Conversations are not Flat

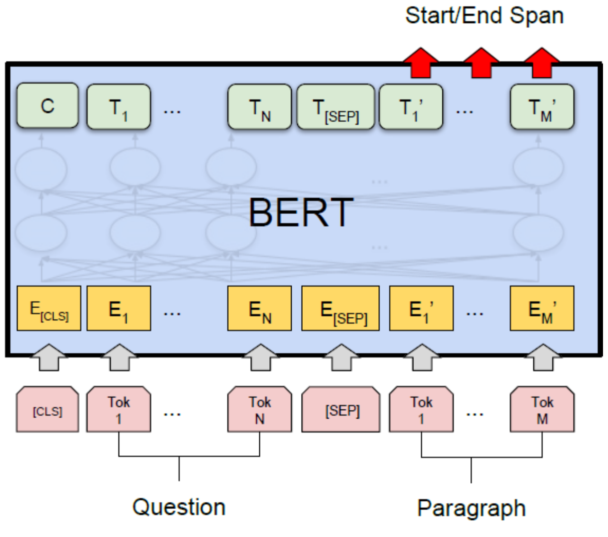
毕冠群

2021.6.18

对话数据的特征

- 对话上下文之间存在多轮依赖关系
- 对话必须是语句之间连贯的，并在多轮上达到交际的目的。
- 对话在本质上是互动的，说话者轮流进行发言，互相之间有反馈。
- 对话类文本包括了说话人（参与者）。存在意图、人格、情感等不同特点。更好的建模说话人可以帮助模型理解对话类文本。
- 主题漂移：在一段对话中同时存在多个对话主题
- ……

序列形式建模是最常见的形式，也就是像对待普通文本一样，将上下文语句串接起来，

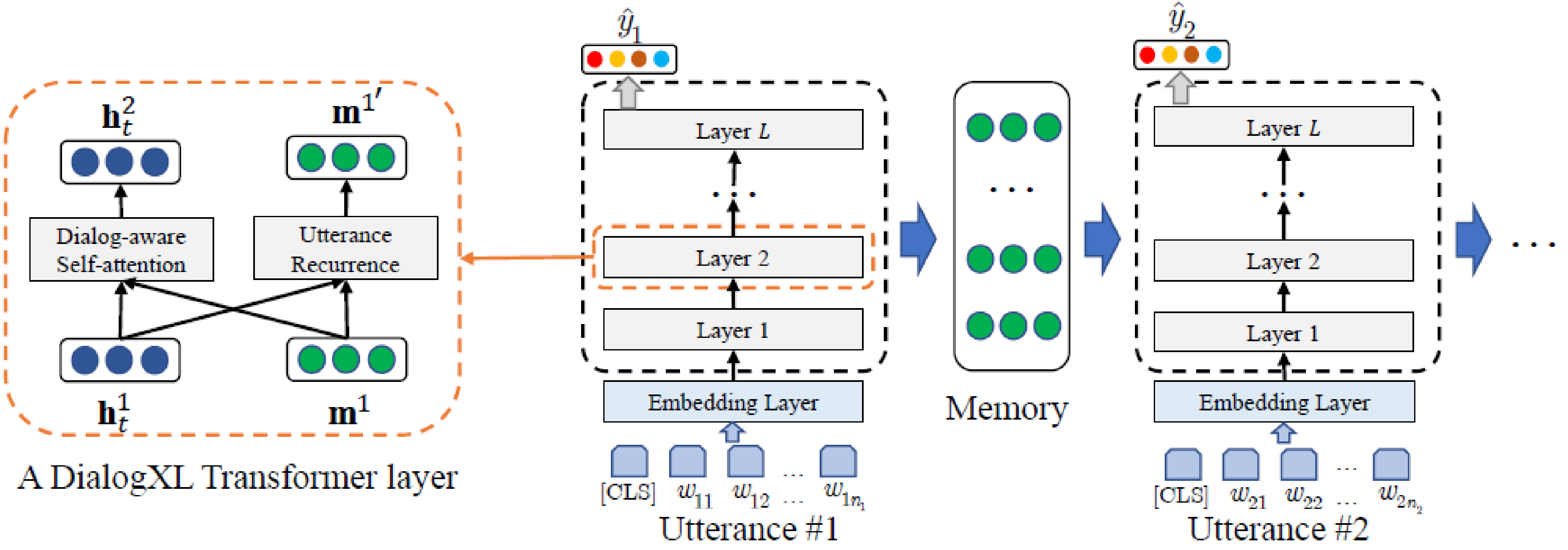


(c) Question Answering Tasks:

DialogXL

- 将预训练语言模型应用于会话中的情感识别
  - 如何用数百个单词编码一个长的历史上下文
  - 如何对不同方的说话人内部和说话人之间的依赖性进行建模。
- Contributions
  - 第一个专为对话情感识别设计的预训练语言模型
  - 使用节省内存的utterance recurrence来代替XLNet的segment recurrence。DialogXL最多存储1000个对话的历史单词，这比普通的XLNet模型更强大。
  - 与最初的自注意力仅计算单词之间的注意权重不同，模型的dialog-aware self-attention可以通过不同的接收字段和参与方角色进行计算，从而捕获有用的intra- and inter的依存关系。

DialogXL



- Model Input
  - 每个时间步输入query utterance

$$x_t = \{[\text{CLS}], w_{t1}, w_{t2}, ..., w_{tn_t}\}.$$

Utterance Recurrence

- memory就像堆栈一样。 每当为新的话语生成一组新的隐藏状态时，它们就会与当前memory连接在一起：
- 为了防止将噪声引入内存，仅存储utterance tokens的隐状态，而忽略[CLS]和padding
- 对第t条话语，在每个Transformer层l，新memory更新为：

$$\mathbf{m}^{l'} = \mathbf{m}^{l} \parallel \mathbf{h}^{l}_{t,1:1+n_t}$$

- 仅使用utterance tokens的隐藏状态更新memory使memory更紧凑，因为填充引入的噪声已消除，释放了更多空间来缓存较长的上下文。

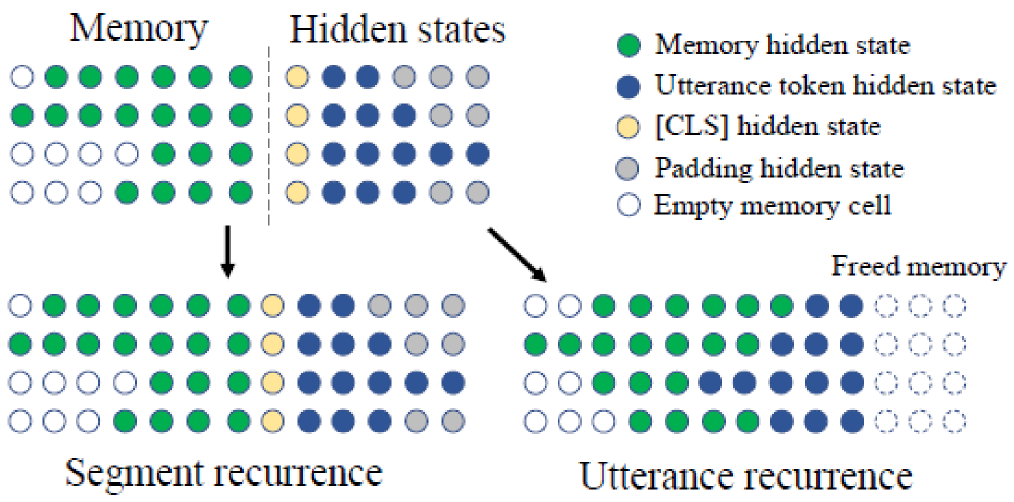


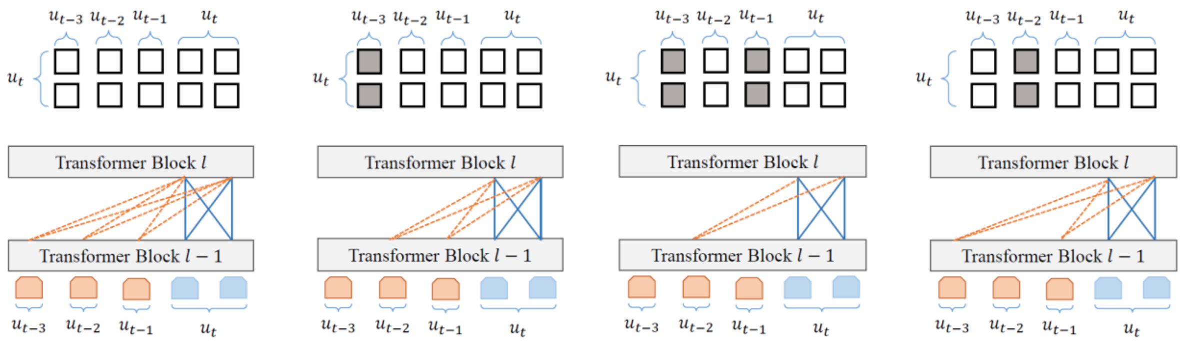
Figure 2: Illustration of the memory update strategies by utterance recurrence and segment recurrence. The batch size is 4, with each row corresponding to a conversation.

