基于检索的多轮对话

汇报人：陶思雨

task

	Context
utterance 1	*Human*: How are you doing?
utterance 2	*ChatBot*: I am going to **hold a drum class** in Shanghai. Anyone wants to join? The location is near Lujiazui.
utterance 3	*Human*: Interesting! Do you have coaches who can help me practice **drum**?
utterance 4	*ChatBot*: Of course.
utterance 5	*Human*: Can I have a free first lesson?
	Response Candidates
response 1	Sure. Have you ever played drum before? ✓
response 2	What lessons do you want? ✗

task

1.Difference with single-turn

* ⁎ the topic will change
* ⁎ need to consider the relation between utterances

2.Chanllenge of this task

* ⁎ how to identity important information(word, phrase, and sentences) in context
* ⁎ how to model relationships among the utterances in the context

SMN

ACL2017

Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots

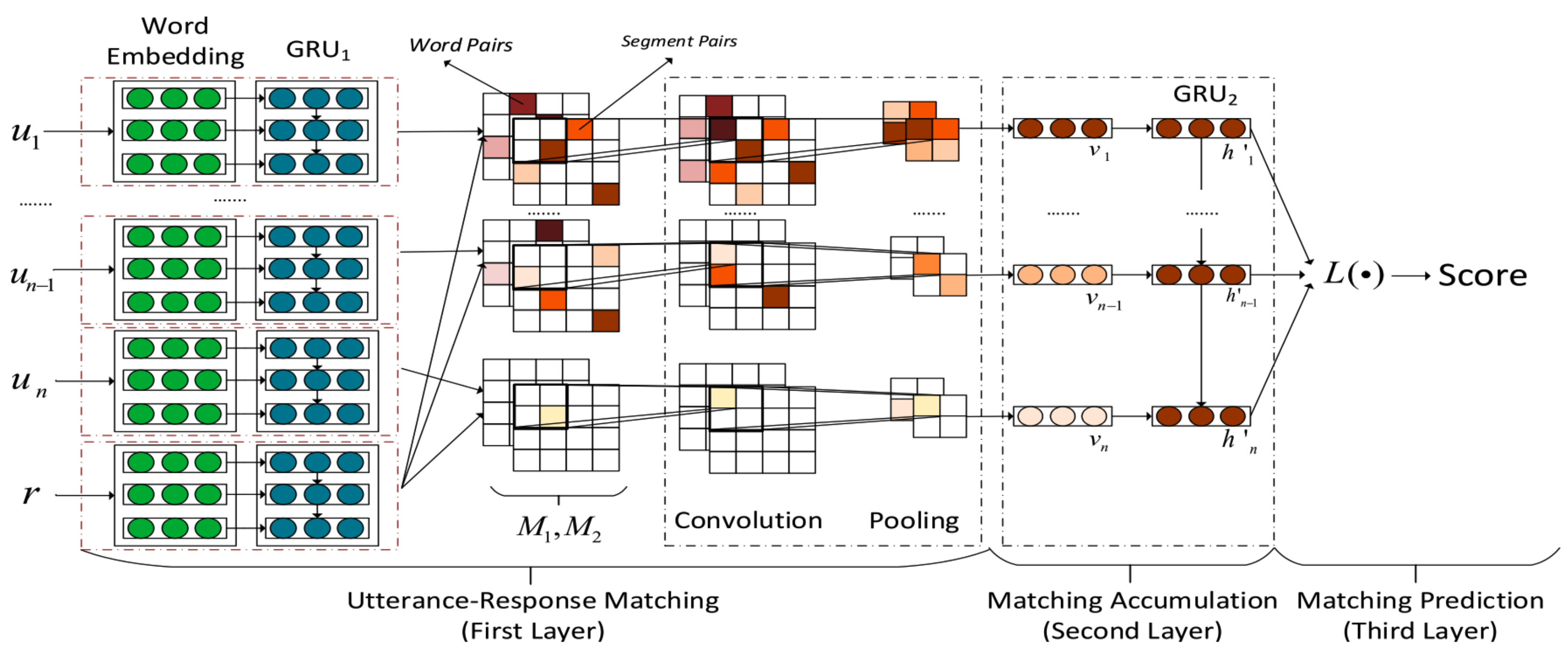


Figure 1: Architecture of SMN

SMN

ACL2017

Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots

Utterance-Response Matching

U = $[e_{u,1}, \ldots, e_{u,n_u}]$

$$\mathbf{R} = [e_{r,1}, \ldots, e_{r,n_r}]$$

word-word similarity

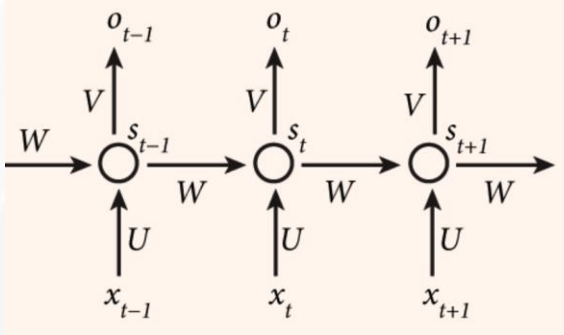
$$e_{1,i,j} = e_{u,i}^{\top} \cdot e_{r,j} \rightarrow \mathbf{M}_1 \in \mathbb{R}^{n_u \times n_r}$$

sequence-sequence similarity

GRU:

$$\begin{aligned} z_i &= \sigma(\mathbf{W_z} e_{u,i} + \mathbf{U_z} h_{u,i-1}) \\ r_i &= \sigma(\mathbf{W_r} e_{u,i} + \mathbf{U_r} h_{u,i-1}) \\ \widetilde{h}_{u,i} &= tanh(\mathbf{W_h} e_{u,i} + \mathbf{U_h}(r_i \odot h_{u,i-1})) \\ h_{u,i} &= z_i \odot \widetilde{h}_{u,i} + (1 - z_i) \odot h_{u,i-1}, \end{aligned}$$

