Knowledge-Aware Dialogue System

Xingwu Lu

Outline

- Augmenting End-to-End Dialogue Systems with Commonsense Knowledge AAAI-2018
- Commonsense Knowledge Aware Conversation Generation with Graph Attention IJCAI-2018

Background

- Domain
  - Task-oriented dialogue
  - Chatbots
- Methods
  - Retrieval-based Methods
  - Generation-based Methods
- Scenarios
  - Single-Turn dialogue
  - Multi-turn dialogue

Background

- Semantic information
- External knowledge
    - Structured Knowledge
    - Unstructured Texts

Augmenting End-to-End Dialogue Systems with Commonsense Knowledge

AAAI-2018

Tsinghua University

Tom Young, Erik Cambria, Iti Chaturvedi

Hao Zhou, Subham Biswas, Minlie Huang

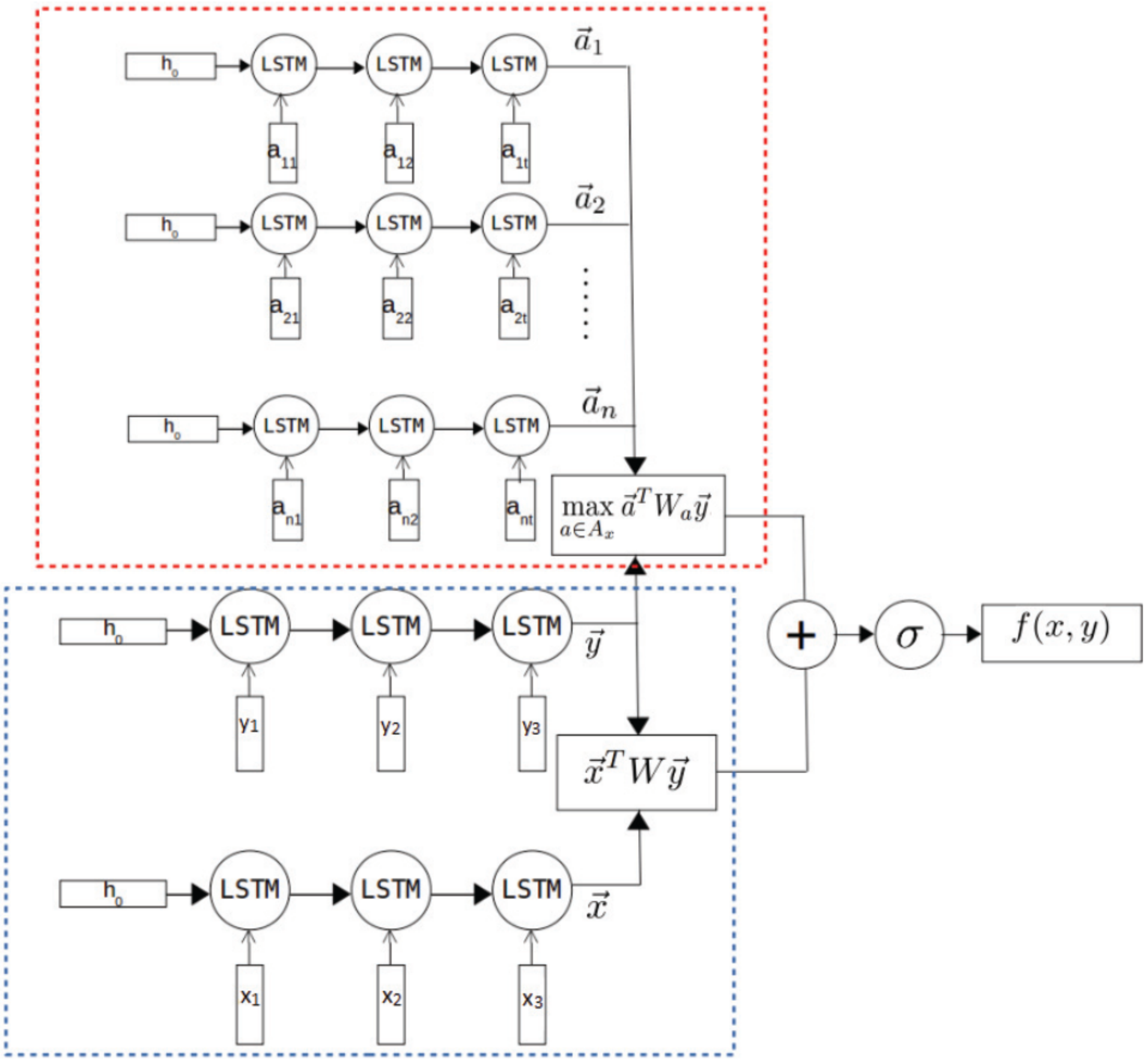
Motivation

验证在对话中引入Commonsense Knowledge的有效性。

在基于检索的对话系统中引入：

- 容易验证
- 少量训练数据即可

Model



h_0
LSTM
a_{11}
a_{12}
a_{1t}
$\vec{a}_1$
a_{21}
a_{22}
a_{2t}
$\vec{a}_2$
a_{n1}
a_{n2}
a_{nt}
$\vec{a}_n$
$\max_{a \in A_x} \vec{a}^T W_a \vec{y}$
y_1
y_2
y_3
$\vec{y}$
x_1
x_2
x_3
$\vec{x}$
$\vec{x}^T W \vec{y}$
+
σ
$f(x, y)$

Dual-LSTM

以Dual-LSTM为基本架构，也就是最早的Ubuntu数据集上的模型为基本架构：

$$f(x, y) = \sigma(x^T W y)$$

一个双仿射来计算上下文和回复之间的匹配得分

Commonsense Knowledge Retrieval

