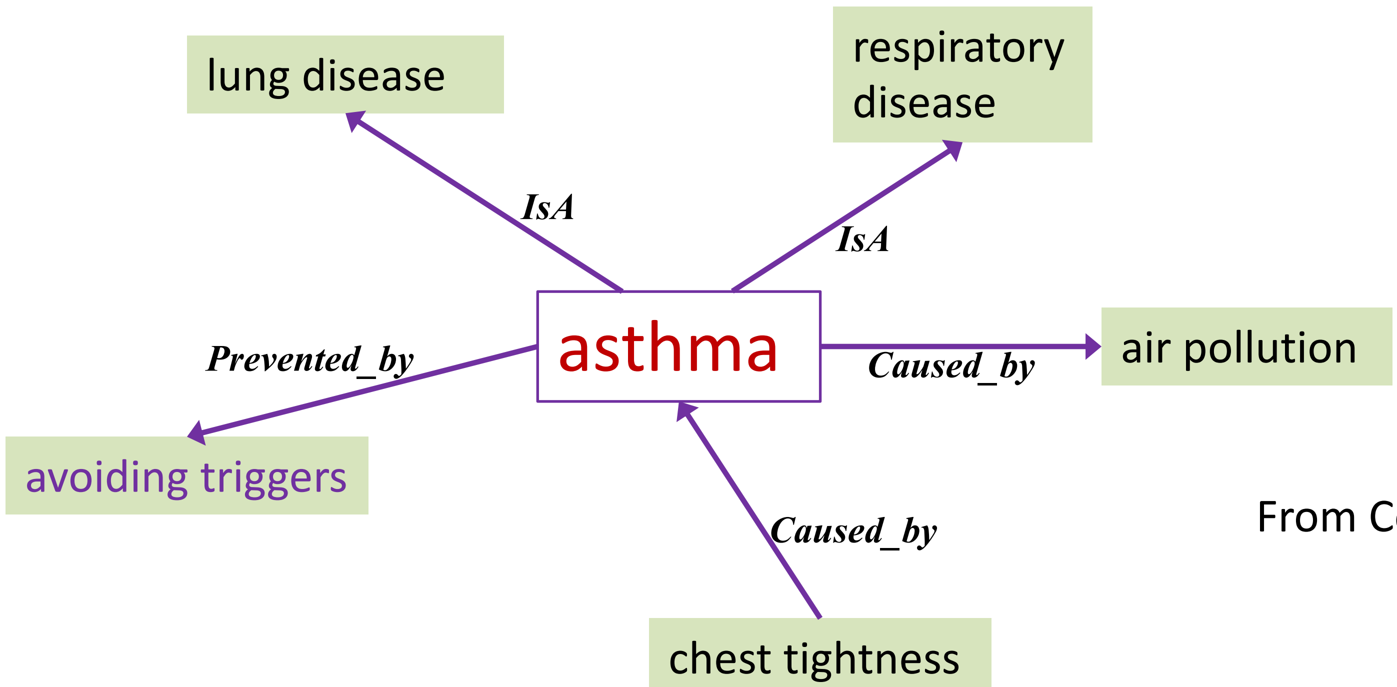
From ConceptNet

Commonsense Knowledge in Chatbots

Post: I have an asthma since three years old.

Triples in knowledge graph:
(lung disease, IsA, asthma)
(asthma, Prevented_by, avoiding triggers)

Response: I am sorry to hear that. Maybe avoiding triggers can prevent asthma attacks.



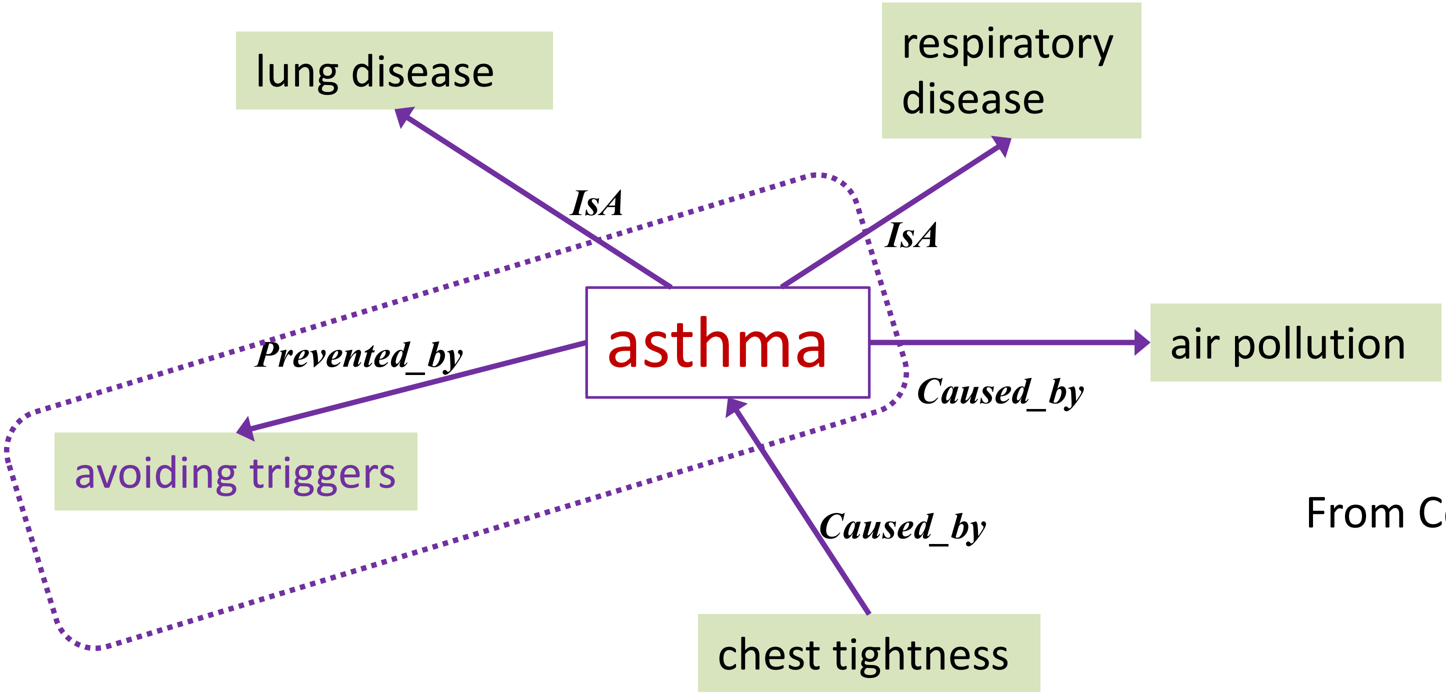
From ConceptNet

Commonsense Knowledge in Chatbots

Post: I have an asthma since three years old.

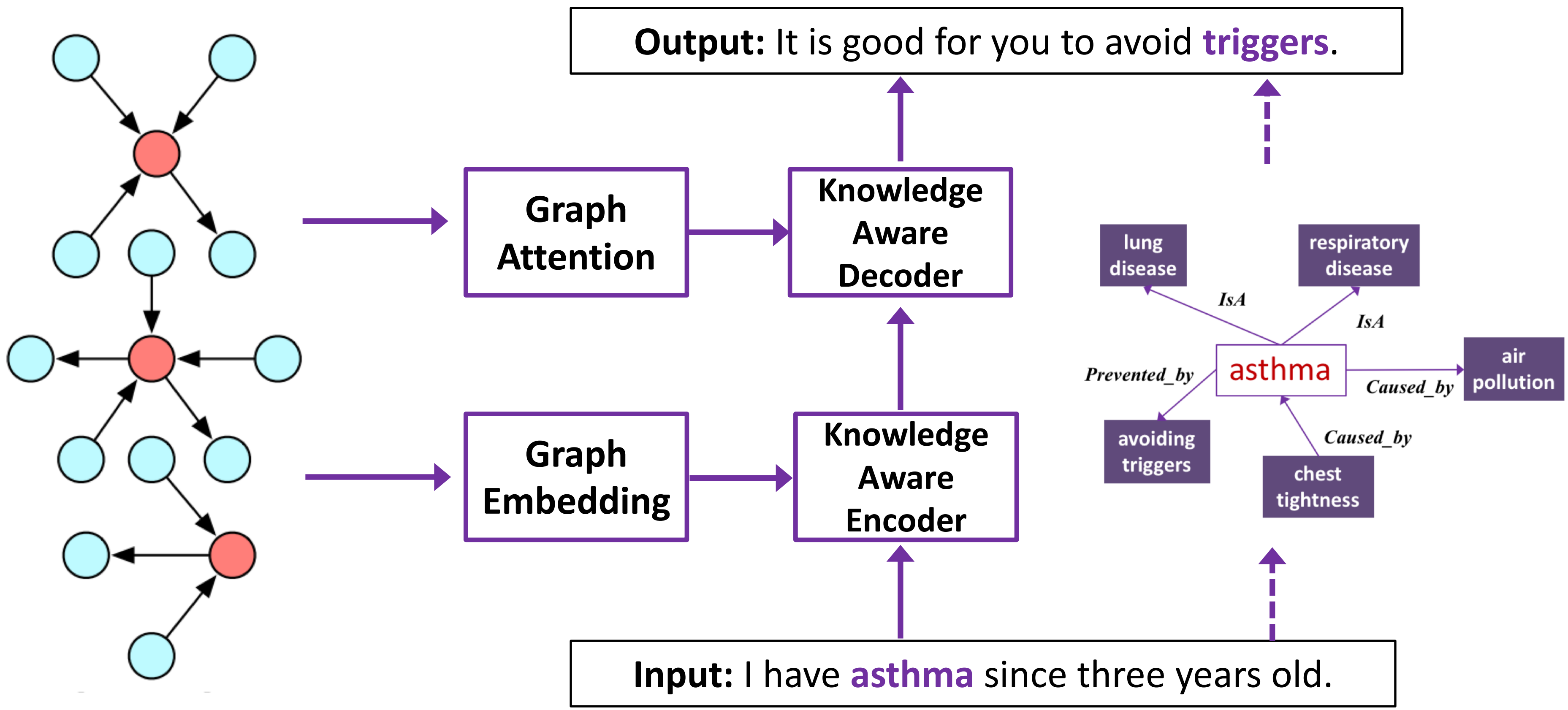
Triples in knowledge graph:
(lung disease, IsA, asthma)
(asthma, Prevented_by, avoiding triggers)

Response: I am sorry to hear that. Maybe avoiding triggers can prevent asthma attacks.