Dialog-Aware Self Attention

- 多方对话
- 新的self-atteniton包括四种类型：
  - 针对不同大小的感受野：
    - Global Self-Attention、Local Self-Attention
  - 针对speaker之间和之内的依赖：
    - Speaker Self-Attention、Listener Self-Attention
- 巧妙地改变自我注意的mask策略来实现，而无需添加任何额外的嵌入或参数

$$
\begin{aligned}
\widetilde{\mathbf{h}}_t^{l-1} &= \mathbf{m}^{l-1} \| \mathbf{h}_t^{l-1} \\
\mathbf{q}_t^l,\ \mathbf{k}_t^l,\ \mathbf{v}_t^l &= \mathbf{h}_t^{l-1}\mathbf{W}_q^{l\top},\ \widetilde{\mathbf{h}}_t^{l-1}\mathbf{W}_k^{l\top},\ \widetilde{\mathbf{h}}_t^{l-1}\mathbf{W}_v^{l\top} \\
\mathbf{a}_t^l &= \mathrm{RelPosAttn}(\mathbf{q}_t^l, \mathbf{k}_t^l) \\
\widetilde{\mathbf{a}}_t^l &= \mathbf{a}_t^l - \mathbf{s} \\
\mathbf{o}_t^l &= \mathrm{softmax}(\widetilde{\mathbf{a}}_t^l)\mathbf{v}_t^l
\end{aligned}
$$

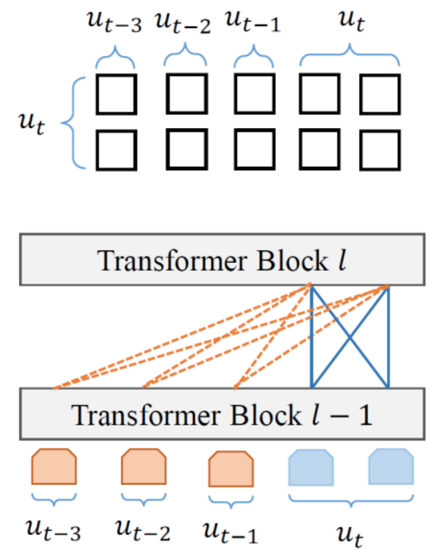
$$
\mathbf{o}_t^l = f(\mathbf{m}^{l-1}, \mathbf{h}_t^{l-1}, \mathbf{s})
$$



Dialog-Aware Self Attention

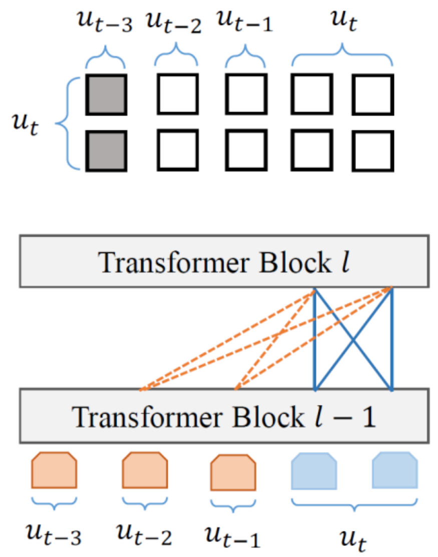
- Global Self-Attention
  - 与普通的自注意力相同，将所有历史背景和查询话语作为接收字段。
  - 查询话语会注意整个上下文，使模型可以处理以前很遥远的话语
  - 没有进行mask

$$\mathbf{s}_{ij}^{global} = 0$$



- Local Self-Attention
  - 直觉上说话者的情绪主要受最近话语的影响，局部自注意力只看最近的ω句历史话语
  - mask了查询话语与接收字段之外的历史话语之间的注意：

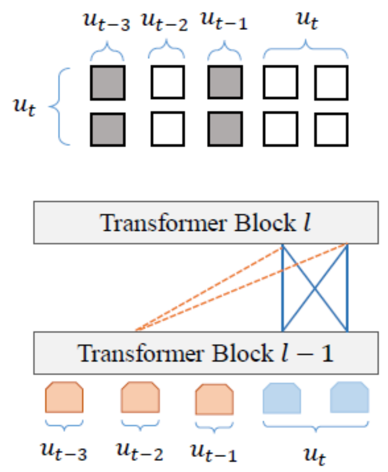
$$\mathbf{s}_{ij}^{local} = \begin{cases} +\infty, & j \notin \mathrm{Idx}(\{u_{t-\omega}, u_{t-\omega+1}, \ldots, u_{t-1}, u_t\}) \\ 0, & \text{Otherwise} \end{cases}$$



Dialog-Aware Self Attention

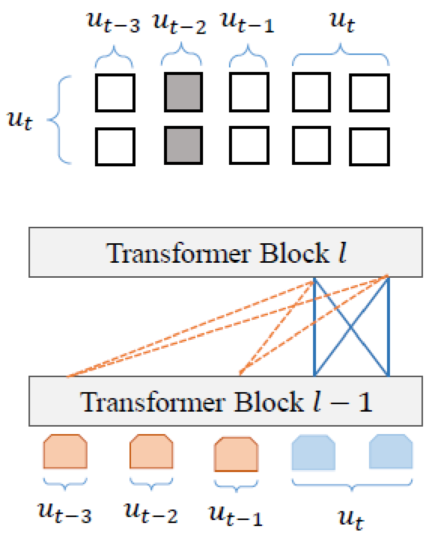
- Speaker Self-Attention
  - 仅考虑当前演讲者所说的历史信息,旨在通过识别说话者历史话语中的情感线索来模拟说话者内部的依存关系。
  - mask了查询话语与其他说话人的话语之间的注意:

$$\mathbf{s}_{ij}^{speaker} = \begin{cases} +\infty, & j \in \mathrm{Idx}(\{u \mid p(u) \neq p(u_t)\}) \\ 0, & \text{Otherwise} \end{cases}$$



- Listener Self-Attention
  - 仅考虑其他说话者所说的历史话语，旨在建模发言人之间的依存关系，这意味着当前发言人的情绪可能是受到其他说话者的话语的影响
  - mask了查询话语与当前讲话者的话语之间的注意:

$$\mathbf{s}_{ij}^{listener} = \begin{cases} +\infty, & j \in \mathrm{Idx}(\{u \mid p(u) = p(u_t)\}) \\ 0, & \text{Otherwise} \end{cases}$$



- 四种类型的自我注意的输出被串联并通过归一化层，然后通过前馈网络生成此Transformer层的输出：

$$\widetilde{\mathbf{o}}_t^l = \|_{k=1}^{K} f_k(\mathbf{m}^{l-1}, \mathbf{h}_t^{l-1}, \mathbf{s}^{c_k})$$
$$\mathbf{h}_t^l = \mathrm{FeedForward}(\mathrm{LayerNorm}(\widetilde{\mathbf{o}}_t^l))$$

- 将最后一层的“[CLS]的隐藏状态作为查询话语和历史上下文的最终编码，并将其通过前馈神经网络传递以获得预期的情绪

$$\mathbf{h}_t = \mathbf{h}_{t,0}^L$$

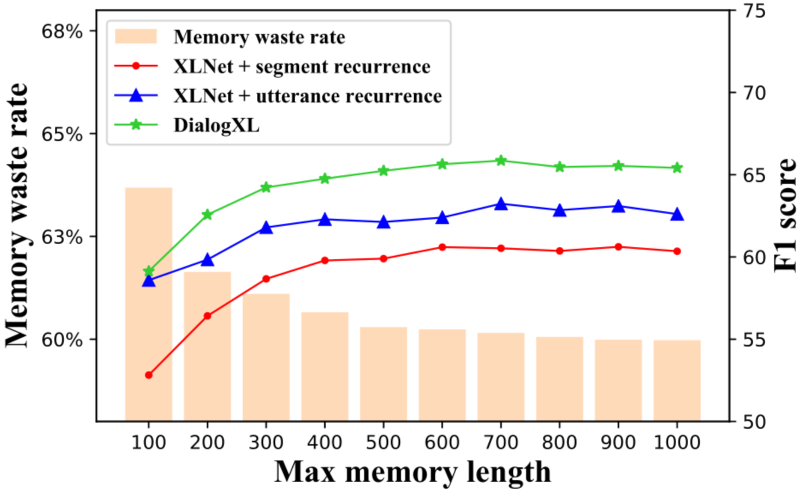
$$\mathbf{z}_t = \mathrm{ReLU}(\mathbf{W}_h \mathbf{h}_t + \mathbf{b}_h)$$
$$P_t = \mathrm{softmax}(\mathbf{W}_z \mathbf{z}_t + \mathbf{b}_z)$$
$$\widehat{y}_t = \mathrm{argmax}_{k \in \mathcal{S}}(P_t[k])$$