- 定义：对于每一个concept，与他相关的关系三元组为assertion$< c_1, r, c_2 >$
- step1：预先构建好一个字典，key为每一个concept，值为与concept相关的所有assertions
- step2：对于每一个上下文，检索与之匹配的少于五个的concept的所有assertions

Tri-LSTM Encoder

- 将assertion转换成序列：$[c_{11}, c_{12}, c_{13}, \ldots, r, c_{21}, c_{22}, c_{23}, \ldots]$
- 用额外的LSTM对assertions进行建模，分别与候选回复做匹配，对得到的所有的匹配得分做最大化
- 最后将最大的匹配得分与Dual-LSTM的匹配得分相加

$$f(x, y) = \sigma(X^T W Y + m(A_x, y))$$

$$m(A_x, y) = a^T W_a T$$

Experiments

DataSet

Twitter Dialogue Dataset

1.4M Twitter <message, response> pairs

ConceptNet

- 1.4M concepts
- 4.3 assertions

Results

Recall@k	TF-IDF	Word Embeddings*	Memory Networks*	Dual-LSTM	Tri-LSTM*	Human
Recall@1	32.6%	73.5%	72.1%	73.6%	**77.5%**	87.0%
Recall@2	47.3%	84.0%	83.6%	85.6%	**88.0%**	-
Recall@5	68.0%	95.5%	94.2%	95.9%	**96.6%**	-

Baseline

- Supervised Word Embeddings：不用LSTM直接用Embedding
- Memory Networks：用断言构作为memory

Conclusion

缺点：

- 模型上的创新性，用的别人的模型baseline也很老
- 性能也不咋的
- 实验量也不够大

亮点：

在对话上初步尝试了在开放领域融入通用知识

Commonsense Knowledge Aware Conversation Generation with Graph Attention

IJCAI-2018

Tsinghua University

Hao Zhou, Tom Young, Minlie Huang,

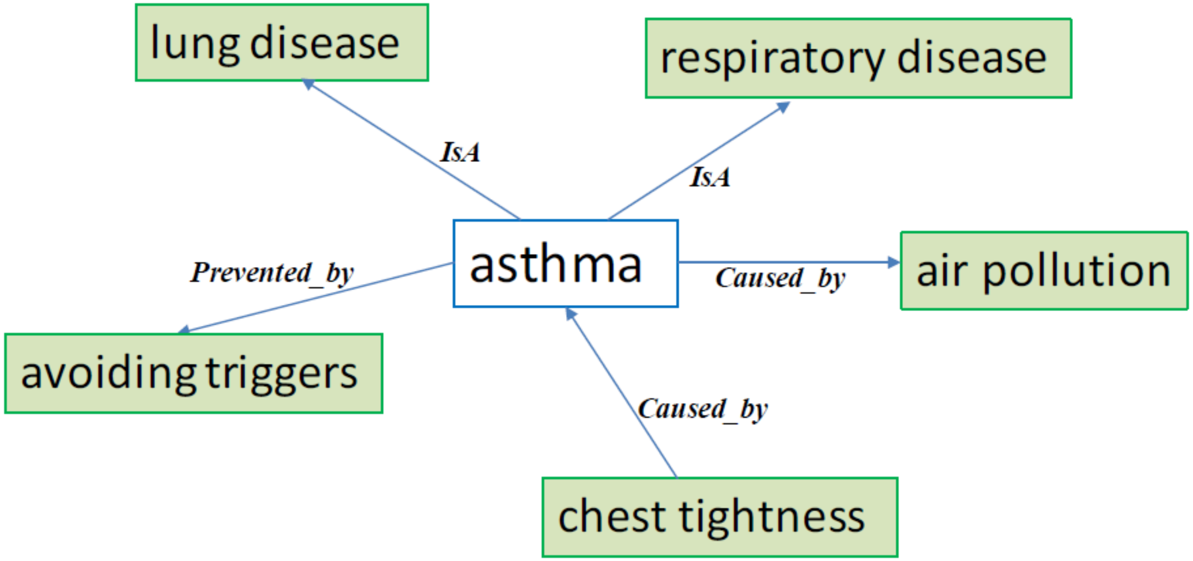
Haizhou Zhao, Jingfang Xu and Xiaoyan Zhu