$$e_{2,i,j} = h_{u,i}^{\top} \mathbf{A} h_{r,j} \rightarrow \mathbf{M}_2 \in \mathbb{R}^{n_u \times n_r}$$



Utterance-Response Matching

CNN

$$z_{i,j}^{(l,f)} = \sigma\left(\sum_{f'=0}^{F_{l-1}} \sum_{s=0}^{r_w^{(l,f)}} \sum_{t=0}^{r_h^{(l,f)}} \mathbf{W}_{s,t}^{(l,f)} \cdot z_{i+s,j+t}^{(l-1,f')} + \mathbf{b}^{l,k}\right)$$

Max-pool

$$z_{i,j}^{(l,f)} = \max_{p_w^{(l,f)} > s \geq 0} \max_{p_h^{(l,f)} > t \geq 0} z_{i+s,j+t} \rightarrow v \in \mathbb{R}^q$$

MSN

ACL2019 **Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots**

Turns	Dialogue Text	SMN	DAM
Turn-1	A: Are there any **discounts** activities recently?		
Turn-2	B: No. Our product have been **cheaper** than before.		
Turn-3	A: Oh.		
Turn-4	B: Hum!		
Turn-5	A: I'll buy these nuts. Can you sell me **cheaper**?		
Turn-6	B: You can get some **coupons** on the homepage.		
Turn-7	A: Will you give me some nut **clips**?		
Turn-8	B: Of course **we will**.		
Turn-9	A: **How many clips** will you give?		
Resp-1	**One clip** for every package. (True)	0.832	0.854
Resp-2	OK, **we will** give you a **coupons** worth $1. (False)	0.925	0.947

We can see that although “Resp-1” is the right answer for utterance “Turn-9”, the SMN and DAM models still choose “Resp-2”. Because it has more words overlap with context utterances, thus accumulating a larger similarity score.