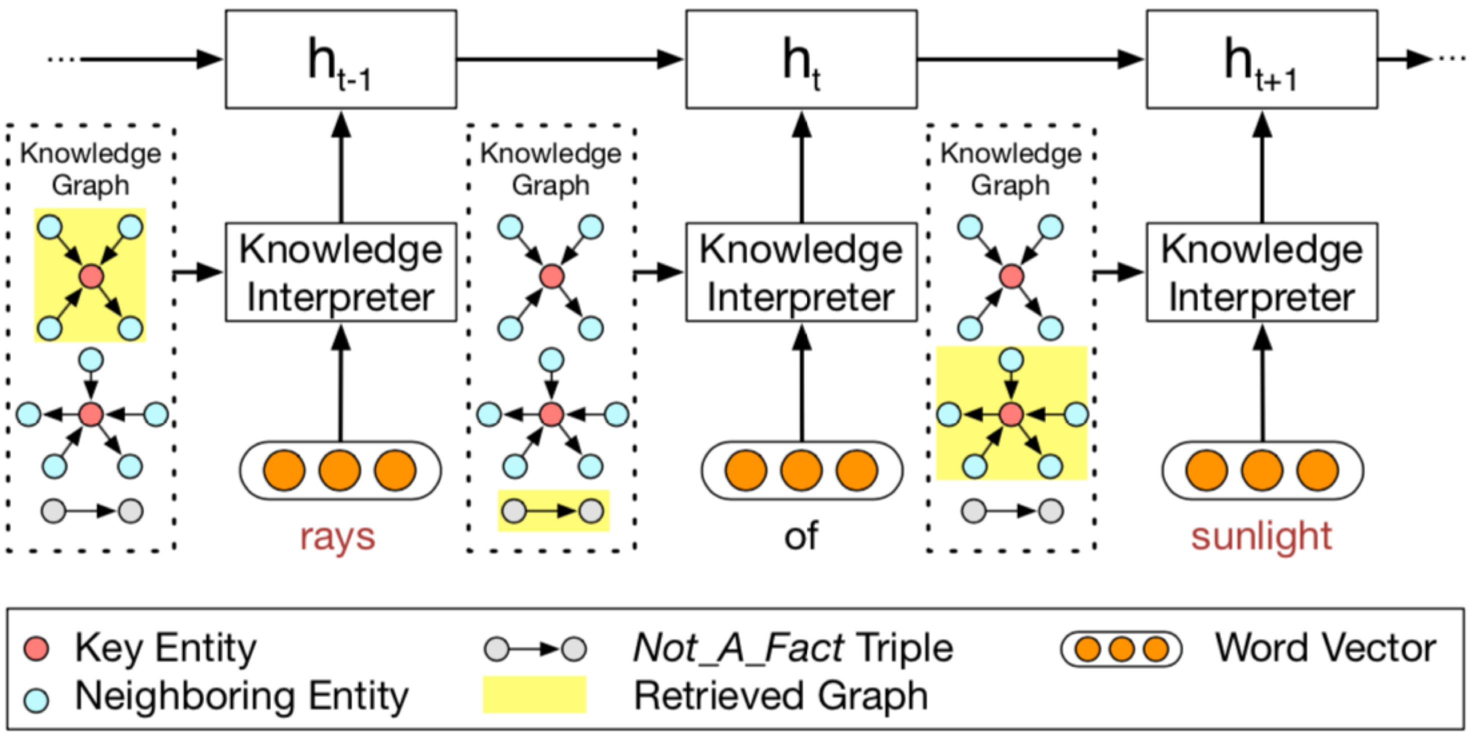
From ConceptNet

Commonsense-aware Dialog Generation



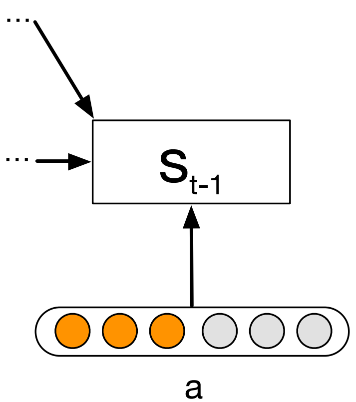
Commonsense Knowledge in Chatbots

Static graph attention: encoding semantics in graph, Feeding knowledge-enhanced info. into the encoder

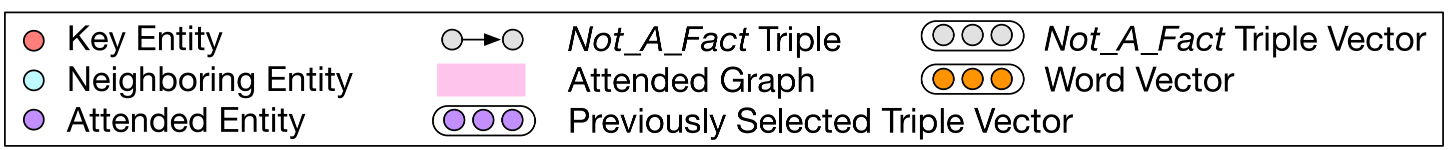


Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph

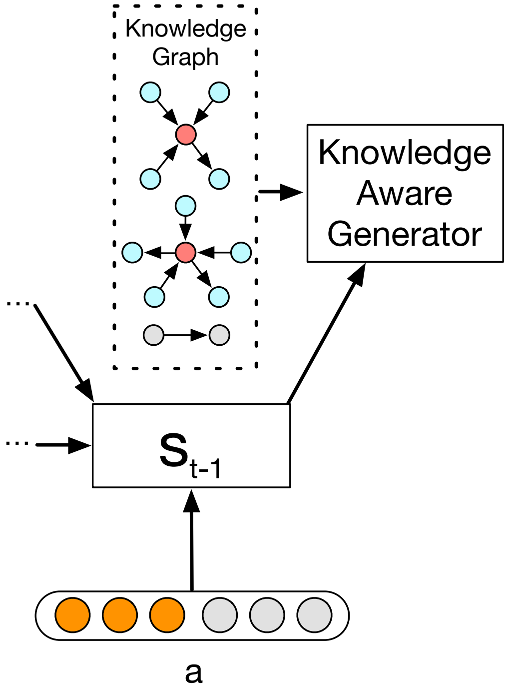


$$
\begin{aligned}
\boldsymbol{s}_{t+1} &= \mathbf{GRU}(\boldsymbol{s}_t, [\boldsymbol{c}_t; \boldsymbol{c}_t^g; \boldsymbol{c}_t^k; \boldsymbol{e}(y_t)]), \\
\boldsymbol{e}(y_t) &= [\boldsymbol{w}(y_t); \boldsymbol{k}_j],
\end{aligned}
$$

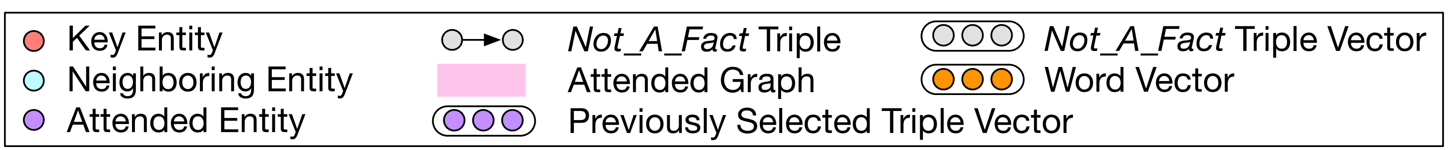


Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph

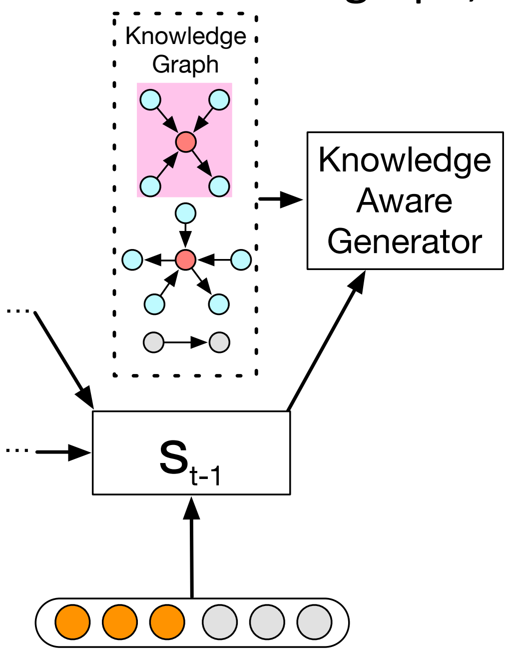


$$
\begin{aligned}
\boldsymbol{g}_i &= \sum_{n=1}^{N_{g_i}} \alpha_n^s [\boldsymbol{h}_n; \boldsymbol{t}_n], \\
\alpha_n^s &= \frac{\exp(\beta_n^s)}{\sum_{j=1}^{N_{g_i}} \exp(\beta_j^s)}, \\
\beta_n^s &= (\mathbf{W_r} \boldsymbol{r}_n)^\top \tanh(\mathbf{W_h} \boldsymbol{h}_n + \mathbf{W_t} \boldsymbol{t}_n),
\end{aligned}
$$