Motivation

previous:

- highly dependent on the quality of unstructured texts
- limited by the small-scale, domain-specific knowledge
- make use of knowledge triples (entities) separately and independently, instead of treating knowledge triples as a whole in a graph

Motivation



lung disease
respiratory disease
IsA
IsA
asthma
Prevented_by
Caused_by
air pollution
avoiding triggers
Caused_by
chest tightness

Motivation

- large-scale commonsense knowledge (first attempt)
  - language understanding
  - language generation
- our model treats each knowledge graph as a whole, which encodes more structured, connected semantic information in the graphs

Model

- augments the semantic information of the post (Encoder)
- the model attentively reads the retrieved knowledge graphs and the knowledge triples (Decoder)

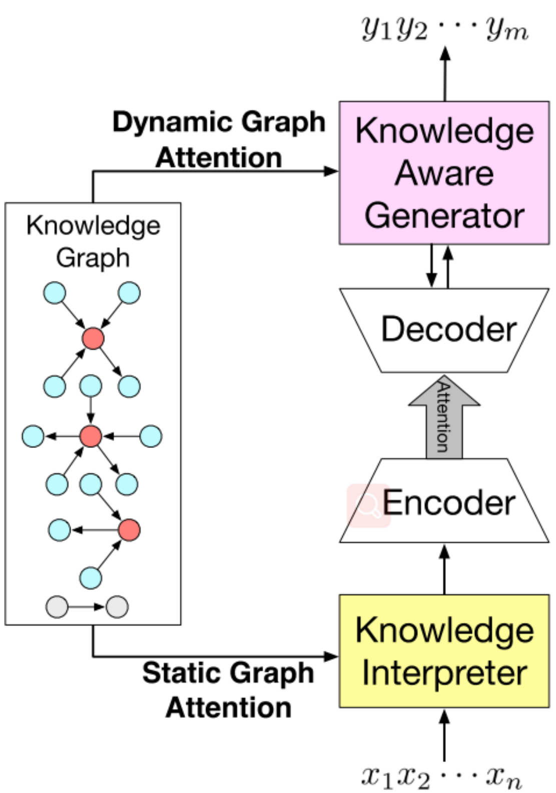
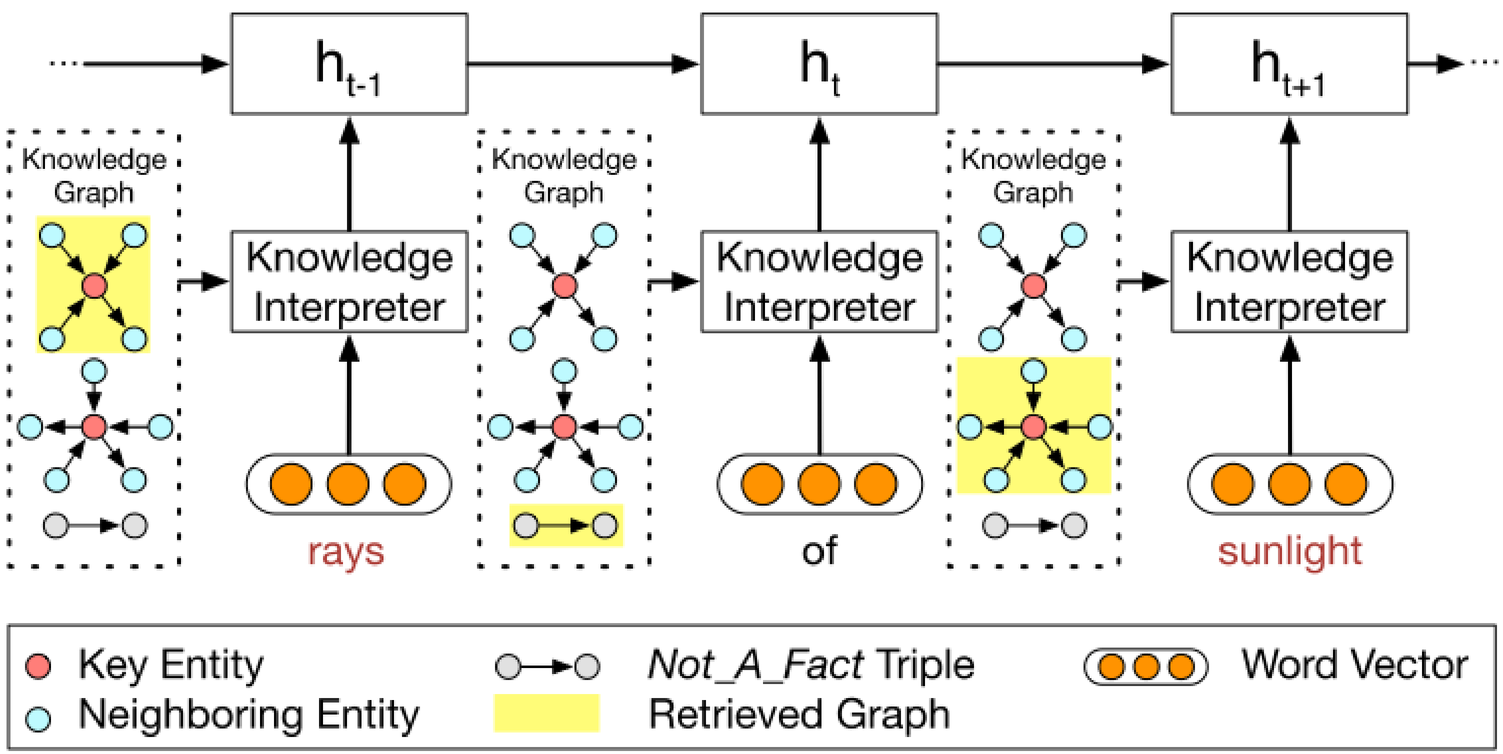


Figure 2: Overview of CCM.

Task Definition

- Each word corresponds to a graph
- Each graph consists of a set of triples
- Each triple (head entity, relation, tail entity)
- TransE to represent the entities and relations: adopt a MLP to bridge the representation gap between knowledge base and unstructured conversational texts, a knowledge triple τ is represented by $k = (h, r, t) = MLP(TransE(h, r, t))$

Knowledge Interpreter



Static Graph Attention

$$\boldsymbol{g}_i = \sum_{n=1}^{N_{g_i}} \alpha_n^s [\boldsymbol{h}_n; \boldsymbol{t}_n], \quad (4)$$

$$\alpha_n^s = \frac{\exp(\beta_n^s)}{\sum_{j=1}^{N_{g_i}} \exp(\beta_j^s)}, \quad (5)$$