MSN

ACL2019

Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots

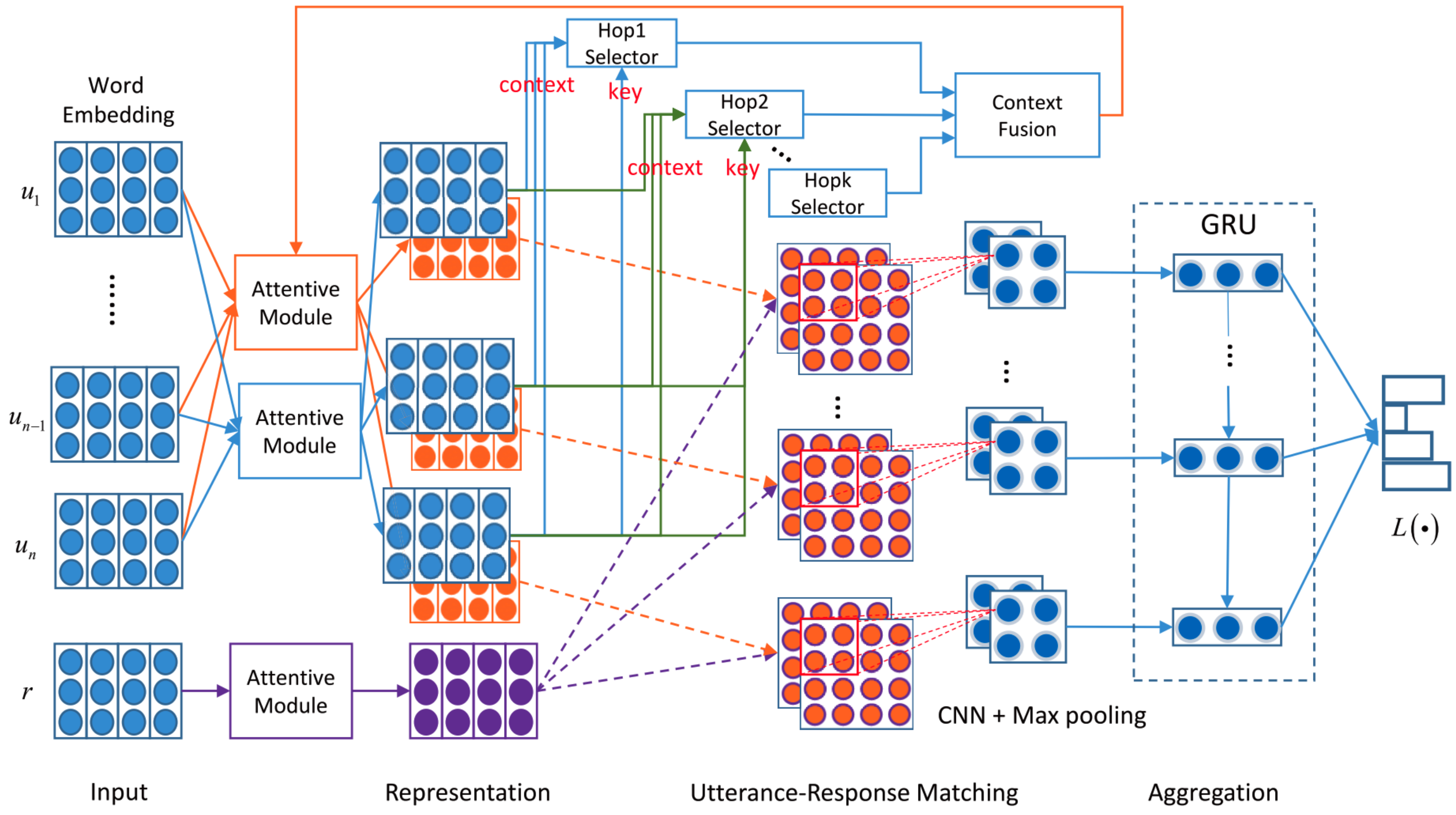


Figure 1: Architecture of multi-hop selector network.

Hop1 Selector

$$\mathbf{U}_i = [\mathbf{u}_{i1}, \ldots, \mathbf{u}_{ij}, \ldots, \mathbf{u}_{iL}]$$

$$\mathbf{u}'_{ij} = \textbf{AttentiveModule}(\mathbf{u}_{ij}, \mathbf{u}_{ij}, \mathbf{u}_{ij})$$

$$\text{Attention}(Q, K, V, M) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}} + M)V,$$

$$\text{head}_i = \text{Attention}(EW_i^Q, EW_i^K, EW_i^V, M),$$

$$\text{MHSA}(E, M) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O,$$

Word Selector

$$\mathbf{A} = \mathbf{v}^T \tanh(\mathbf{K}_1^T \mathbf{W} \mathbf{U}'_i + \mathbf{b}) \quad \mathbf{K}_1 = \mathbf{u}'_{iL}$$

$$\mathbf{A} \in \mathbb{R}^{L \times T \times T}$$

$$m_1(\mathbf{K}_1, \mathbf{U}'_i) = [\max_{dim=2} \mathbf{A}; \max_{dim=3} \mathbf{A}]$$

$$m_1(\mathbf{K}_1, \mathbf{U}'_i) \in \mathbb{R}^{L \times 2T}$$

$$\mathbf{s}_1 = \textbf{softmax}(m_1(\mathbf{K}_1, \mathbf{U}'_i)\mathbf{c} + b)$$