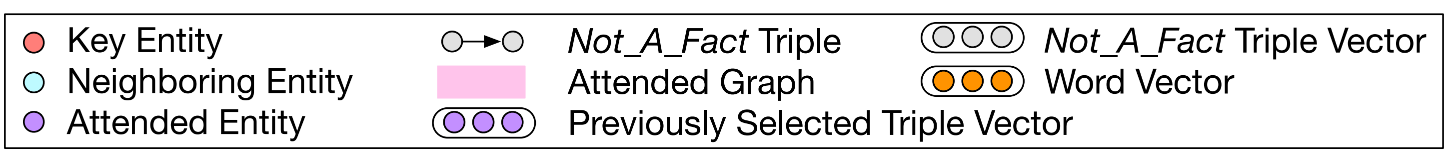


Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph

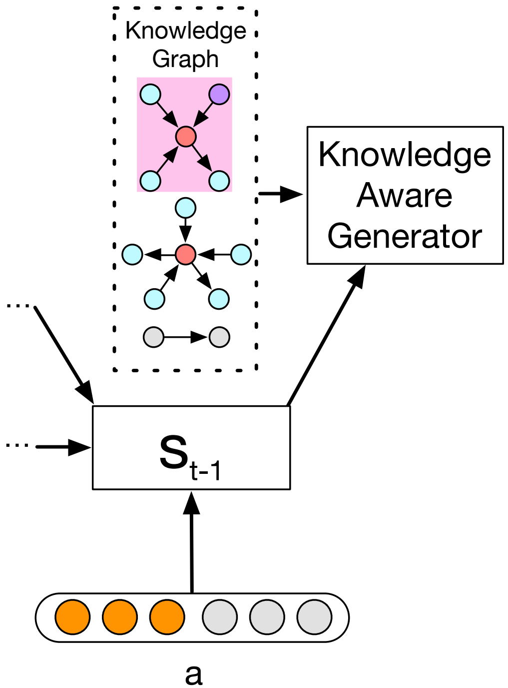


$$
\begin{aligned}
\boldsymbol{c}_t^g &= \sum_{i=1}^{N_G} \alpha_{ti}^g \boldsymbol{g}_i, \\
\alpha_{ti}^g &= \frac{\exp(\beta_{ti}^g)}{\sum_{j=1}^{N_G} \exp(\beta_{tj}^g)}, \\
\beta_{ti}^g &= \boldsymbol{V_b}^\top \tanh(\mathbf{W_b} \boldsymbol{s}_t + \mathbf{U_b} \boldsymbol{g}_i),
\end{aligned}
$$



Commonsense Knowledge in Chatbots

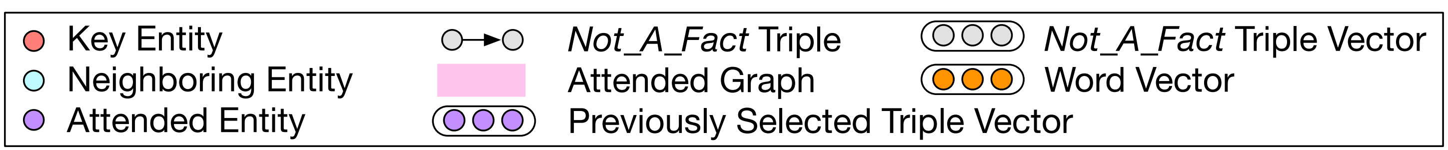
Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph



$$c_t^k = \sum_{i=1}^{N_G} \sum_{j=1}^{N_{g_i}} \alpha_{ti}^g \alpha_{tj}^k k_j,$$

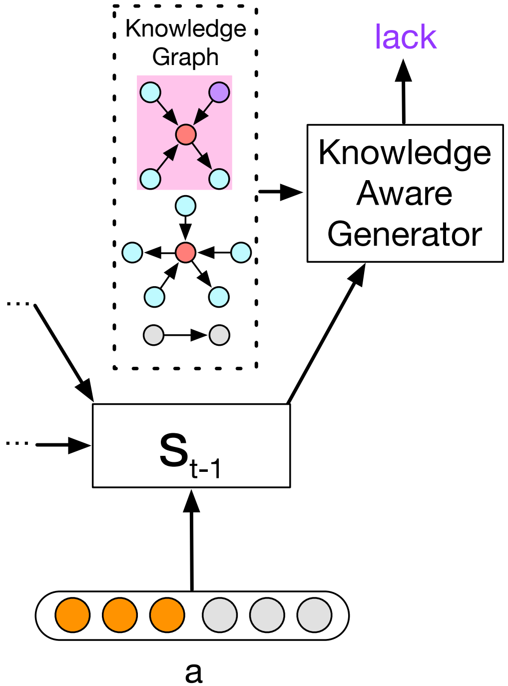
$$\alpha_{tj}^k = \frac{\exp(\beta_{tj}^k)}{\sum_{n=1}^{N_{g_i}} \exp(\beta_{tn}^k)},$$