- 损失函数：标准交叉熵损失

$$\mathcal{L}(\theta) = -\sum_{i=1}^{M}\sum_{t=1}^{N} P_t[y_{i,t}]$$

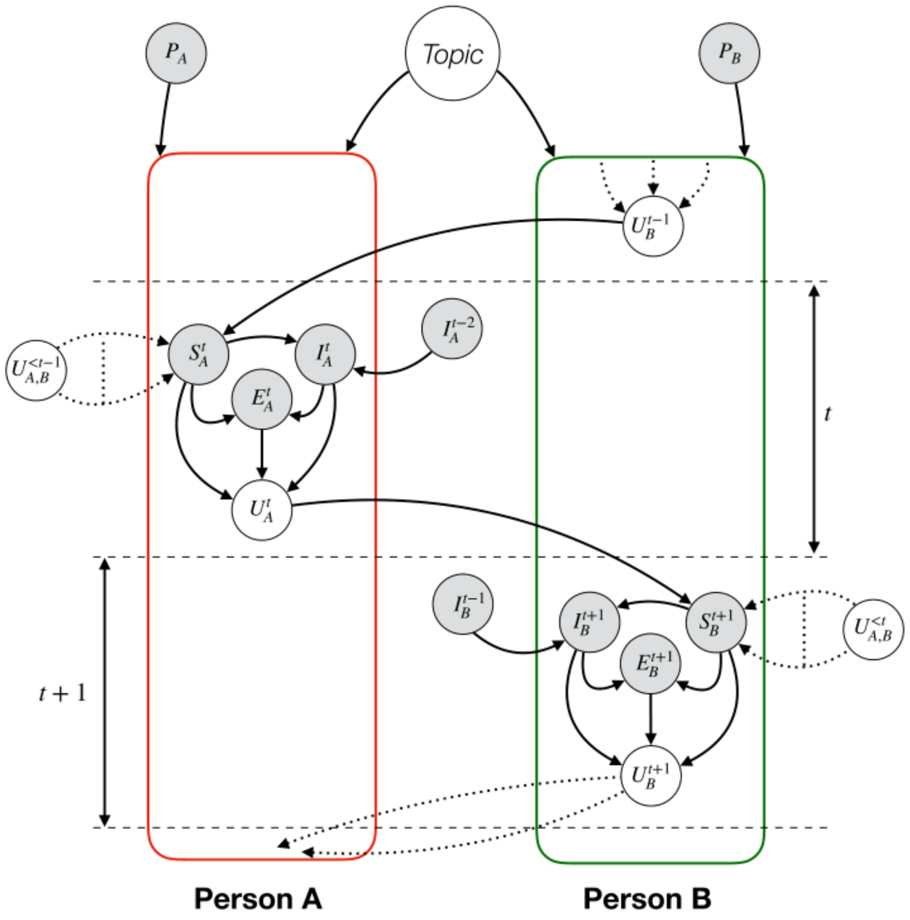
Experiments

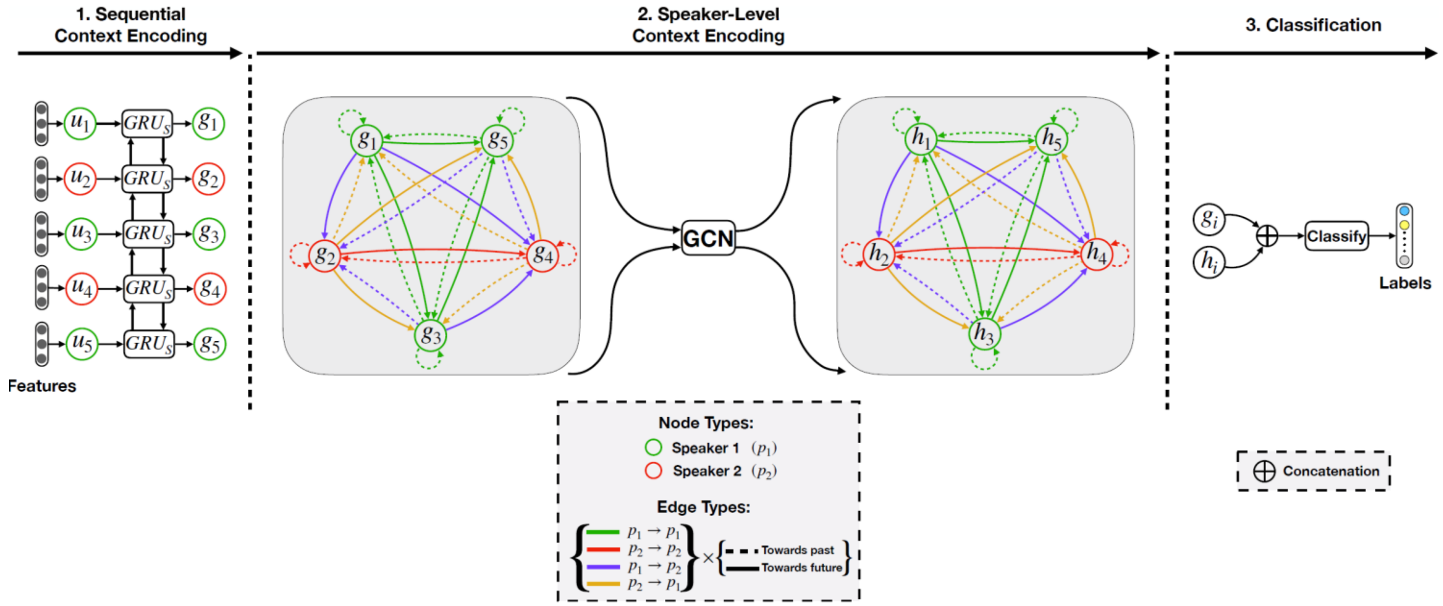
Model	IEMOCAP	MELD	DailyDialog	EmoryNLP
CMN	56.13	-	-	-
DialogueRNN	62.75	57.03	-	-
HiGRU	59.79*	56.92*	52.01*	31.88*
DialogueGCN	64.18	58.10	-	-
TL-ERC	59.30	57.46*	52.46*	30.57*
KET	59.56	58.18	53.37	33.95*
BERT	60.98	61.50	54.09	34.17
XLNet	61.33	61.65	53.62	34.13
DialogXL	**65.94**	**62.41**	**54.93**	**34.73**



Method	F1 score	
	IEMOCAP	MELD
DialogXL	**65.94**	**62.41**
- speaker self-attention	62.30 (↓3.64)	61.92 (↓0.49)
- listener self-attention	62.87 (↓3.07)	62.03 (↓0.38)
- speaker&listener self-attention	61.71 (↓4.23)	61.70 (↓**0.71**)
- local self-attention	61.66 (↓**4.28**)	61.72 (↓0.69)
- global self-attention	63.34 (↓2.60)	62.15 (↓0.26)

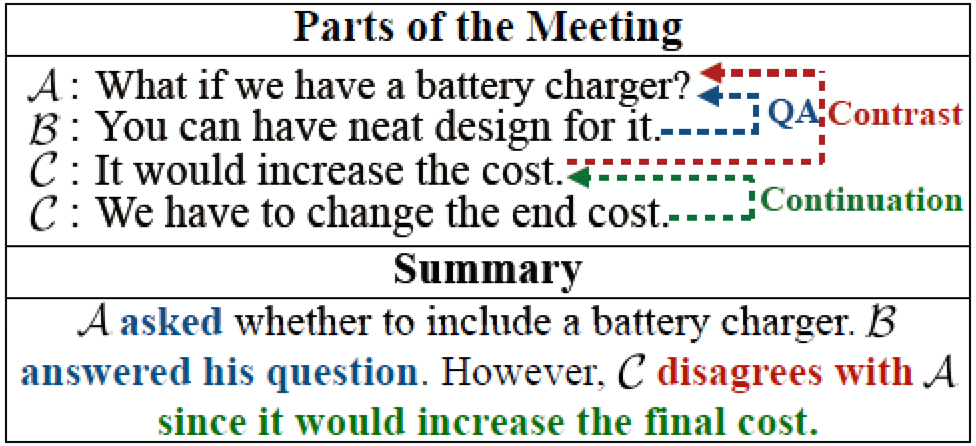
DialogueGCN





Dialogue Discourse-Aware Graph Model

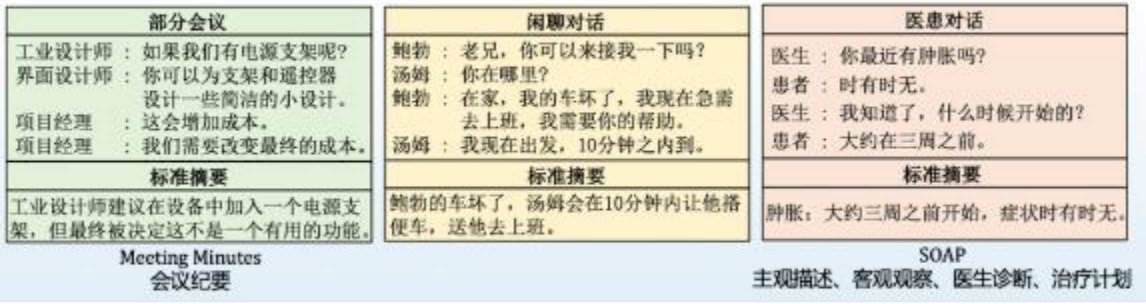
- 现存问题:
  - 会议文本建模不充分，将会议视为顺序句子序列进行建模，忽略了句子之间丰富的交互结构
  - 大规模训练数据集缺乏。
- 对话篇章结构
  - 指示了句子之间的交互关系，例如:问答、支持、反驳等。
  - 这种句子之间的交互关系是对话类型数据的显著特征。

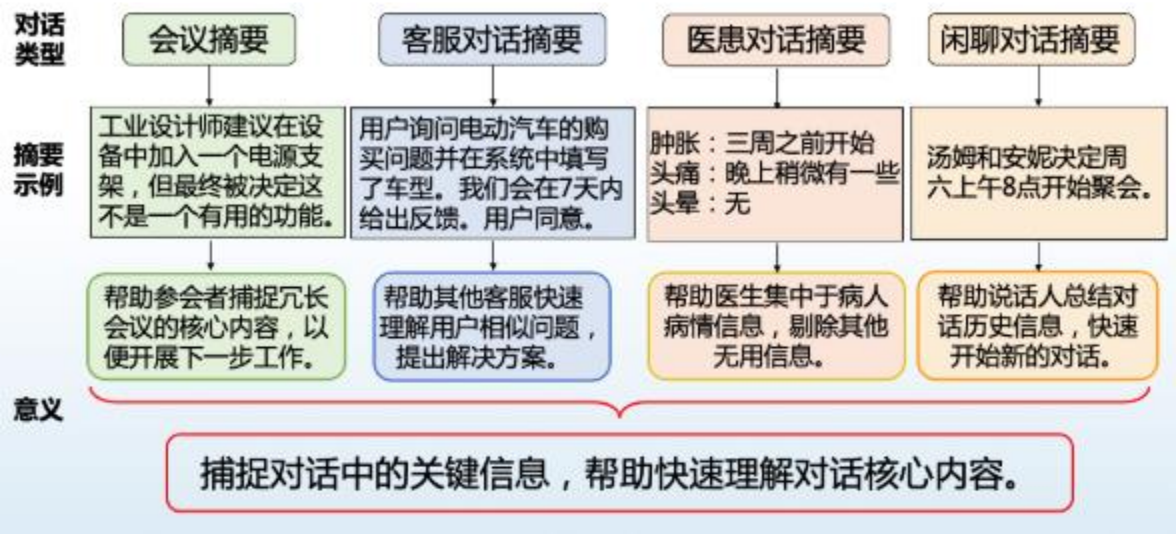


  - 评论，澄清-提问，阐述，致谢，继续，解释，条件，问答，交替，提问-阐述，结果，背景，叙述，纠正，平行，对比.