Utterance Selector

$$\widetilde{\mathbf{U}}_i = \mathbf{mean}(\mathbf{U}'_i)$$

$$s_2 = \frac{\widetilde{\mathbf{U}}_i \mathbf{K}_2^T}{||\widetilde{\mathbf{U}}_i||_2 \, ||\mathbf{K}_2||_2} \quad \mathbf{K}_2 = \widetilde{\mathbf{U}}_{iL}$$

$$\mathbf{s}_2 \in \mathbb{R}^{L \times 1}$$

$$\mathbf{s}^{(1)} = \alpha * \mathbf{s}_1 + (1 - \alpha) * \mathbf{s}_2$$

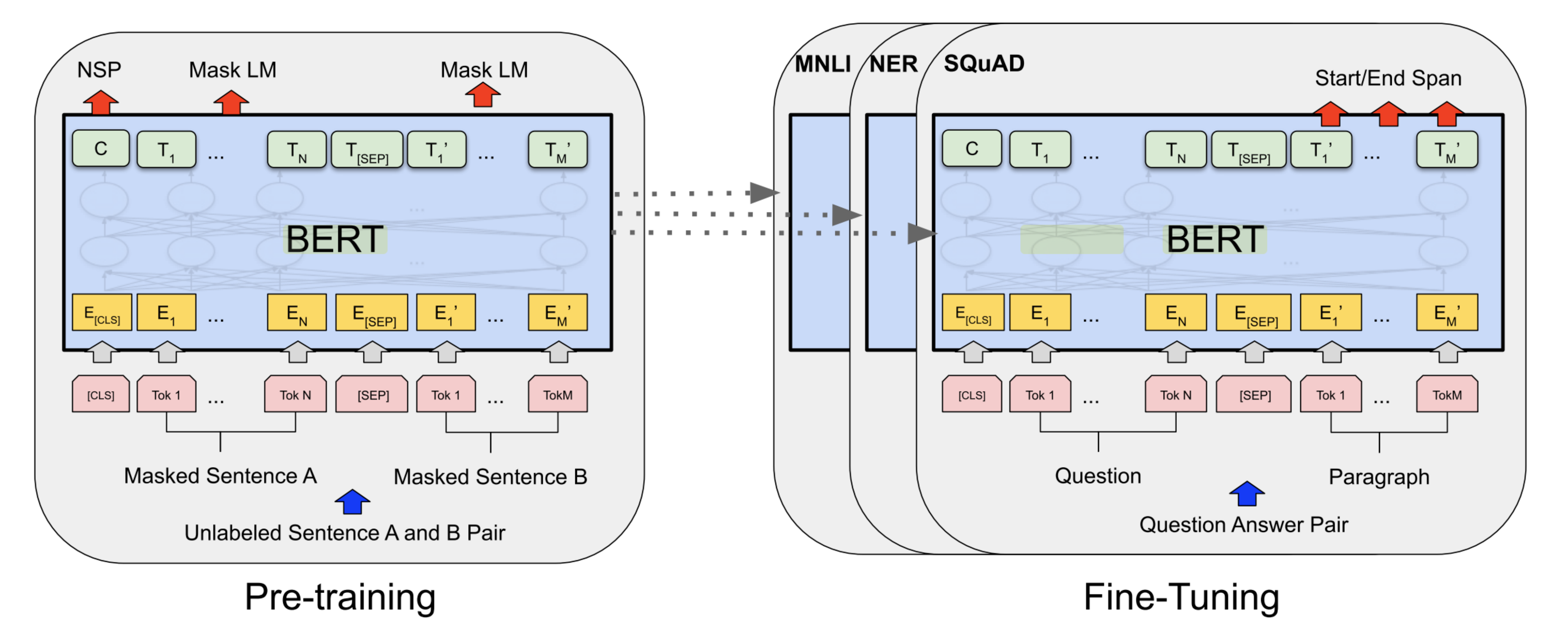
Hopk Selector

there are many samples whose last utterance contains very little information (such as “good”, “ok”), which will cause the selector lose too much useful context information

combine $\widetilde{u}_{i,L-1}, \widetilde{u}_{i,L-2}, \ldots, \widetilde{u}_{i,L-k}$ to get K speaker

CIKM2020 **Speaker-Aware BERT for Multi-Turn Response Selection in Retrieval-Based Chatbots**

1.In-Domain Adaption



CIKM2020 **Speaker-Aware BERT for Multi-Turn Response Selection in Retrieval-Based Chatbots**