$$\beta_{tj}^k = k_j^\top \mathbf{W_c} s_t,$$

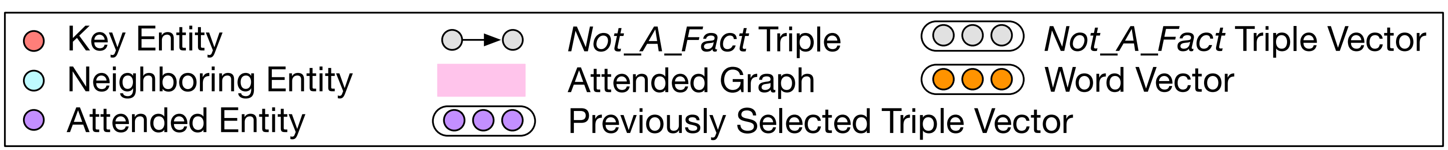


Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph

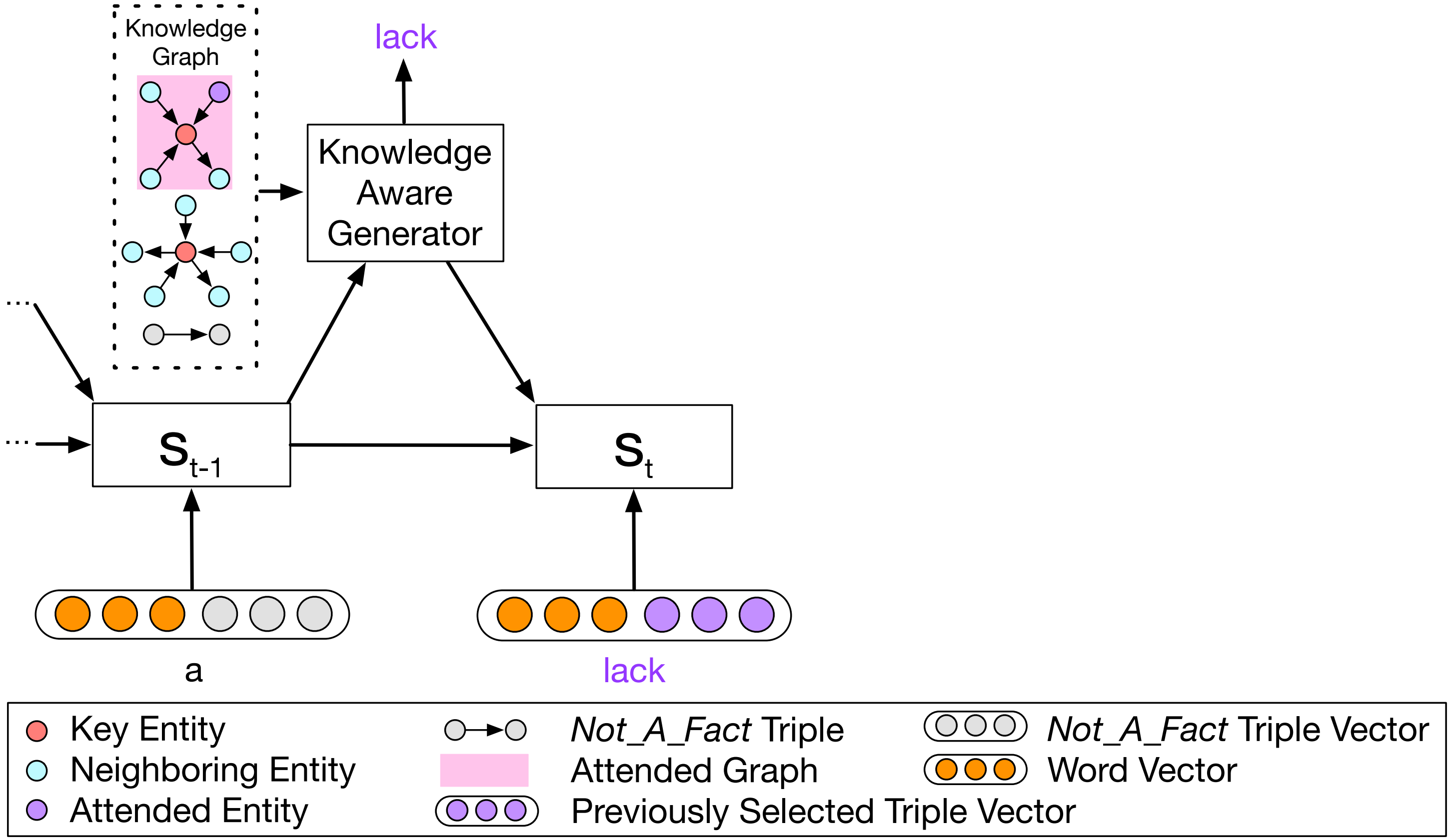


$$
\begin{aligned}
\boldsymbol{a}_t &= [\boldsymbol{s}_t; \boldsymbol{c}_t; \boldsymbol{c}_t^g; \boldsymbol{c}_t^k], \\
\gamma_t &= \text{sigmoid}(\mathbf{V_o}^\top \boldsymbol{a}_t), \\
P_c(y_t = w_c) &= \text{softmax}(\mathbf{W_o} \boldsymbol{a}_t), \\
P_e(y_t = w_e) &= \alpha_{ti}^g \alpha_{tj}^k, \\
y_t \sim \boldsymbol{o}_t = P(y_t) &= \begin{bmatrix} (1-\gamma_t) P_g(y_t = w_c) \\ \gamma_t P_e(y_t = w_e) \end{bmatrix},
\end{aligned}
$$



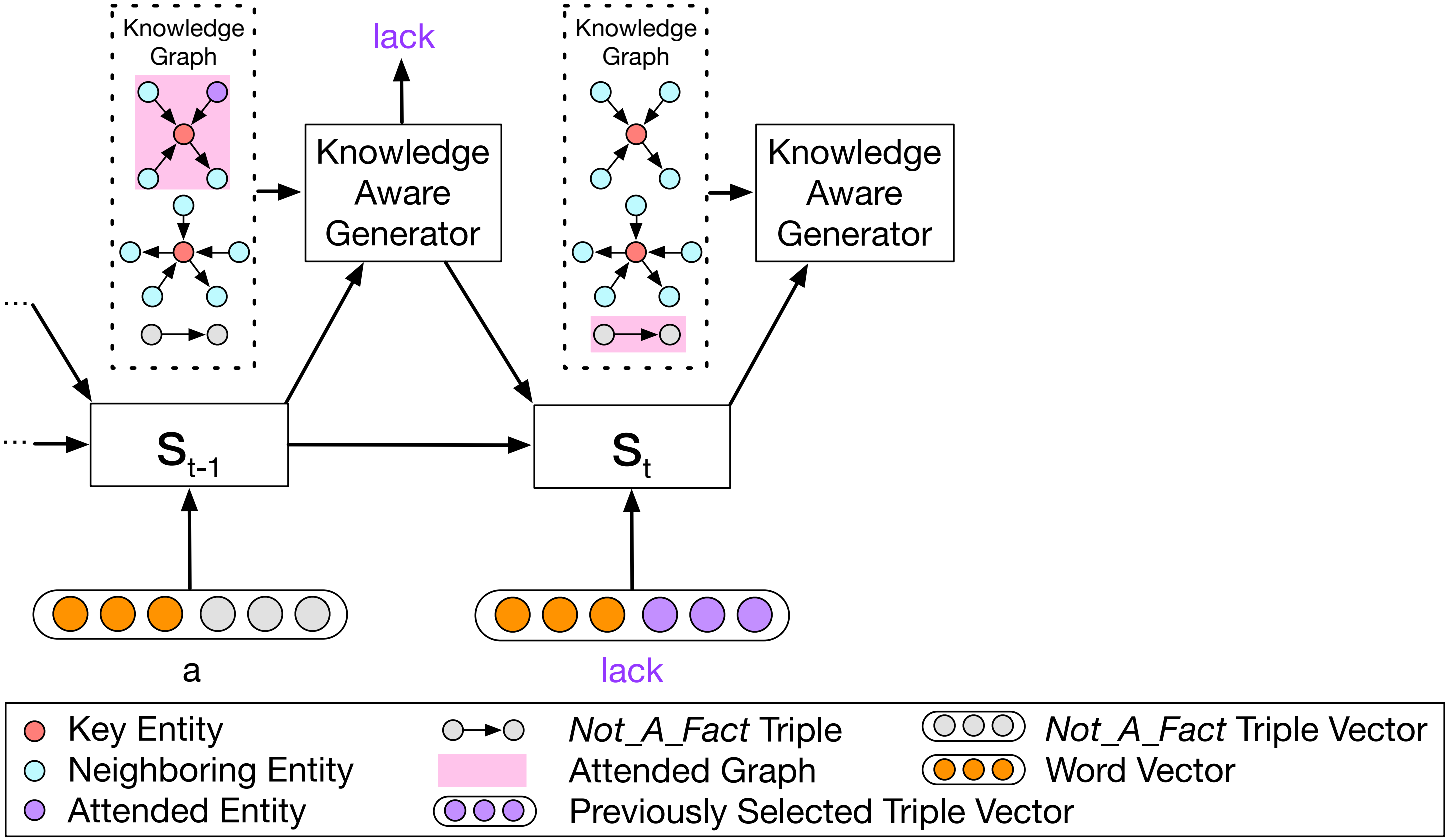
Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph



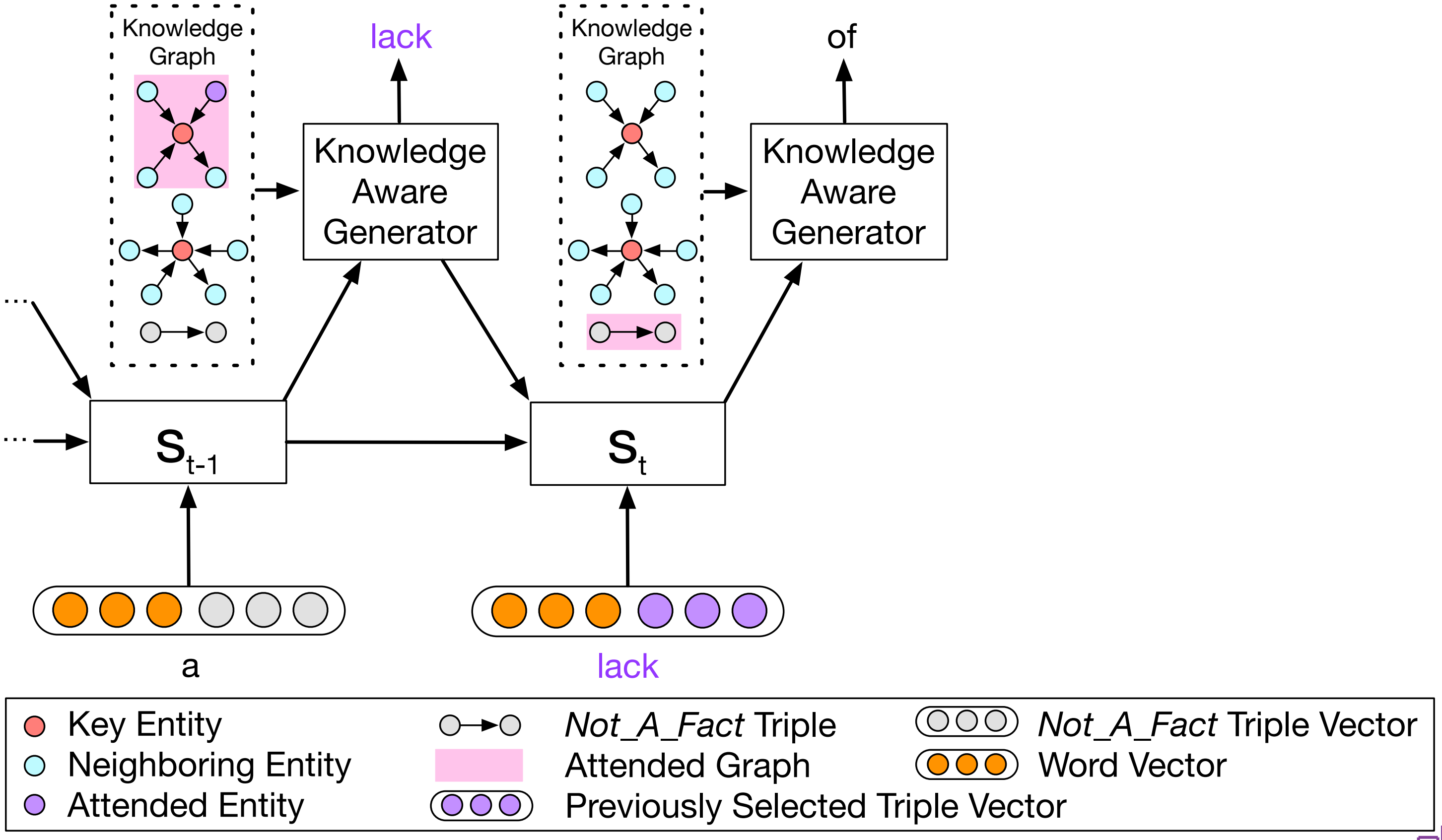
Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph



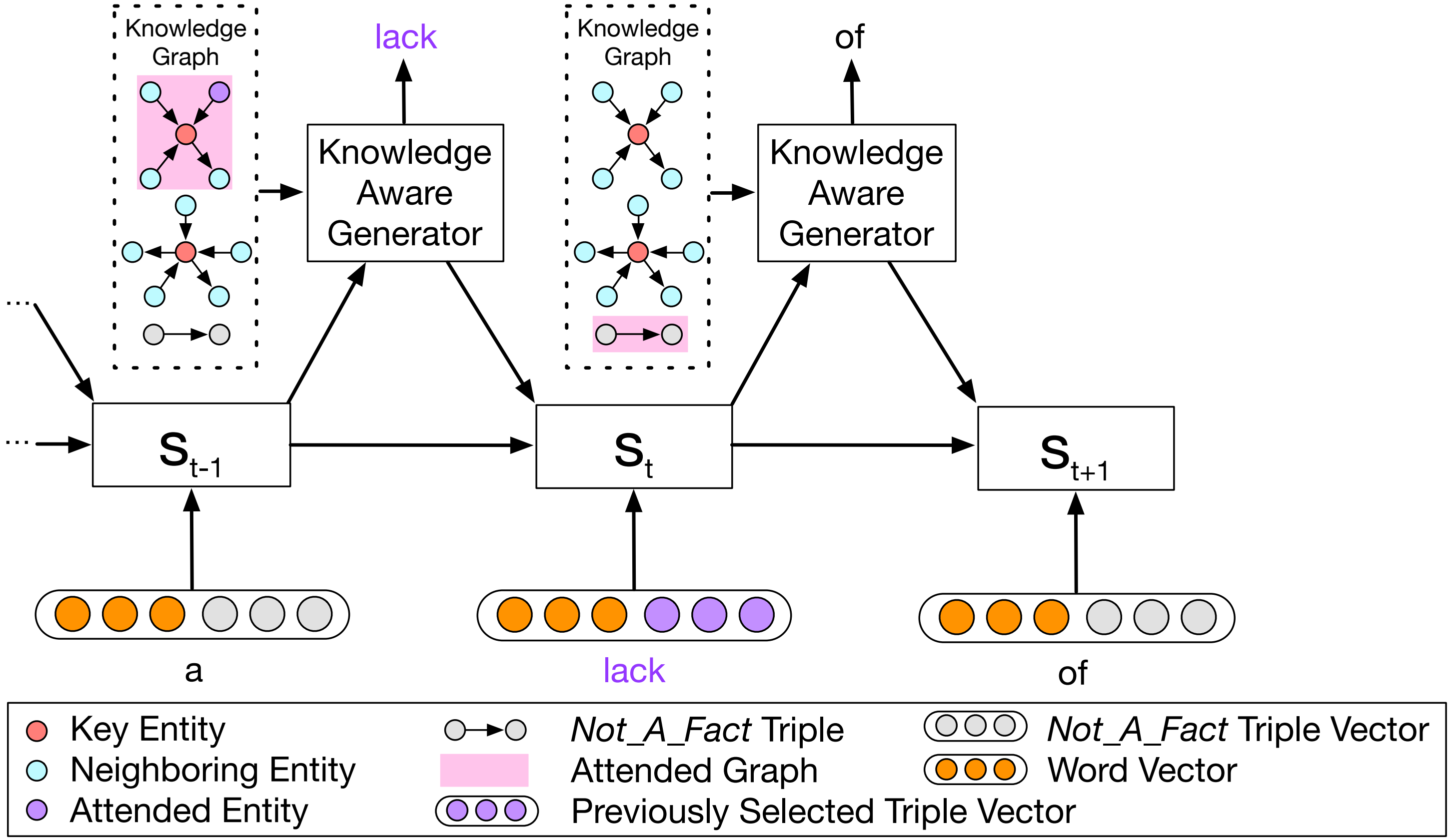
Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph



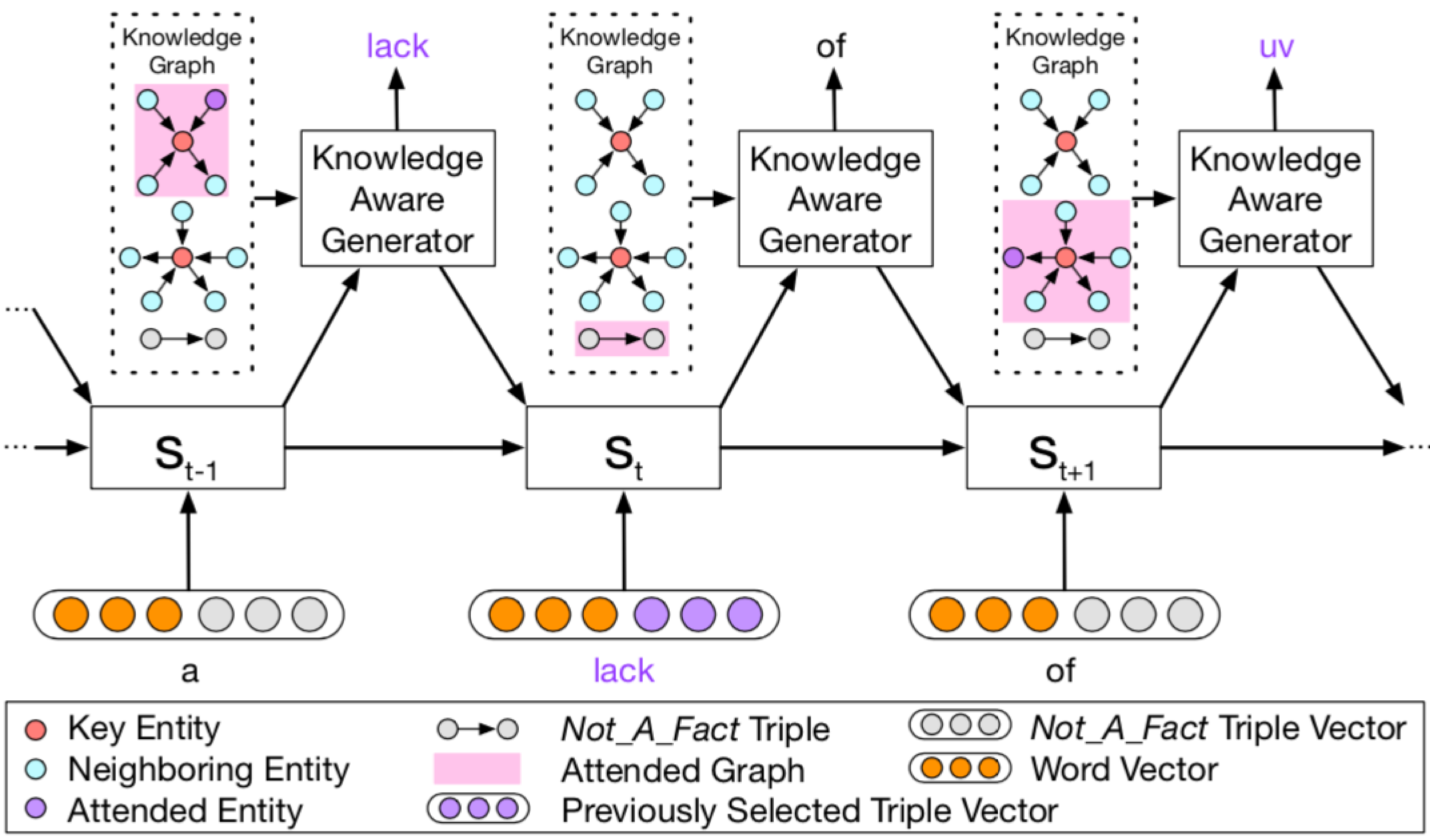
Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph



Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph



Commonsense Knowledge in Chatbots

- Dataset: filtered from 10M reddit single-round dialogs

Conversational Pairs		Commonsense KB	
Training	3,384,185	Entity	21,471
Validation	10,000	Relation	44
Test	20,000	Triple	120,850

Table 1: Statistics of the dataset and the knowledge base.

Commonsense Knowledge in Chatbots

Automatic evaluation

Model	Overall		High Freq.		Medium Freq.		Low Freq.		OOV	
	ppx.	ent.	ppx.	ent.	ppx.	ent.	ppx.	ent.	ppx.	ent.
Seq2Seq	47.02	0.717	42.41	0.713	47.25	0.740	48.61	0.721	49.96	0.669
MemNet	46.85	0.761	41.93	0.764	47.32	0.788	48.86	0.760	49.52	0.706
CopyNet	40.27	0.96	36.26	0.91	40.99	0.97	42.09	0.96	42.24	0.96
CCM	**39.18**	**1.180**	**35.36**	**1.156**	**39.64**	**1.191**	**40.67**	**1.196**	**40.87**	**1.162**

Manual evaluation (Sign-test, p-value<0.005)

Model	Overall		High Freq.		Medium Freq.		Low Freq.		OOV	
	app.	inf.	app.	inf.	app.	inf.	app.	inf.	app.	inf.
CCM vs. Seq2Seq	0.616	0.662	0.605	0.656	0.549	0.624	0.636	0.650	0.673	0.716
CCM vs. MemNet	0.602	0.647	0.593	0.656	0.566	0.640	0.622	0.635	0.626	0.657
CCM vs. CopyNet	0.600	0.640	0.606	0.669	0.586	0.619	0.610	0.633	0.596	0.640

Hao Zhou, Tom Yang, Minlie Huang, Haizhou Zhao, Jingfang Xu, Xiaoyan Zhu. Commonsense Knowledge Aware Conversation Generation with Graph Attention. **IJCAI-ECAI 2018**, Stockholm, Sweden.