对话摘要

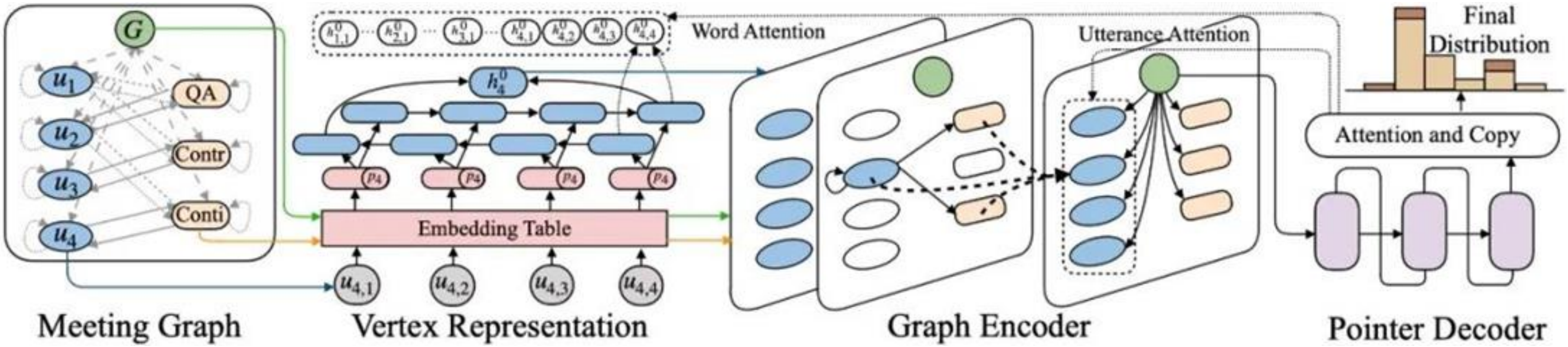
- 会议、闲聊、邮件、客服对话、医患对话、辩论等





Dialogue Discourse-Aware Meeting Summarizer

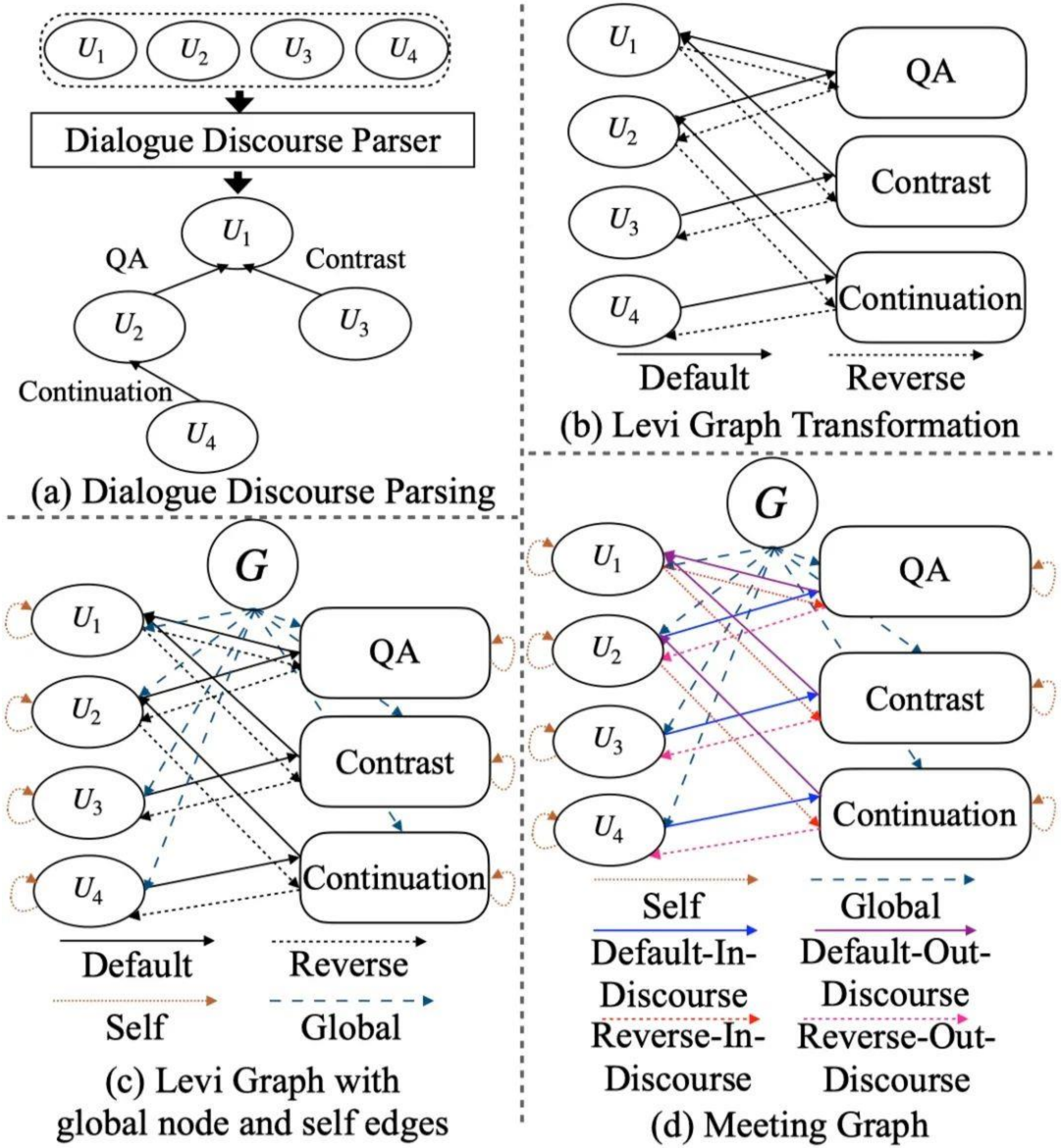
（1）会议图构建（2）节点表示（3）图编码器（4）指针解码器



会议图构建

1. 使用对话篇章结构解析器获得对话篇章结构关系；
2. Levi图转换，将边关系转换为节点，并添加正向与反向边；
3. 添加全局节点、全局边和自连边；
4. 添加出入边。

- default-in-discourse、default-out-discourse、reverse-in-discourse、reverse-out-discourse、global、self



Levi图

- 将原有的图变为二分图——左侧为原有节点，右侧为边
- 把边的类型转化为节点类型
- 通过levi图转换，我们可以显式建模模型篇章结构关系，并同步更新语句和篇章关系节点。

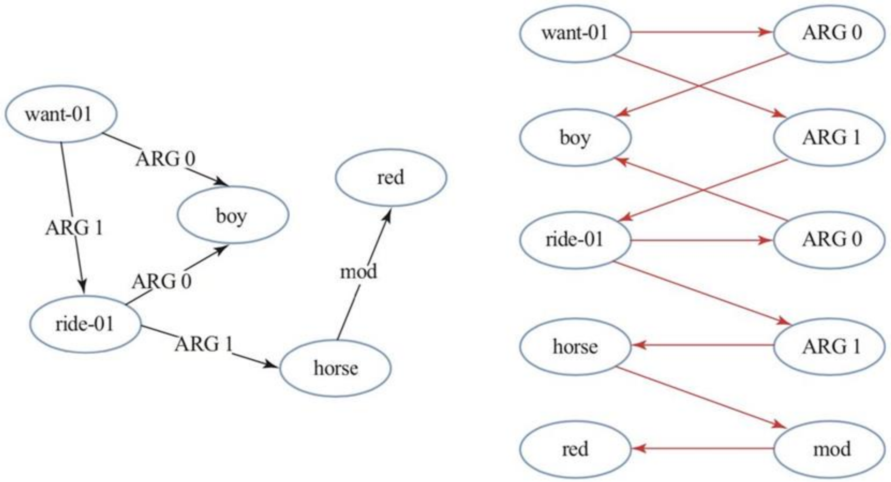


Figure 9.1: An example of AMR graph and its corresponding Levi graph. Left: The AMR graph of sentence *The boy wants to ride the red horse*. Right: The Levi transformation of the AMR graph.

节点表示

- 关系节点

  从一个可学习的关系编码矩阵中初始化
- 全局节点

  初始化为0向量
- 话语节点

  利用双向LSTM进行初始化。

  为了包含说话人信息，用one-hot向量编码speaker p_i，与词嵌入串接起来共同输入到LSTM中

- 图编码器：引入门控机制的RGCN

$$h_i^{(l+1)} = \mathrm{ReLU}\left(\sum_{r\in\mathbb{R}_M}\sum_{v_j\in\mathbb{N}_i^r}\frac{1}{|\mathbb{N}_i^r|}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_j^{(l)}\right)$$

$$g_j^{(l)} = \mathrm{sigmoid}\left(\boldsymbol{W}_{g,r}^{(l)}\boldsymbol{h}_j^{(l)}\right)$$

$$h_i^{(l+1)} = \mathrm{ReLU}\left(\sum_{r\in\mathbb{R}_M}\sum_{v_j\in\mathbb{N}_i^r}g_j^{(l)}\frac{1}{|\mathbb{N}_i^r|}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_j^{(l)}\right)$$