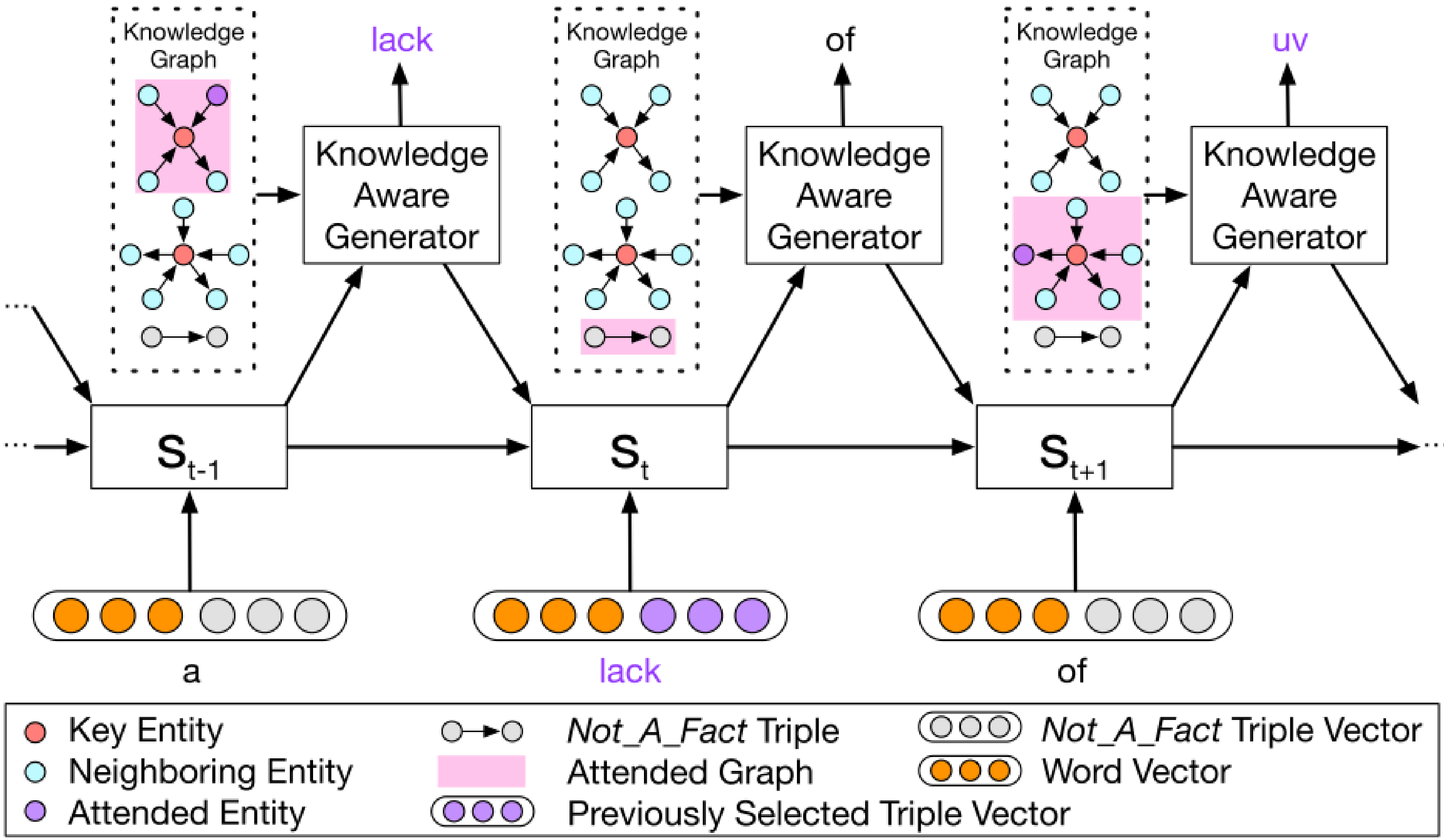
$$\beta_n^s = (\mathbf{W_r} \boldsymbol{r}_n)^\top \tanh(\mathbf{W_h} \boldsymbol{h}_n + \mathbf{W_t} \boldsymbol{t}_n), \quad (6)$$

where $(h_n, r_n, t_n) = k_n$

The attention weight measures the association of a relation r_n to a head entity h_n and a tail entity t_n.

concatenated vector $e(x_t)$ = [$w(x_t)$; g_i] is obtained and fed to the GRU cell of the encoder

Knowledge Aware Generator



$$
\begin{aligned}
\boldsymbol{s}_{t+1} &= \mathbf{GRU}(\boldsymbol{s}_t, [\boldsymbol{c}_t; \boldsymbol{c}_t^g; \boldsymbol{c}_t^k; \boldsymbol{e}(y_t)]), && (7)\\
\boldsymbol{e}(y_t) &= [\boldsymbol{w}(y_t); \boldsymbol{k}_j], && (8)
\end{aligned}
$$

- c_t is the context vector
- $e(y_t)$ is the concatenation of the word vector $w(y_t)$ and the previous knowledge triple vector k_j from which the previous word (y_t) is selected
- c_t^g is context vectors attended on knowledge graph vectors $\{g_1, g_2, ..., g_{N_G}\}$
- c_t^k is context vectors attended on knowledge triple vectors $\{K(g_1), K(g_2), ..., K(g_{N_G})\}$

Dynamic Graph Attention

a hierarchical, top-down process:

第一层attention：对整个图的表示做attention，整个图的表示是通过静态attention获取的

$$
\begin{aligned}
\boldsymbol{c}_t^g &= \sum_{i=1}^{N_G} \alpha_{ti}^g \boldsymbol{g}_i, && (9)\\
\alpha_{ti}^g &= \frac{\exp(\beta_{ti}^g)}{\sum_{j=1}^{N_G} \exp(\beta_{tj}^g)}, && (10)\\
\beta_{ti}^g &= \boldsymbol{V}_b^\top \tanh(\mathbf{W_b} \boldsymbol{s}_t + \mathbf{U_b} \boldsymbol{g}_i), && (11)
\end{aligned}
$$

第二层attention：对每个图中的三元组做attention

$$\boldsymbol{c}_t^k = \sum_{i=1}^{N_G} \sum_{j=1}^{N_{g_i}} \alpha_{ti}^g \alpha_{tj}^k \boldsymbol{k}_j, \quad (12)$$

$$\alpha_{tj}^k = \frac{\exp(\beta_{tj}^k)}{\sum_{n=1}^{N_{g_i}} \exp(\beta_{tn}^k)}, \quad (13)$$

$$\beta_{tj}^k = \boldsymbol{k}_j^\top \mathbf{W_c} \boldsymbol{s}_t, \quad (14)$$

Finally selects a generic word or an entity word with the following distributions:

$$\begin{aligned}
\boldsymbol{a}_t &= [\boldsymbol{s}_t; \boldsymbol{c}_t; \boldsymbol{c}_t^g; \boldsymbol{c}_t^k], && (15)\\
\gamma_t &= \mathrm{sigmoid}({\mathbf{V_o}}^\top \boldsymbol{a}_t), && (16)\\
P_c(y_t = w_c) &= \mathrm{softmax}(\mathbf{W_o} \boldsymbol{a}_t), && (17)\\
P_e(y_t = w_e) &= \alpha_{ti}^g \alpha_{tj}^k, && (18)\\
y_t \sim \boldsymbol{o}_t = P(y_t) &= \begin{bmatrix} (1-\gamma_t) P_g(y_t = w_c) \\ \gamma_t P_e(y_t = w_e) \end{bmatrix}, && (19)
\end{aligned}$$

Note: Entity words are taken from the neighboring entities of the knowledge triples. 有点类似于 Pointer-Generator Net

Loss Function

$$L(\theta) = -\sum_{t=1}^{m} \boldsymbol{p}_t \log(\boldsymbol{o}_t) - \sum_{t=1}^{m} (q_t \log(\gamma_t) + (1-q_t)\log(1-\gamma_t)), \tag{20}$$

Additionally, we apply supervised signals on the knowledge aware generator layer to teacher-force the selection of an entity or a generic word

Experiments

Dataset

- Commonsense Knowledge Base (ConceptNet)
    - removed triples containing multi-word entities
    - 120,850 triples
    - 21,471 entities
    - 44 relations

Dataset

- Commonsense Conversation Dataset (reddit single-round dialogs)
  connected by any triple (one entity appears in the post and the other in the response)
  - sampled 10,000 pairs for validation
  - constructed four test sets:
    - high- frequency pairs
    - medium-frequency pairs
    - low-frequency pairs
    - OOV pairs

Dataset

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.

Baselines

- seq2seq model (Seq2Seq)
- A knowledge-grounded model
- A copy network

knowledge-grounded model 是用的三元组的Embedding作为memory的单元

copy network是拷贝实体还是生成词

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**

Table 2: Automatic evaluation with *perplexity* (ppx.), and *entity score* (ent.).

Manual Evaluation

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

Table 3: Manual evaluation with *appropriateness* (app.), and *informativeness* (inf.). The score is the percentage that CCM wins its competitor after removing "Tie" pairs. CCM is significantly better (sign test, p-value < 0.005) than all the baselines on all the test sets.

Case Study

Post	Why are you so **breakable**?
Knowledge	(**glass**, RelatedTo, **breakable**), (**brittle**, RelatedTo, **breakable**), (**rule**, RelatedTo, **breakable**)
Seq2Seq	I'm not a **OOV** , I'm just a **OOV**.
MemNet	I'm not **OOV**. I'm just a really nice person.
CopyNet	I'm not. I'm just a lurker.
CCM	Because I'm a **brittle** man .

Table 4: Sample responses generated by all the models.