Generation Examples 1

Post: He proposed March 5th. We will be married October 10th. So 7 months

Knowledge: (engaged, RelatedTo, married), (wedding, RelatedTo, married), (miss, DistinctFrom, married), (March, RelatedTo, months)

- **Seq2Seq**: I'm not sure how to answer that. I'm not sure how to make a date.
- **MemNet**: I'm assuming he's married to a gay marriage?
- **CopyNet**: I'm sorry, I'm not sure what you mean by that.
- **CCM**: I'm pretty sure he's going to be **engaged** in the next 6 **months**.

Generation Examples 2

Post: Totally thought it was going to be doug's grave.
Knowledge: (thought, RelatedTo, thinking), (doug, HasProperty, dead), (dead, RelatedTo, grave), (grave, RelatedTo, death)

- **Seq2Seq**: I was thinking the same thing.
- **MemNet**: I thought it was a dead end.
- **CopyNet**: I think it's a bit of a stretch.
- **CCM**: I thought it was doug's **death**. I was **thinking** the same thing.

Logic:
Story Ending Generation

Story Ending Generation

- Given a story context, conclude the story and complete the plot (**reasonable in logic and causality**)

Context: Today is Halloween .
Jack is so excited to go trick or treating tonight .
He is going to dress up like a monster .
The costume is real scary .

Ending : He hopes to get a lot of candy .

Story Ending Generation

Generating a good ending requires:

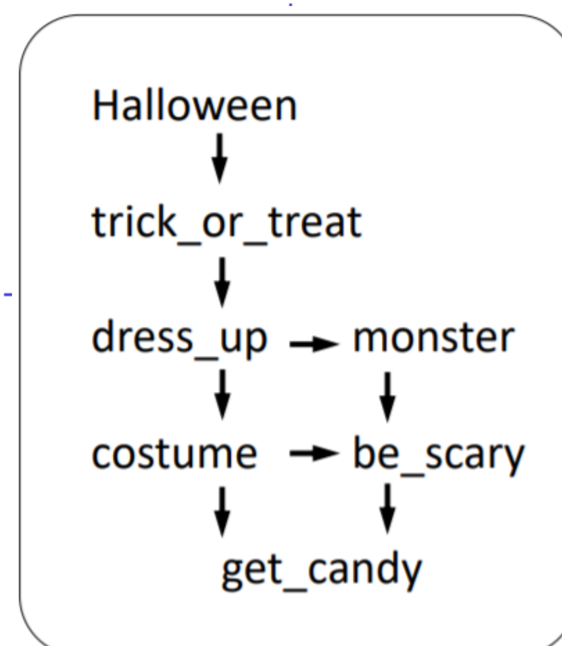
- Representing the **context clues** which contain key information for planning a reasonable ending
- Using **implicit knowledge** (e.g., commonsense knowledge) to facilitate understanding of the story and better predict what will happen next.

Logic: Story Ending Generation

Finding context clues: plan the order of events and entities.

Today is **Halloween** .
Jack is so excited to go **trick or treating** tonight .
He is going to **dress up** like a **monster** .
The **costume** is real **scary** .

He hopes to get a lot of **candy** .



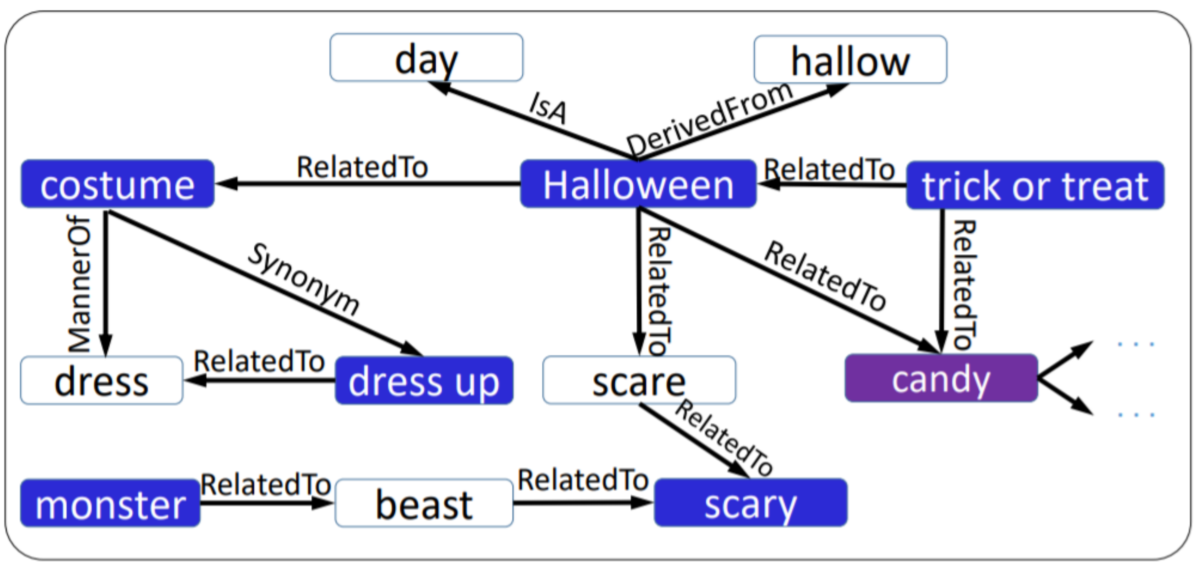
Jian Guan, Yansen Wang, Minlie Huang. **Story Ending Generation with Incremental Encoding and Commonsense Knowledge**. AAAI 2019

Logic: Story Ending Generation

Commonsense knowledge

Today is **Halloween** .
Jack is so excited to go **trick or treating** tonight .
He is going to **dress up** like a **monster** .
The **costume** is real **scary** .

He hopes to get a lot of **candy** .



Task Overview

- Given a story context consisting of a sentence sequence:

$$X = \{X_1, X_2, X_2, \dots, X_K\}, \text{ where } X_i = x_1^{(i)} x_2^{(i)} \dots x_{l_i}^{(i)}$$

- The model should generate a one-sentence ending:

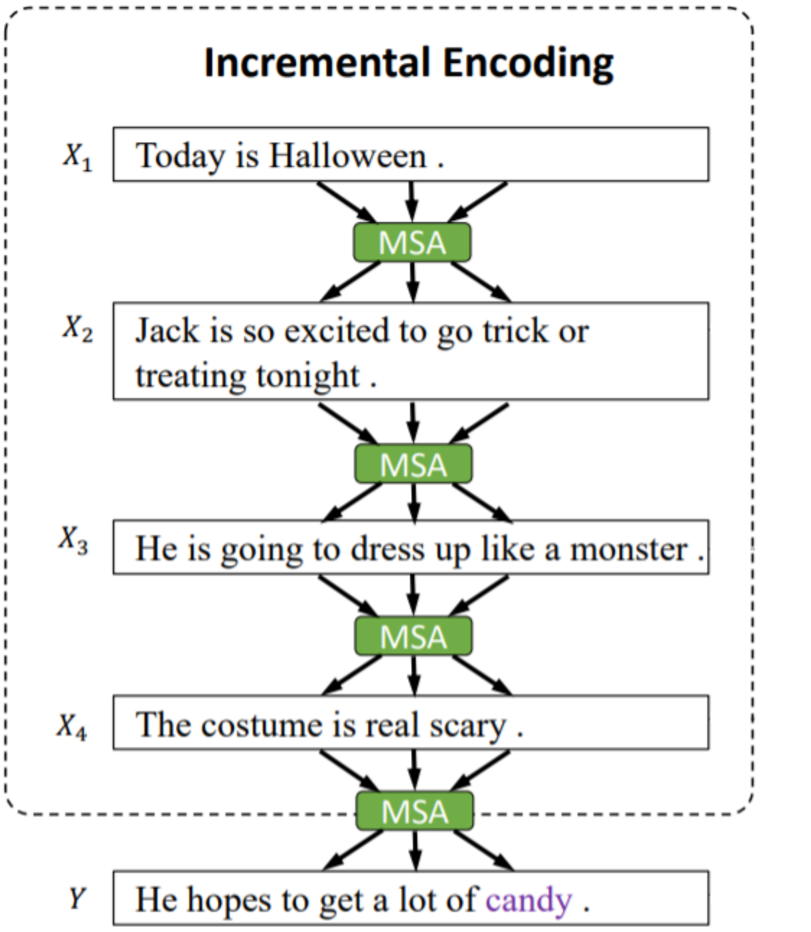
$$Y = y_1 y_2 \dots y_l$$

- Formally:

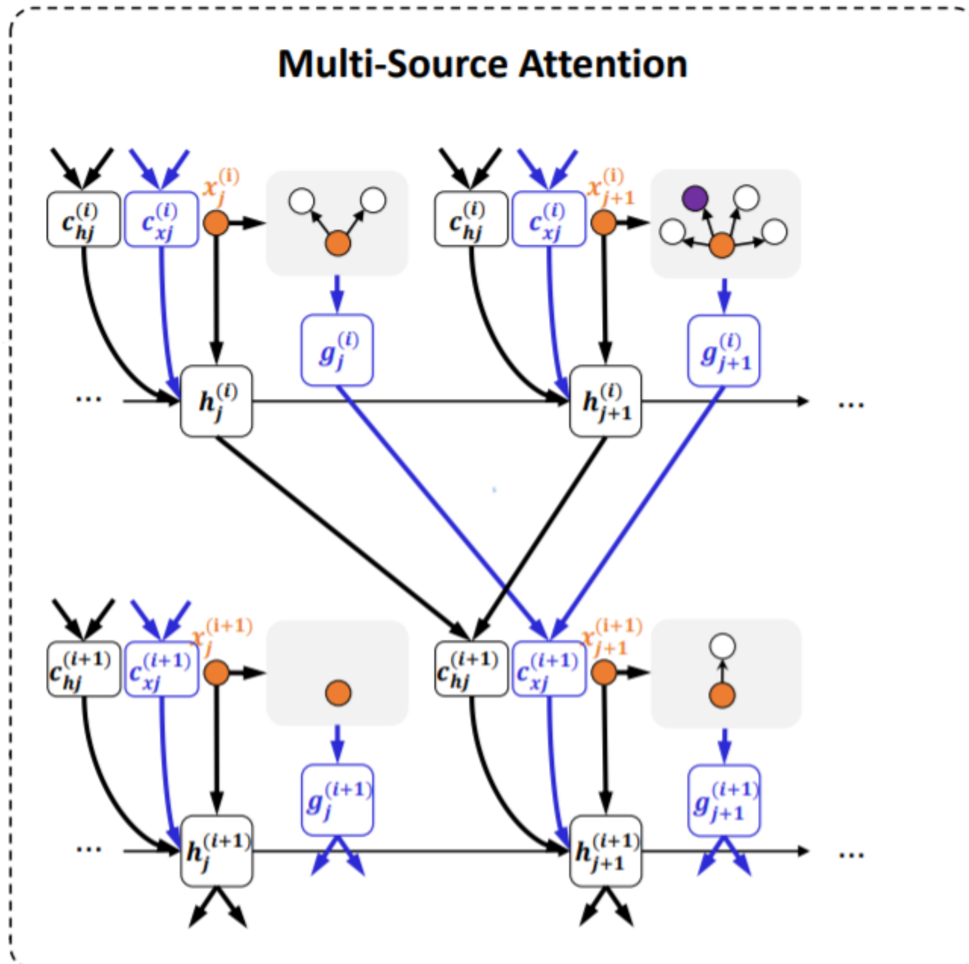
$$Y^* = \underset{Y}{argmax}\, \mathcal{P}(Y|X).$$

Logic: Story Ending Generation

Incremental Encoding



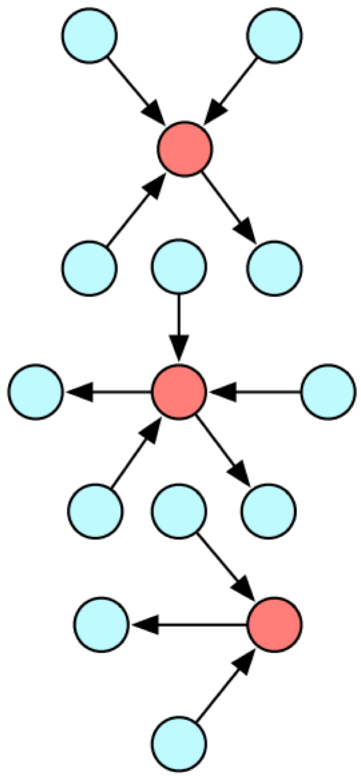
Multi-Source Attention

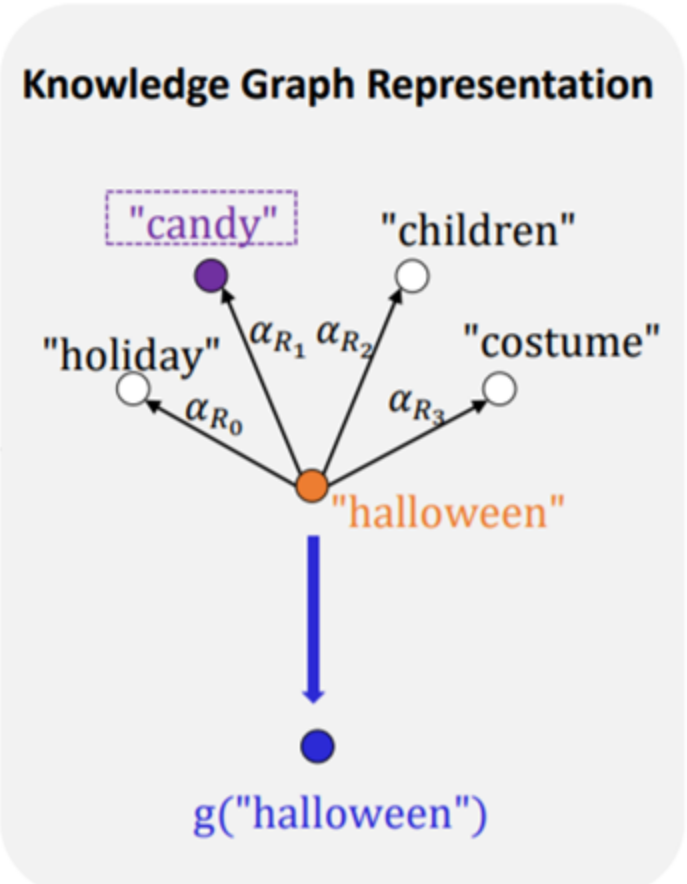


Logic: Story Ending Generation

Attention to the knowledge base: static graph attention

Graph attention





Model--- Encoder

Possible solutions for encoding:

- Concatenating the *K* sentences to a long sentence and encoding it with an LSTM
- Using a hierarchical LSTM with hierarchical attention (Yang et al. 2016)
- **Incremental Encoding (our proposal)**

Model --- Encoder

Incremental Encoding

- Effective to represent the context clues which may **capture the key logic information.**
- The current sentence X_i
- An **attentive read of the preceding sentence** X_{i-1}: $\boldsymbol{c}_{\boldsymbol{l}j}^{(i)}$

$$\mathbf{h}_j^{(i)} = \mathbf{LSTM}(\mathbf{h}_{j-1}^{(i)}, e(x_j^{(i)}), \mathbf{c}_{\mathbf{l}j}^{(i)}),\ i \geq 2.$$

- Story ending generation:

$$\mathbf{s}_t = \mathbf{LSTM}(\mathbf{s}_{t-1}, e(y_{t-1}), \mathbf{c}_{\mathbf{l}t}),$$

$$\mathcal{P}(y_t|y_{<t}, X) = \mathbf{softmax}(\mathbf{W}_0 \boldsymbol{s}_t + \mathbf{b}_0),$$

Model ---Encoder

Context vector

- Capture the relationship between words (or states) in the current sentence and those in the preceding sentence
- Encode implicit knowledge that is beyond the text
- Formally: $\mathbf{c}_{\mathbf{l}j}^{(i)} = \mathbf{W}_{\mathbf{l}}([\mathbf{c}_{\mathbf{h}j}^{(i)}; \mathbf{c}_{\mathbf{x}j}^{(i)}]) + \mathbf{b}_{\mathbf{l}},$
  - $\mathbf{c}_{\mathbf{h}j}^{(i)}$ is called **state context vector pointing to** X_{i-1}
  - $\mathbf{c}_{\mathbf{x}j}^{(i)}$ is called **knowledge context vector pointing to** X_{i-1}

Model --- Encoder

- **State context vector**

$$\mathbf{c}_{\mathbf{h}j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{h_k,j}^{(i)} \mathbf{h}_k^{(i-1)},$$

$$\alpha_{h_k,j}^{(i)} = \frac{e^{\beta_{h_k,j}^{(i)}}}{\sum_{m=1}^{l_{i-1}} e^{\beta_{h_m,j}^{(i)}}},$$

$$\beta_{h_k,j}^{(i)} = \mathbf{h}_{j-1}^{(i)\mathrm{T}} \mathbf{W_s} \mathbf{h}_k^{(i-1)},$$

- **Knowledge context vector**

$$\mathbf{c}_{\mathbf{x}j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{x_k,j}^{(i)} \mathbf{g}(x_k^{(i-1)}),$$

$$\alpha_{x_k,j}^{(i)} = \frac{e^{\beta_{x_k,j}^{(i)}}}{\sum_{m=1}^{l_{i-1}} e^{\beta_{x_m,j}^{(i)}}},$$

$$\beta_{x_k,j}^{(i)} = \mathbf{h}_{j-1}^{(i)\mathrm{T}} \mathbf{W_k} \mathbf{g}(x_k^{(i-1)}),$$

Model --- Knowledge

Knowledge graph retrieval

- **ConceptNet**: a commonsense semantic network
- Consists of triples $\boldsymbol{R} = (\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ meaning that head concept $\boldsymbol{h}$ has the relation $\boldsymbol{r}$ with tail concept $\boldsymbol{t}$
  - e.g. (*costume*, */R/MannerOf*, *dress*)
- Each word in a sentence is used as a query to **retrieve a one-hop graph** from ConceptNet.