- 指针解码器：引入copy机制的decoder+词语级别注意力、句子级别注意力

$$e_{i,j}^t = \boldsymbol{s}_t^\top \boldsymbol{W}_a \boldsymbol{h}_{i,j}^0$$

$$\boldsymbol{a}^t = \mathrm{softmax}(\boldsymbol{e}^t)$$

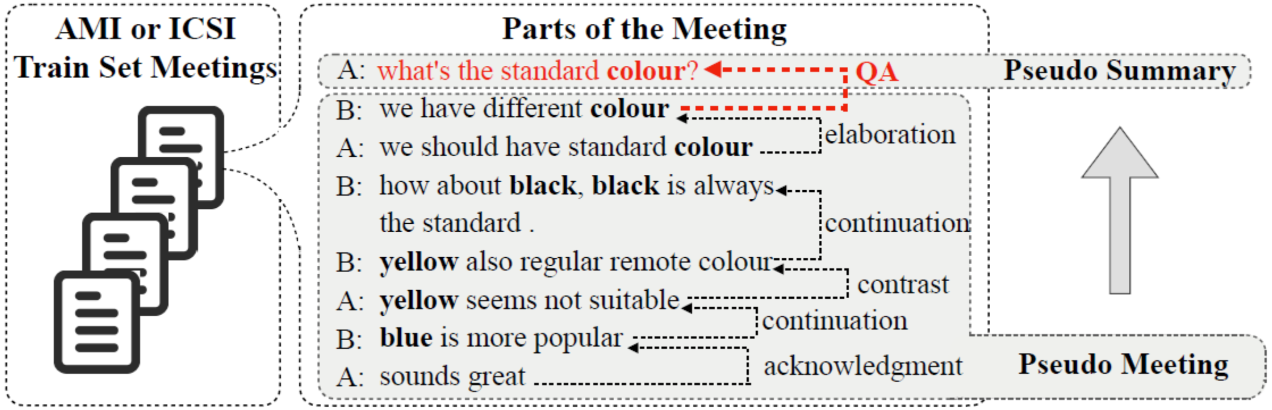
$$\boldsymbol{h}_t^{wl} = \sum_i\sum_j a_{i,j}^t \boldsymbol{h}_{i,j}^0$$

- 训练目标

$$\mathcal{L} = -\sum_{t=1}^{|\mathcal{Y}^*|}\log p\left(y_t^* | y_1^* \ldots y_{t-1}^*, \mathcal{U}\right)$$

数据增强

- 假设：一个问题往往会引起一段讨论，问题通常包含了讨论的核心要点内容
- 方法：问题视为伪造的摘要，讨论视为伪造的会议，从原始的会议数据集中构造了伪造摘要数据集。
- 过滤掉不包含名词和形容词的问题，以提高数据质量



Experiments

- 数据集
- 评价指标：ROUGE-1, ROUGE-2, ROUGE-L
- 人工评估：相关度、信息量

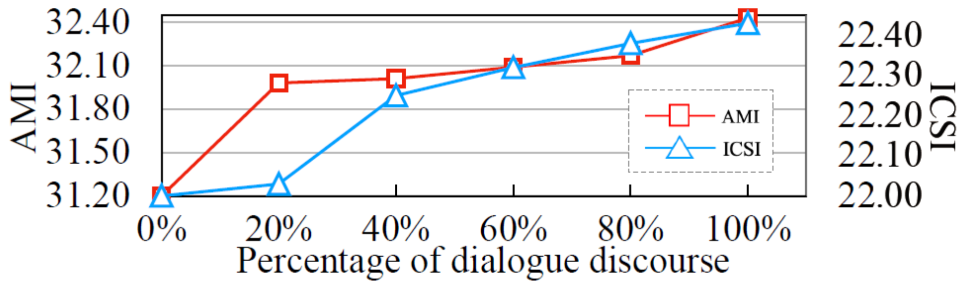
	AMI Pseudo Corpus	ICSI Pseudo Corpus
# of Original Data	97	53
# of Pseudo Data	1539	1877
Avg.Tokens	124.44	107.44
Avg.Sum	13.18	11.97

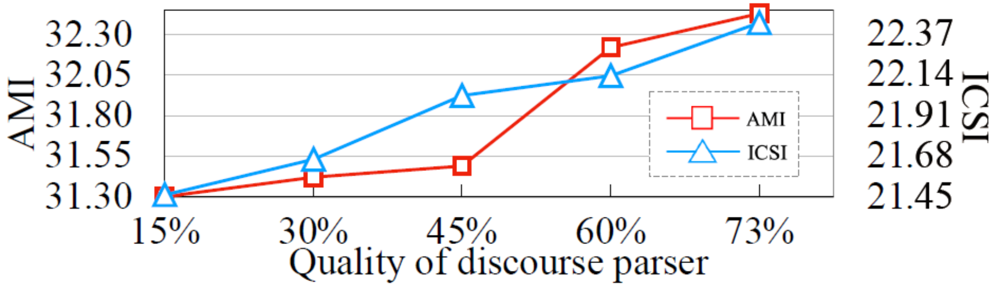
		AMI			ICSI		
	Model	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Extractive	TextRank [Mihalcea and Tarau, 2004]	35.19	6.13	15.70	30.72	4.69	12.97
	SummaRunner [Nallapati *et al.*, 2017]	30.98	5.54	13.91	27.60	3.70	12.52
Abstractive	UNS [Shang *et al.*, 2018]	37.86	7.84	13.72	31.73	5.14	14.50
	Pointer-Generator [See *et al.*, 2017]	42.60	14.01	22.62	35.89	6.92	15.67
	HRED [Serban *et al.*, 2016]	49.75	18.36	23.90	39.15	7.86	16.25
	Sentence-Gated [Goo and Chen, 2018]	49.29	19.31	24.82	39.37	9.57	17.17
	TopicSeg [Li *et al.*, 2019]	51.53	12.23	25.47	-	-	-
	HMNet [Zhu *et al.*, 2020]	52.36	18.63	24.00	**45.97**	10.14	18.54
Ours	DDAMS	51.42	20.99	24.89	39.66	10.09	17.53
	DDAMS + DDADA	**53.15**	**22.32**	**25.67**	40.41	**11.02**	**19.18**
	DDAMS + DDADA (w/o fine-tune)	28.35	4.67	14.92	25.94	4.18	13.92

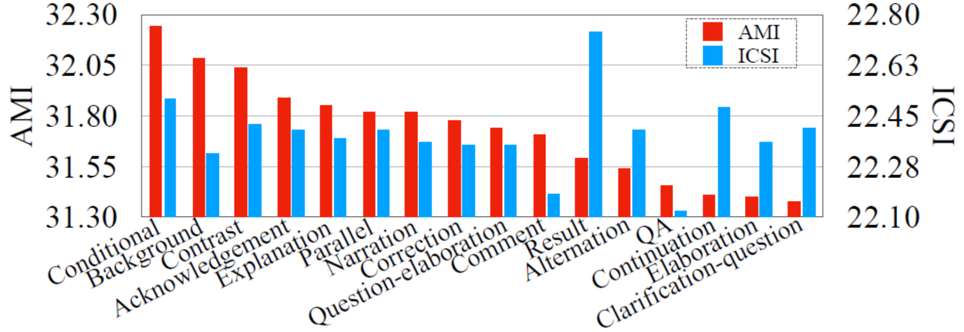
	Model	**Relevance**	**Informativeness**
AMI	Ground-truth	4.60	4.56
	Sentence-Gated	3.16	3.60
	HMNet	3.60	3.72
	DDAMS	3.80	3.76
	DDAMS +DDADA	**3.84**	**3.88**
ICSI	Ground-truth	4.76	4.48
	Sentence-Gated	3.32	3.48
	HMNet	3.80	3.52
	DDAMS	3.76	3.28
	DDAMS +DDADA	**3.84**	**3.60**