2.Speaker Embeddings

Input	[CLS]	How	are	you	[EOU]	[EOT]	Go	to	hold	a	drum	class	[EOU]	Anyone	joins	[EOU]	[EOT]	Sure	[EOU]	[EOT]	[SEP]	Have	fun	in	class	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{How}	E_{are}	E_{you}	$E_{[EOU]}$	$E_{[EOT]}$	E_{Go}	E_{to}	E_{hold}	E_{a}	E_{drum}	E_{class}	$E_{[EOU]}$	E_{Anyone}	E_{joins}	$E_{[EOU]}$	$E_{[EOT]}$	E_{Sure}	$E_{[EOU]}$	$E_{[EOT]}$	$E_{[SEP]}$	E_{Have}	E_{fun}	E_{in}	E_{class}	$E_{[SEP]}$
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	E_{17}	E_{18}	E_{19}	E_{20}	E_{21}	E_{22}	E_{23}	E_{24}	E_{25}
Speaker Embeddings	E_0	E_0	E_0	E_0	E_0	E_0	E_1	E_1	E_1	E_1	E_1	E_1	E_1	E_1	E_1	E_1	E_1	E_0	E_0	E_0	E_0	E_1	E_1	E_1	E_1	E_1

MDFN

arxiv 2020 **Filling the Gap of Utterance-aware and Speaker-aware Representation for Multi-turn Dialogue**

Disadvantages int using of PrLMS

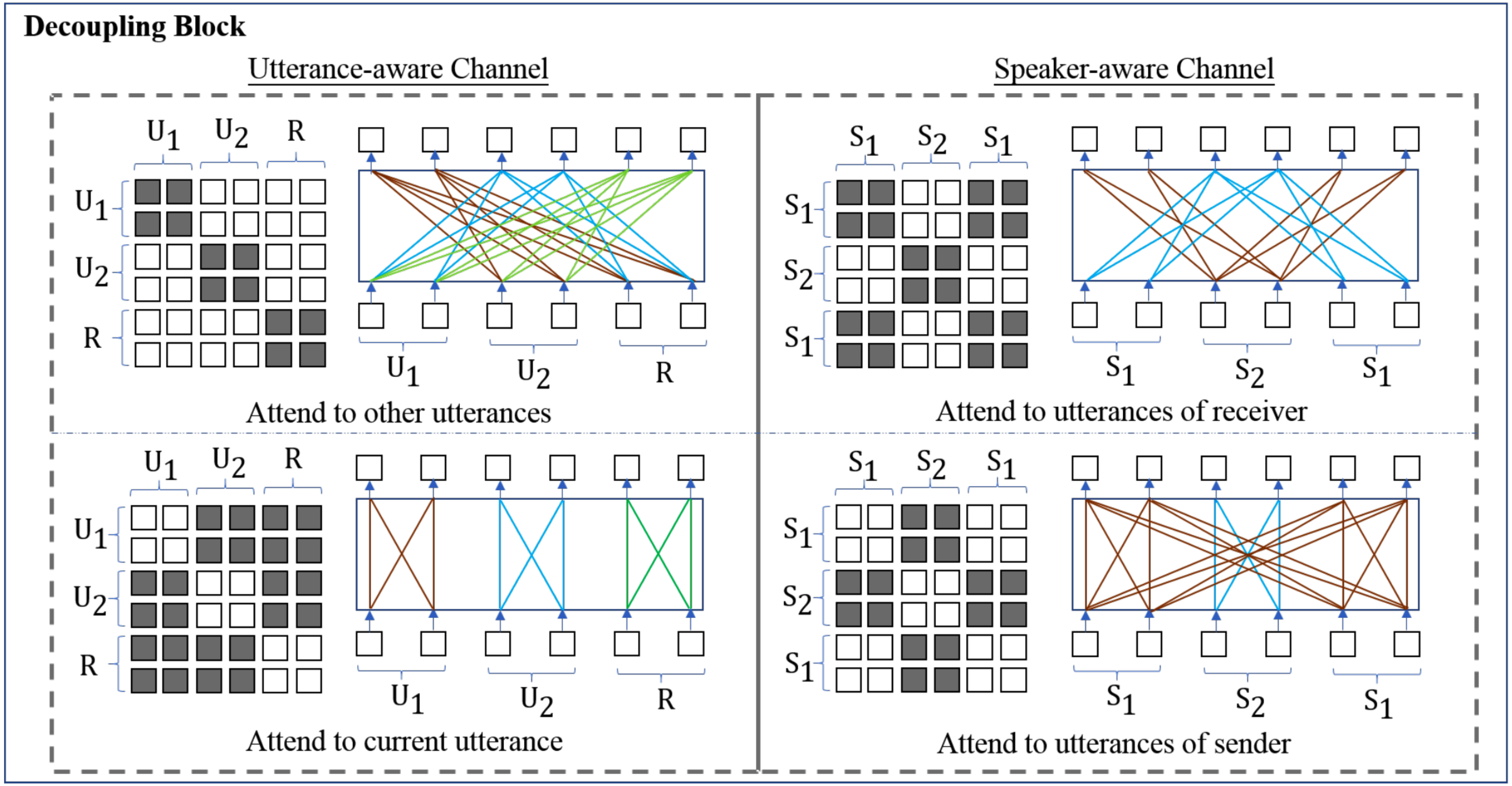
*Simply embedding the token to high-dimensional space cannot faithfully model the additional information, such as positional or turn order information.