Model --- Knowledge

- The knowledge graph for a word extends (encodes) its meaning by **representing the graph** from neighboring concepts and relations.
  - **Graph Attention** (Velikovi et al. 2018; Zhou et al. 2018)
  - **Contextual attention** (Mihaylov and Frank 2018)

Model --- Knowledge

- **Graph Attention**

$$\mathbf{g}(x) = \sum_{i=1}^{N_x} \alpha_{R_i} [\mathbf{h}_i; \mathbf{t}_i],$$

$$\alpha_{R_i} = \frac{e^{\beta_{R_i}}}{\sum_{j=1}^{N_x} e^{\beta_{R_j}}},$$

$$\beta_{R_i} = (\mathbf{W_r r}_i)^{\mathrm{T}} \, tanh(\mathbf{W_h h}_i + \mathbf{W_t t}_i),$$

- **Contextual Attention**

$$\mathbf{g}(x) = \sum_{i=1}^{N_x} \alpha_{R_i} \mathbf{M}_{R_i},$$

$$\mathbf{M}_{R_i} = BiGRU(\mathbf{h}_i, \mathbf{r}_i, \mathbf{t}_i),$$

$$\alpha_{R_i} = \frac{e^{\beta_{R_i}}}{\sum_{j=1}^{N_x} e^{\beta_{R_j}}},$$

$$\beta_{R_i} = \mathbf{h}_{(x)}^{\mathrm{T}} \mathbf{W_c M}_{R_i},$$

Model --- Knowledge

- Impose supervision on both the encoding network and decoding network

$$\Phi = \Phi_{en} + \Phi_{de}$$

$$\Phi_{en} = \sum_{i=2}^{K} \sum_{j=1}^{l_i} -\log \mathcal{P}(x_j^{(i)} = \widetilde{x}_j^{(i)} | x_{<j}^{(i)}, X_{<i}),$$

$$\Phi_{de} = \sum_{t} -\log \mathcal{P}(y_t = \tilde{y}_t | y_{<t}, X),$$

Datasets

- ROCStories corpus
  - Each story consists of **five sentences**, our task is to generate the ending given the first 4 sentence
  - 90,000 for training and 8,162 for evaluation
  - Average length of $X_1/X_2/X_3/X_4/Y$ is 8.9/9.9/10.1/10.0/10.5

- ConceptNet
  - Only retrieve the relations whose head entity and tail entity are **noun or verb**, meanwhile **both occurring in SCT**.
  - Retain at most 10 triples if there are too many for a word.
  - Average number of triples for each query word is 3.4

Metrics

- **Automatic Evaluation**
  - Perplexity, BLEU-1 and BLEU-2
    - How well a model fits the data

- **Manual Evaluation**
  - Grammar (Gram.)
    - Score 2 : without any grammar errors
    - Score 1 : with a few errors but still understandable
    - Score 0 : with severe errors and incomprehensible
  - Logicality (Logic.)
    - Score 2 : totally reasonable endings
    - Score 1 : relevant but with some discrepancy
    - Score 2 : totally incompatible endings

Results

Model	PPL	BLEU-1	BLEU-2	Gram.	Logic.
Seq2Seq	18.97	0.1864	0.0090	1.74	0.70
HLSTM	17.26	0.2459	0.0242	1.57	0.84
HLSTM+Copy	19.93	0.2469	0.0248	1.66	0.90
HLSTM+MSA(GA)	15.75	0.2588	0.0253	1.70	1.06
HLSTM+MSA(CA)	12.53	0.2514	0.0271	1.72	1.02
IE (ours)	11.04	0.2514	0.0263	**1.84**	1.10
IE+MSA(GA) (ours)	9.72	0.2566	0.0284	1.68	**1.26**
IE+MSA(CA) (ours)	**8.79**	**0.2682**	**0.0327**	1.66	1.24

Table 1: Automatic and manual evaluation results.

Examples

Context:	Martha is **cooking** a special **meal** for her family. She **wants everything to be just right** for when they eat. Martha **perfects everything** and puts her **dinner** into the **oven**. Martha goes to **lay down** for a quick **nap**.
Golden Ending:	She **oversleeps** and runs into the **kitchen** to take out her **burnt dinner**.
Seq2Seq:	She was so happy to have a ***new cake***.
HLSTM:	Her family ***and her family*** are very happy with her **food**.
HLSTM+Copy:	**Martha** is happy to be able to ***eat her family***.
HLSTM+GA:	She is happy to be able to **cook her dinner**.
HLSTM+CA:	She is very happy that she has made a new **cook** .
IE:	She is very happy with her **family**.
IE+GA:	When she gets back to the **kitchen**, she sees a **burning light** on the **stove**.
IE+CA:	She realizes the **food** and is happy she was ready to **cook** .

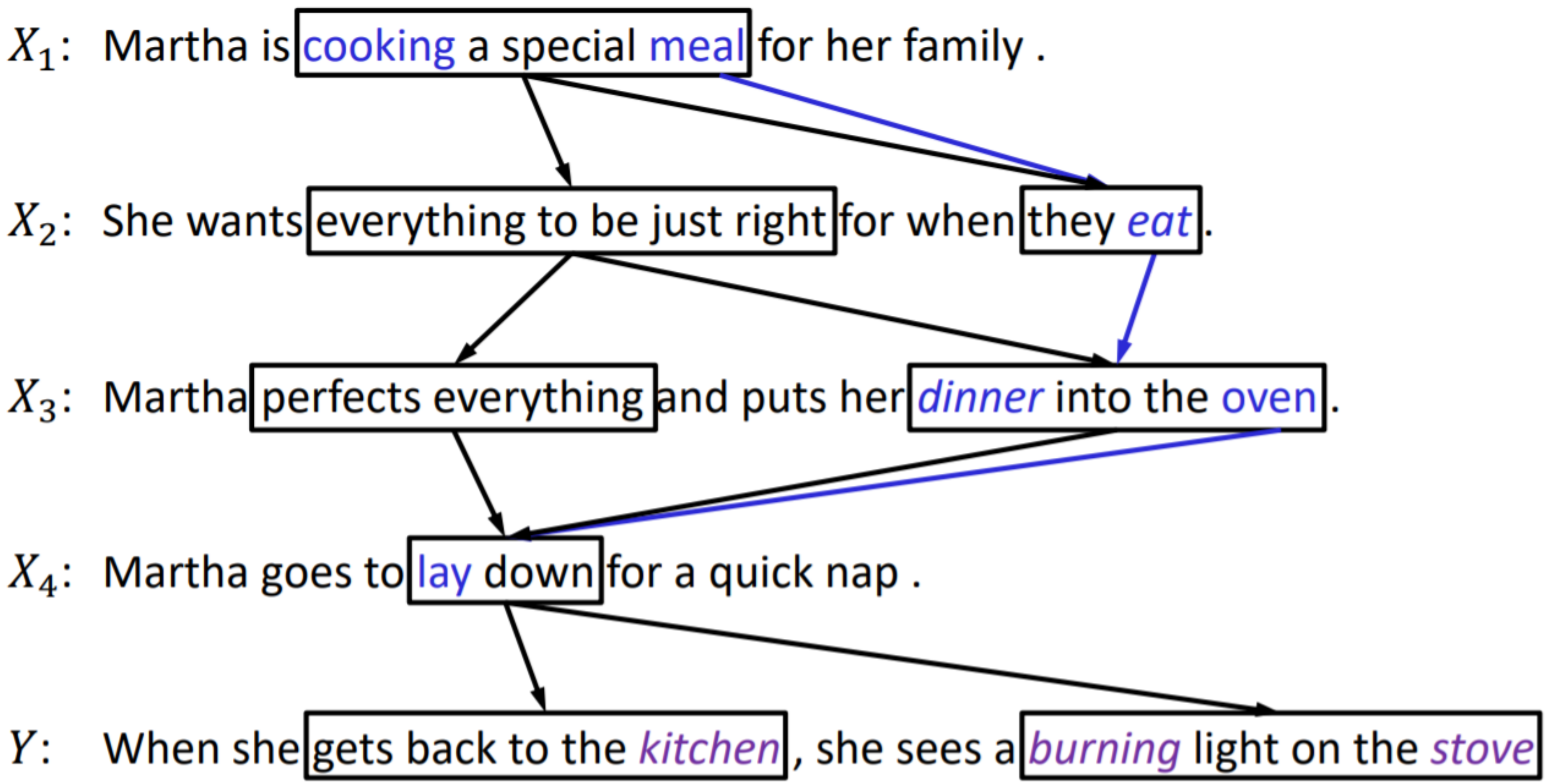
Generation Examples

Story 1:
Context:
Taj has never drank an espresso drink.
He ordered one while out with his friends.
The shot of espresso tasted terrible to him.
Taj found that he couldn't stop talking or moving.
Generated Ending:
He decided to never drink again.

Story 2:
Context:
Martha is cooking a special meal for her family.
She wants everything to be just right for when they eat.
Martha perfects everything and puts her dinner into the oven.
Martha goes to lay down for a quick nap.
Generated Ending:
When she gets back to the kitchen, she sees a burning light on the stove.

“Logic Chains”: Contextual Clue

Building context clues incrementally



Controllable Language Generation

- Three **fundamental problems** in current neural language generation models
  - **Semantics**
  - **Consistency** (long text generation)
  - **Logic** (reasonable and making sense)
- Long text generation: **planning**

Thanks for Your Attention

- http://coai.cs.tsinghua.edu.cn/ds/ 对话系统技术平台
- Acknowledgements
  - Prof Xiaoyan Zhu, Tsinghua colleagues, collaborators
  - Our students
- Contact:
  - Minlie Huang, Tsinghua University
  - aihuang@tsinghua.edu.cn
  - http://coai.cs.tsinghua.edu.cn/hml