分析实验

- 对话篇章结构的数量、质量、类型；会话图；伪摘要数据

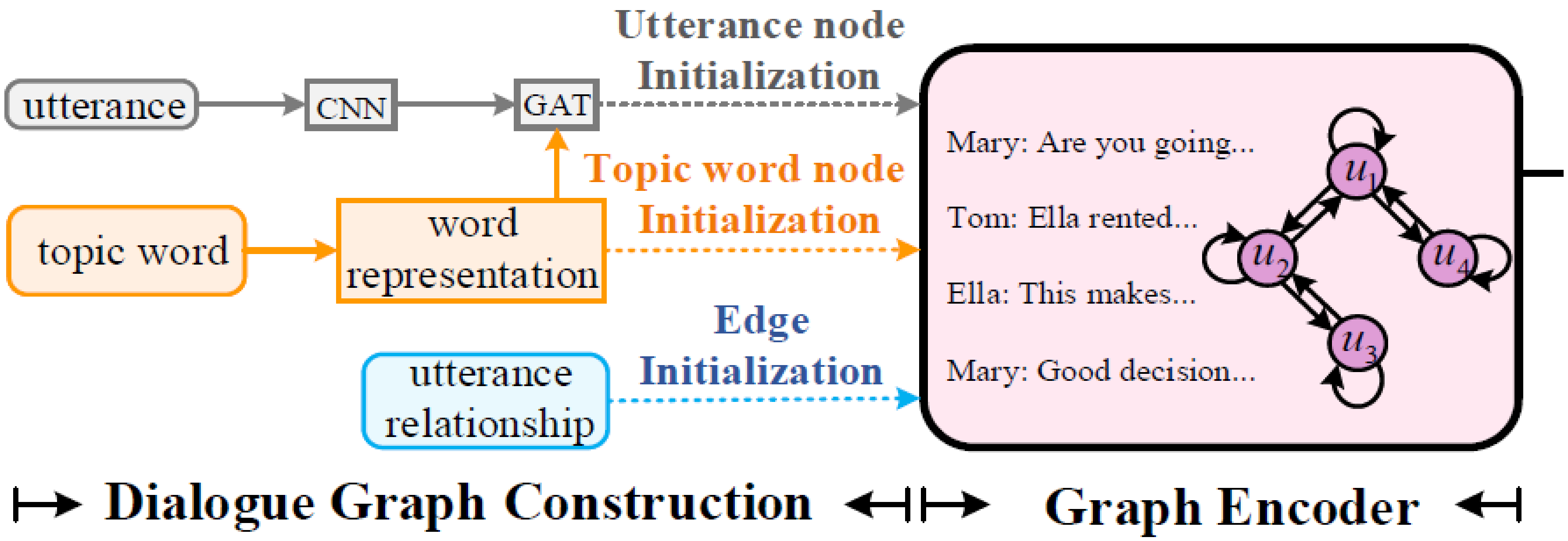


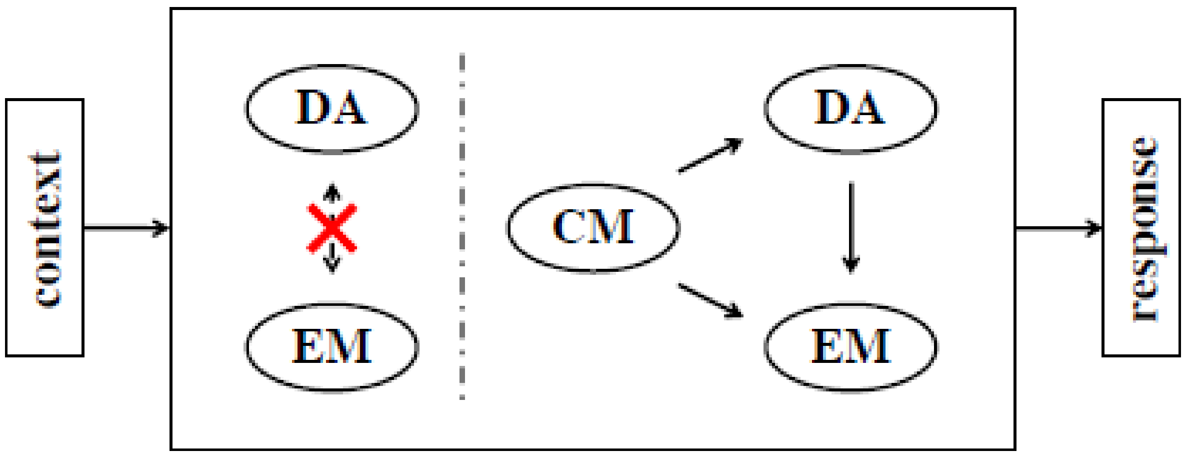




	Model	R-1	R-2	R-L
AMI	DDAMS	51.42	**20.99**	**24.89**
	DDAMS (w/ Levi graph)	**51.46**	20.75	24.31
ICSI	DDAMS	**39.66**	**10.09**	**17.53**
	DDAMS (w/ Levi graph)	39.20	9.54	17.48

	Model	R-1	R-2	R-L
AMI	DDAMS	51.42	20.99	24.89
	+ RBDA	52.94	21.96	25.05
	+ DDADA	**53.15**	**22.32**	**25.67**
ICSI	DDAMS	39.66	10.09	17.53
	+ RBDA	39.42	10.60	18.19
	+ DDADA	**40.41**	**11.02**	**19.18**


Utterance node
Initialization
utterance
CNN
GAT
Topic word node
Initialization
topic word
word
representation
Edge
Initialization
utterance
relationship
Mary: Are you going...
Tom: Ella rented...
Ella: This makes...
Mary: Good decision...
Dialogue Graph Construction
Graph Encoder


context
DA
EM
CM
DA
EM
response

HRED

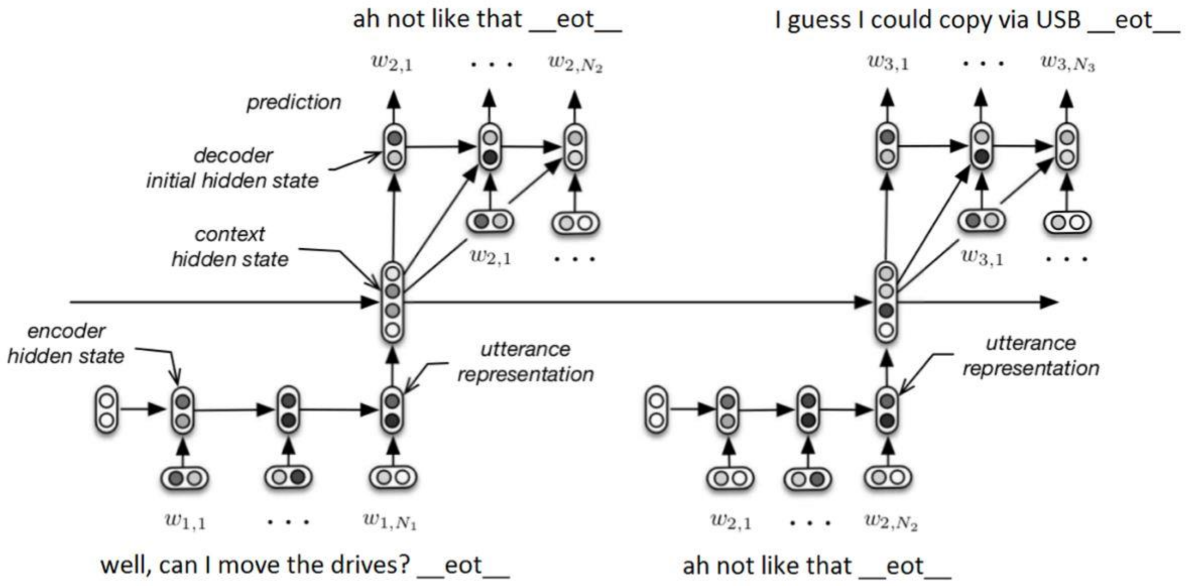
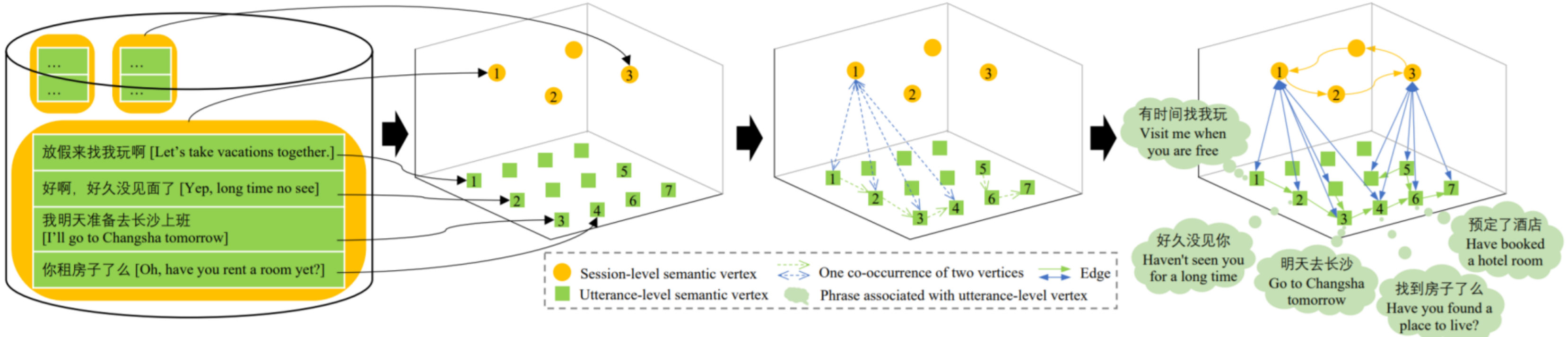


Figure 15: Diagram of the HRED model. Note that each utterance in the context is encoded with a separate 'utterance-level' encoder, which is then fed into a 'context-level' encoder.

DVAE-GNN

- 问题：可以生成局部相关的对话，但难以生成全局相关的对话
- 对话结构的作用：
  - 提供对对话结构的基本见解；
  - 提供背景知识来促进对话的生成
- 目标：从用于对话生成的对话语料库中，自动发现可解释的开放域对话结构



(a) A set of dialog sessions (b) Map each dialog session to vertices (c) Collect co-occurrence statistics of mapped vertices for all dialog sessions (d) Build edges between vertices to form dialog structure graph

DVAE-GNN

- 现状:
  - 先前没有针对开放域对话结构工作
  - 任务对话结构无法直接用于开放域
    - 只能发现话语级别的结构，无法发现会话级（聊天主题）
    - 不适用于开放域对话的大规模
- 贡献:
  - 开放域对话的两层对话结构
  - DVAE-GNN
    - 整合GNN到VAE中，微调语句级别语义，有效识别会话级别语义顶点
    - 耦合机制，用相关短语给话语节点提供先验知识，来缓解VAE发现大规模语义的难度

点、边与初始化

2种类型的点

- Session-level：可学习的隐向量
- Utterance-level：
  可学习的隐向量
  相关短语表示向量

Algorithm 1 Phrase extraction

Input: An utterance U
Output: A set of phrases E extracted from U

Obtain a dependency parse tree T for U;
Get all the head words HED that are connected to ROOT node, and all the leaf nodes in T (denoted as L);
for each leaf node in $|L|$ **do**
    Extract a phrase consisting of words along the tree from HED to current leaf node, denoted as e_i;
    If e_i is a verb phrase, then append it into E;
end for
return E

3种类型的边

- Utter-Utter:
  表示自然的对话间转换
  初始化时，若相关短语可以从同一个session的邻接两句话中顺序提取出来，则连边
- Sess-Sess：
  表示自然的对话间转换
  通过DVAE-GNN动态学习
- Utter-Sess：
  表示话语与会话之间的父子层级关系，一条话语可以有多个父级会话节点
  通过DVAE-GNN动态学习

Vertex Recognition

- Utterance-level
  1. 用RNN编码话语x_i，获得话语表示向量e(x_i)
  2. 用FFN计算话语级别后验分布 $z_i \sim q(z|x_i) = \mathrm{softmax}(\boldsymbol{\Lambda}_x \boldsymbol{e}(x_i))$
  3. 用gumble-softmax从后验分布中采样，获得utter级别的语义节点z_i
- Session-level
  1. 在Utter-Utter边上使用三层GCN计算

$$\boldsymbol{h}^j_{v^u_n} = \sigma^j\Big(\sum_{v^u_{n'} \in \mathcal{N}(v^u_n)} \boldsymbol{h}^{j-1}_{v^u_{n'}}\Big),$$