*The mechanism of self-attention runs through the whole dialogue, resulting in entangled information that originally belongs to different parts.

MDFN

arxiv 2020

Filling the Gap of Utterance-aware and Speaker-aware Representation for Multi-turn Dialogue



MDFN

arxiv 2020 **Filling the Gap of Utterance-aware and Speaker-aware Representation for Multi-turn Dialogue**

$$M_1[i,j] = \begin{cases} 0, & \text{if } \mathbb{T}_i = \mathbb{T}_j \\ -\infty, & \text{otherwise} \end{cases}$$

$$M_2[i,j] = \begin{cases} 0, & \text{if } \mathbb{T}_i \neq \mathbb{T}_j \\ -\infty, & \text{otherwise} \end{cases}$$

$$M_3[i,j] = \begin{cases} 0, & \text{if } \mathbb{S}_i = \mathbb{S}_j \\ -\infty, & \text{otherwise} \end{cases}$$

$$M_4[i,j] = \begin{cases} 0, & \text{if } \mathbb{S}_i \neq \mathbb{S}_j \\ -\infty, & \text{otherwise} \end{cases}$$

Token Embedding

$$E = [e_1, e_2, \ldots, e_{n_0+\ldots+n_k+n_r}]$$

Channel-aware Information Decoupling

$$C_i = \mathrm{MHSA}(E, M_i), i \in \{1,2,3,4\}$$

$$\{C_k\}_{k=1}^{4} \in \mathbb{R}^{l \times d}$$

MDFN

arxiv 2020 **Filling the Gap of Utterance-aware and Speaker-aware Representation for Multi-turn Dialogue**

Complementary Information Fusing

$$P_1 = G_1(E, C_1, C_2),$$
$$P_2 = G_2(E, C_3, C_4),$$
$$C_u = P_1 \odot C_1 + (1 - P_1) \odot C_2,$$
$$C_s = P_2 \odot C_3 + (1 - P_2) \odot C_4,$$

Gated function

$$\tilde{E}_1 = \mathrm{ReLU}(\mathrm{FC}([E, \bar{E}, E - \bar{E}, E \odot \bar{E}])),$$
$$\tilde{E}_2 = \mathrm{ReLU}(\mathrm{FC}([E, \hat{E}, E - \hat{E}, E \odot \hat{E}])),$$
$$P = \mathrm{Sigmoid}(\mathrm{FC}(([\tilde{E}_1, \tilde{E}_2]))),$$
$$G(E, \bar{E}, \hat{E}) = P,$$

Utterance Representations

$$L_u[i,:] = \underset{\mathbb{T}_j = i}{\mathrm{MaxPooling}}(C_u[j,:]) \in \mathbb{R}^d,$$
$$L_s[i,:] = \underset{\mathbb{T}_j = i}{\mathrm{MaxPooling}}(C_s[j,:]) \in \mathbb{R}^d.$$

Dialogue Representation

$$\overleftarrow{\boldsymbol{h}}_j = \overleftarrow{\mathrm{GRU}}(\overleftarrow{\boldsymbol{h}}_{j-1}, \overleftarrow{\boldsymbol{L}}[j]),$$
$$\overrightarrow{\boldsymbol{h}}_j = \overrightarrow{\mathrm{GRU}}(\overrightarrow{\boldsymbol{h}}_{j-1}, \overrightarrow{\boldsymbol{L}}[j]),$$
$$\boldsymbol{h}_j = [\overleftarrow{\boldsymbol{h}}_j; \overrightarrow{\boldsymbol{h}}_j].$$

$\rightarrow$ **v**

$$\boldsymbol{v} = \mathrm{Tanh}(W[\boldsymbol{v_1}; \boldsymbol{v_2}] + b)$$

THANK YOU!

2020 / 10 / 10