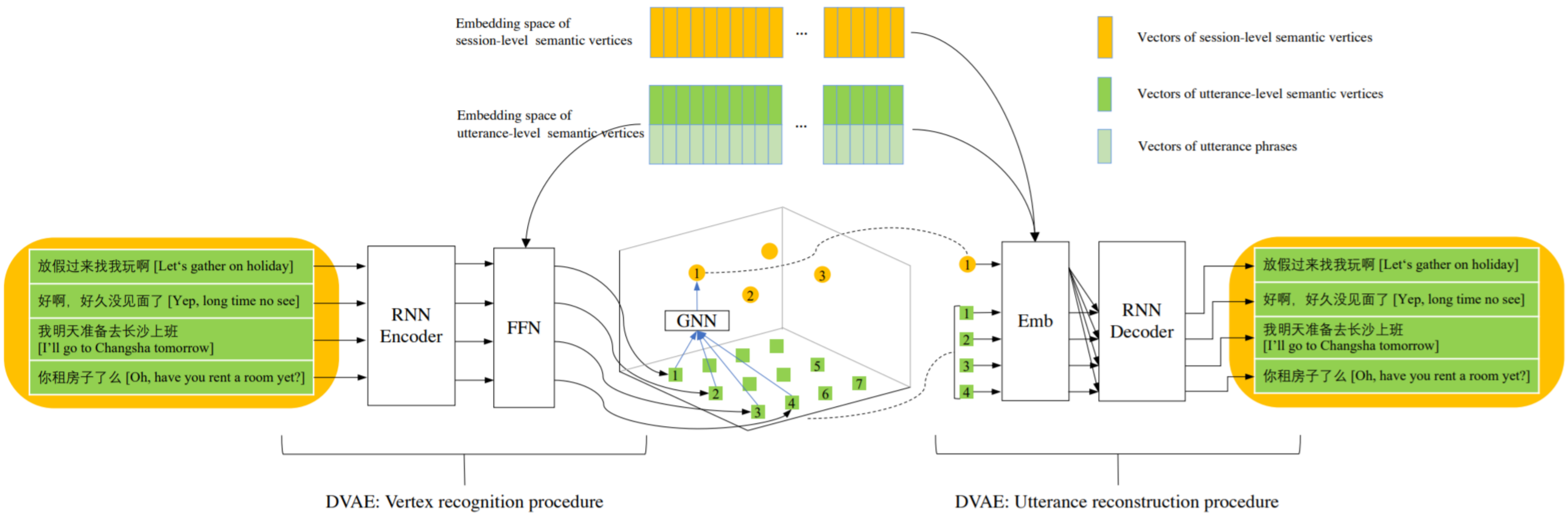
  2. 获得结构感知的语义序列，输入到RNN Encoder，获得结构感知的会话表示$e(z_{1,\ldots,c})$
  3. 计算session级别后验分布 $g \sim q(g|z_{1,\ldots,c}) = \mathrm{softmax}(\boldsymbol{\Lambda}_g \boldsymbol{e}(z_{1,\ldots,c}))$

Utterance Reconstruction

将映射的话语级和会话级语义顶点输入到到RNN解码器

Loss Function

$$\mathbb{E}_{q(Z|X)}[\log p(X|Z)] - KL(q(Z|X)\|p(Z)),$$

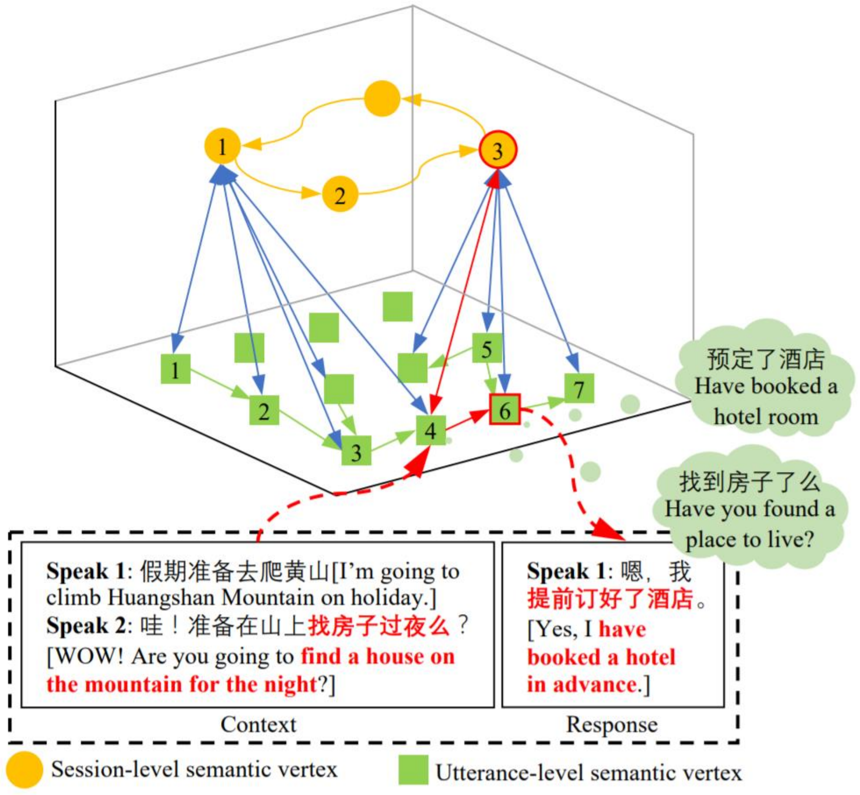


Graph Construction

- 将对话会话映射到图中的语义顶点，并基于映射顶点的共现统计在语义顶点之间建立边。
- 步骤
  - 映射：将corpus中所有的session映射成顶点，包括sess级和utter级。
  - 统计：收集这些顶点的共现统计数据。
    - 每个sess级顶点的映射次数$\#(v_i^s)$
    - 每个utter级语义顶点的映射次数$\#(v_i^u)$
    - 收集分别由对话会话和其中的话语映射的会话级语义顶点和话语级语义顶点的共现频率，$\#(v_i^s, v_i^u)$
    - 对话会话中由两个相邻话语顺序映射的两个话语级语义顶点的共现频率 $\#(v_j^u, v_i^u)$
  - 基于共现统计数据建立边:
    - 有向Utter-Utter: $\#(v_j^u, v_k^u)$/ (v_j^u)>阈值a^{uu}
    - 双向Sess-Utter: $\#(v_i^s, v_j^u)$/ (v_j^u)>阈值a^{su}
    - 有向Sess-Sess: $\#(v_i^s, v_o^s)$/ (v_i^s)>阈值a^{ss}, 分子是连接到两个Sess点的Utter点的数量。Sess-sess边 依赖于 sess-utter边

Graph Grounded Conversational System

- 将多轮对话生成表述为基于图的强化学习问题，其中图中的顶点充当强化学习动作。
- GCS包含三个模块:
  - **对话上下文理解**：识别最相关的话语级语义顶点，映射到图中；获得RL的状态表示
  - **策略**：包含两个子策略
  - **响应生成**：预训练好的带注意力的seq2seq



Dialog Policy Learning

- Sess级别子策略
  - 确定对话目标
  - 选择一个命中utter节点的父sess节点

$$\mu^{g}(s_l, v^{s}_{c^{g}_{j}}) = \frac{\exp(\boldsymbol{e}_{s_l}{}^{T}\boldsymbol{\Lambda}_{g}[c^{g}_{j}])}{\sum_{k=1}^{N^{g}_{l}} \exp(\boldsymbol{e}_{s_l}{}^{T}\boldsymbol{\Lambda}_{g}[c^{g}_{k}])}$$

- Utter级别子策略
  - 从当前对话目标的所有子节点中选择内容合适的

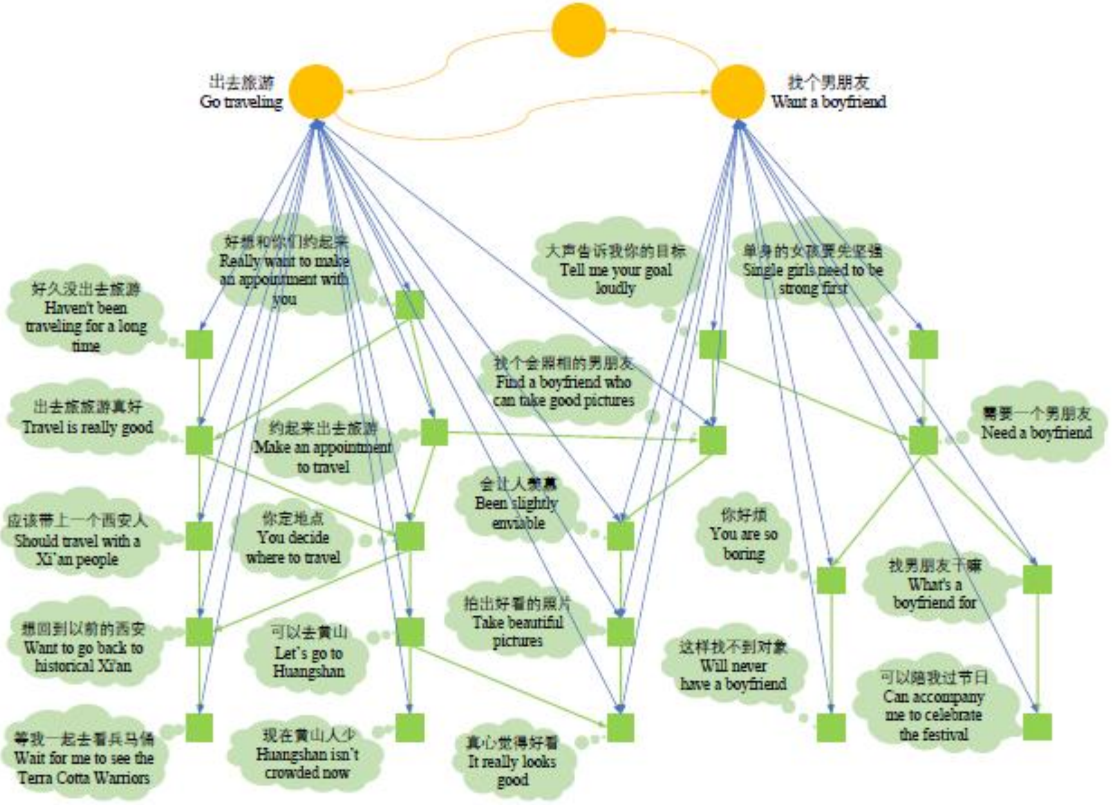
$$\mu^{u}(s_l, v^{u}_{c^{u}_{j}}) = \frac{\exp(\boldsymbol{e}_{s_l}{}^{T}\boldsymbol{\Lambda}_{x}[c^{u}_{j}])}{\sum_{k=1}^{N^{u}_{l}} \exp(\boldsymbol{e}_{s_l}{}^{T}\boldsymbol{\Lambda}_{x}[c^{u}_{k}])}$$

RL rewards

- Sess级子策略:
  - 当前对话目标下所有子utter级子策略的平均值
- Utter级子策略：以下三种因子的加权和。
  - 话语相关度
    - 生成的response与当前对话上下文连贯一致
    - DAM
  - 话语-目标接近程度
    - 选择的语句与当前目标应该尽可能相近
    - 衡量指标：#(v_i^s, v_j^u)/ (v_j^u)
  - 重复惩罚:
    - 提升回复多样性
    - 当所选话语与一个上下文中的话语有60%以上的词相同时，该因子为1，否则为0。

实验：图结构发现

Datasets	Methods	Automatic Evaluation		Human Evaluation		
		NLL	BLEU-1/2.	S-U Appr.	U-U Appr.	Intra-Goal Rele.
Weibo	DVRNN	29.187	0.427/0.322	-	0.16	-
	Phrase Graph	-	-/-	-	0.63	-
	DVAE-GNN	**20.969**	**0.588/0.455**	**0.85**	**0.79**	**1.44**
	DVAE-GNN w/o GNN	23.364	0.560/0.429	0.53	0.78	1.06
	DVAE-GNN w/o phrase	24.282	0.468/0.355	0.43	0.27	0.95
Douban	DVRNN	72.744	0.124/0.093	-	0.14	-
	Phrase Graph	-	-/-	-	0.34	-
	DVAE-GNN	**35.975**	**0.525/0.412**	**0.60**	**0.70**	**0.93**
	DVAE-GNN w/o GNN	37.415	0.504/0.394	0.38	0.54	0.48
	DVAE-GNN w/o phrase	49.606	0.254/0.206	0.28	0.19	0.27

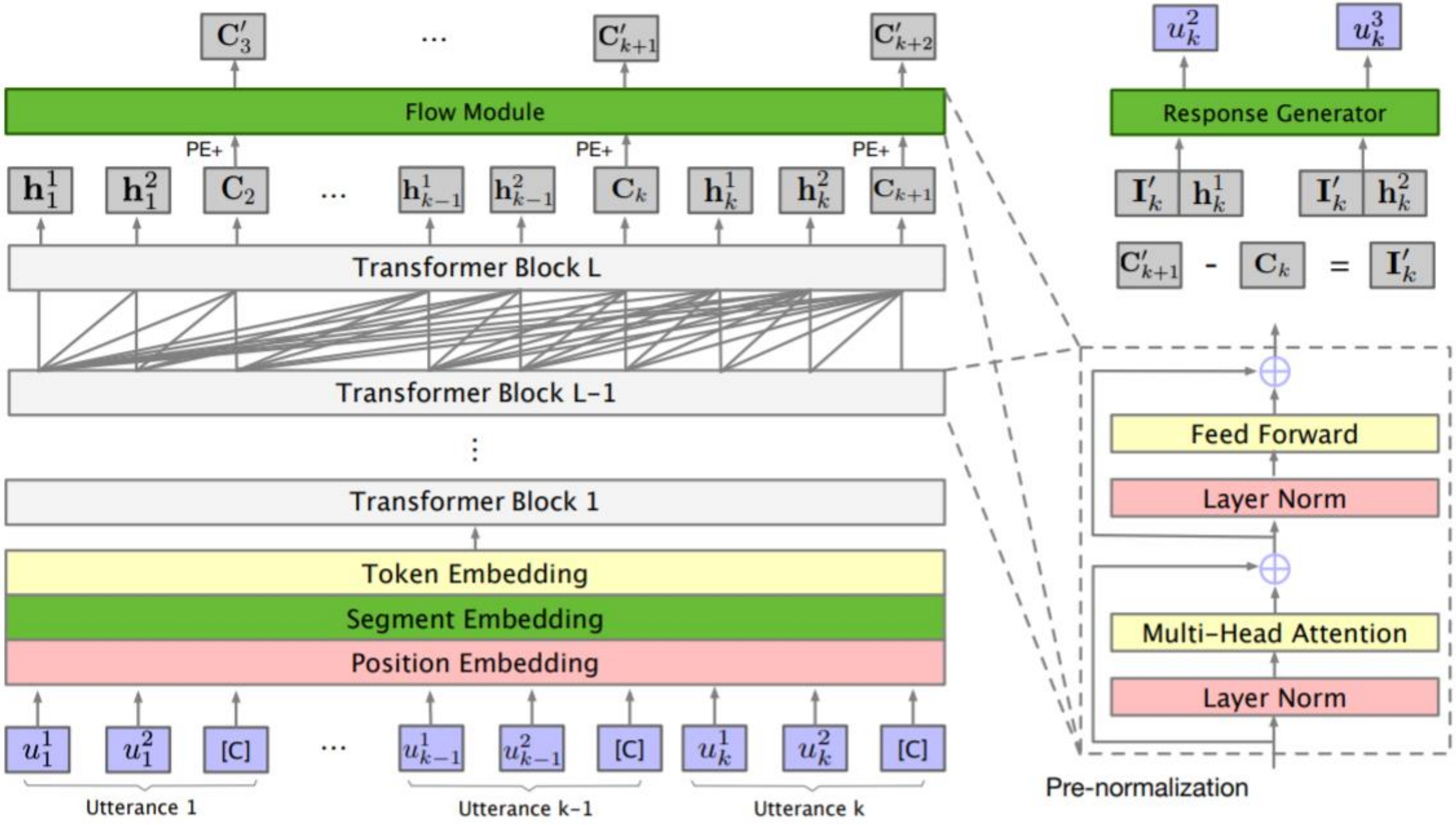


实验：对话生成

Methods	Coherence		Informativeness		Overall Quality	
	Intra-Cohe.*	Appr.*	Info.*	Dist-1/2#	Enga.*	Length#
HRED	0.54	0.43	0.19	0.08/0.26	0.20	5.04
MMPMS	0.66	0.45	0.50	0.08/0.32	0.24	5.82
CVAE	0.58	0.39	0.43	0.11/0.38	0.22	7.74
VHCR-EI	0.68	0.43	0.53	0.12/0.36	0.28	7.30
MemGM	0.53	0.37	0.34	0.09/0.33	0.20	4.08
DVRNN-RL	0.60	0.39	0.39	0.06/0.22	0.22	7.86
GCS	**1.03**	**0.59**	**0.58**	**0.19/0.55**	**0.48**	**8.00**
GCS w/ UtterG	0.93	0.56	0.55	0.16/0.47	0.34	**8.00**

Start	国庆节快到了，我快放假了。[National Day is coming, I'm going to have a holiday.]
User	放假了，我们可以出去玩了，放风筝[In holiday, we can go out and have fun, such as flying kites.]
Bot	我星期五就到了。[I'll come back on Friday.]
User	那我在家等你回来[All right, I'll wait for you at home]
Bot	好的呀，哈哈哈。那就等着你的大餐[Great! Looking forward to your big meal.]
User	哈哈，我想去南京旅游，一起吧[Well, I want to travel to Nanjing, let's go together.]
Bot	不要我了！[I feel like you're abandoning me.]
User	要你，咱们一起去[No, you can come with us.]
Bot	看你秀恩爱好嘛。[Is it appropriate to travel with a lovey-dovey couple?]
User	秀恩爱，分的快，这句话我不同意[I don't think it is reasonable that showing sweet makes a lovey-dovey couple broke up fast.]
Bot	那你好棒棒哦[You are vey nice.]
User	一般啦，并不是很棒[Just so-so.]
Bot	好的吧，我觉得你很好看[All right, you are beautiful in my eyes!]
User	哈哈，你好好笑哦，敷衍我！[Ha-ha, you are so funny. You're just flattering me.]
Bot	对不起哈哈哈[Yeah, I'm sorry.]
User	没关系，我很大度，原谅你了[That's doesn't matter, you're forgiven]
Bot	怎么有这么好的事[You are so generous. How could such a good thing happen.]

DialoFlow



C'_3
...
C'_{k+1}
C'_{k+2}
Flow Module
PE+
PE+
PE+
h_1^1
h_1^2
C_2
...
h_{k-1}^1
h_{k-1}^2
C_k
h_k^1
h_k^2
C_{k+1}
Transformer Block L
Transformer Block L-1
Transformer Block 1
Token Embedding
Segment Embedding
Position Embedding
u_1^1
u_1^2
[C]
...
u_{k-1}^1
u_{k-1}^2
[C]
u_k^1
u_k^2
[C]
Utterance 1
Utterance k-1
Utterance k
u_k^2
u_k^3
Response Generator
I'_k
h_k^1
I'_k
h_k^2
C'_{k+1} - C_k = I'_k
Feed Forward
Layer Norm
Multi-Head Attention
Layer Norm
Pre-normalization

Thanks~ (ฅ>ω<*ฅ)

Good, better, best,
Never let it rest,
Till good is better,
And better best.