Dialogue Relation Extraction with Document-level Heterogeneous Graph Attention Networks

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, Soujanya Poria

Dialogue Relation Extraction

- Task: Given a dialogue and argument pairs, predict their relations.

S1:	Hey Pheebs.
S2:	Hey!
S1:	Any sign of your **brother**?
S2:	No, but he's always late.
S1:	I thought you only met him once?
S2:	Yeah, I did. I think it sounds y'know big sistery, y'know, 'Frank's always late.'
S1:	Well relax, he'll be here.

	Argument pair	**Trigger**	**Relation type**
R1	(Frank, S2)	brother	per:siblings
R2	(S2, Frank)	brother	per:siblings
R3	(S2, Pheebs)	none	per:alternate_names
R4	(S1, Pheebs)	none	unanswerable

Table 1: A dialogue and its associated instances in DialogRE. S1, S2: anoymized speaker of each utterance.

Dialogue Relation Extraction

- Dataset: DialogRE dataset, which contains 1,788 dialogues and 10,168 relational triples.

DialogRE	
Average dialogue length (in tokens)	225.8
Average # of turns	12.9
Average # of speakers	3.3
Average # of sentences	21.8
Average # of relational instances	4.5
Average # of no-relation instances	1.2

Table 3: Statistics per dialogue of DialogRE.

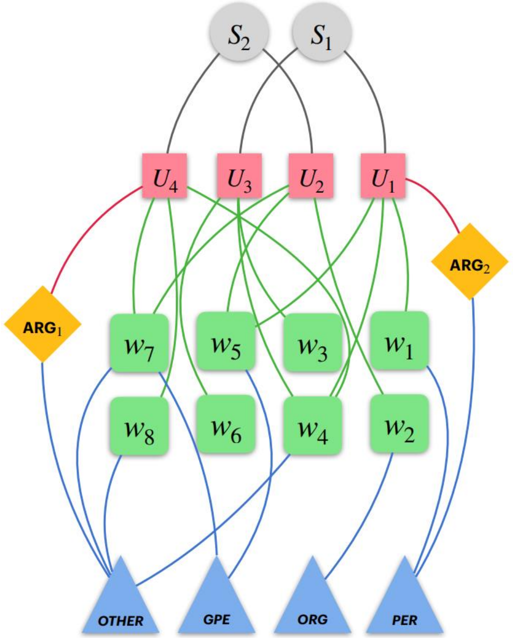
Dialogue Relation Extraction

- DialogueRE: some of the relation types:

ID	Subject	Relation Type	Object	Inverse Relation	TR (%)
1	PER	per:positive_impression	NAME		70.4
2	PER	per:negative_impression	NAME		60.9
3	PER	per:acquaintance	NAME	per:acquaintance	22.2
4	PER	per:alumni	NAME	per:alumni	72.5
5	PER	per:boss	NAME	per:subordinate	58.1
6	PER	per:subordinate	NAME	per:boss	58.1
7	PER	per:client	NAME		50.0
8	PER	per:dates	NAME	per:dates	72.5
9	PER	per:friends	NAME	per:friends	94.7
10	PER	per:girl/boyfriend	NAME	per:girl/boyfriend	86.1
11	PER	per:neighbor	NAME	per:neighbor	71.2
12	PER	per:roommate	NAME	per:roommate	89.9
13	PER	per:children*	NAME	per:parents	85.4
14	PER	per:other_family*	NAME	per:other_family	52.0

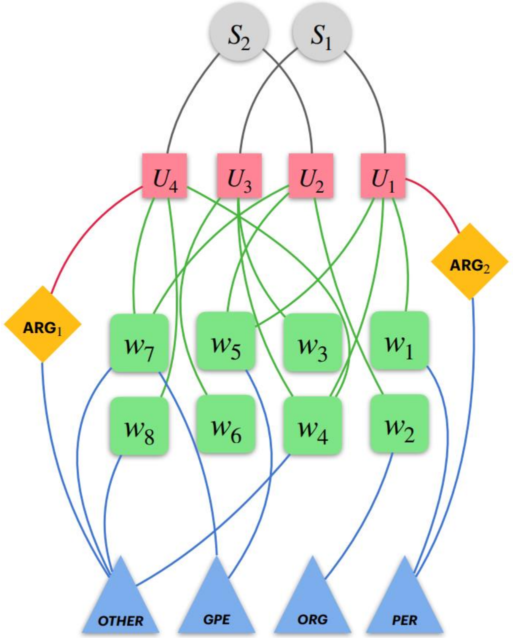
Overview

- This work introduces a graph attention network for DialogueRE.
- They first construct a graph of nodes of different types: utterances, words, entity types, speakers, and arguments.
- They propose a message passing strategy for this hetergenous graph to compute the representations of the nodes.



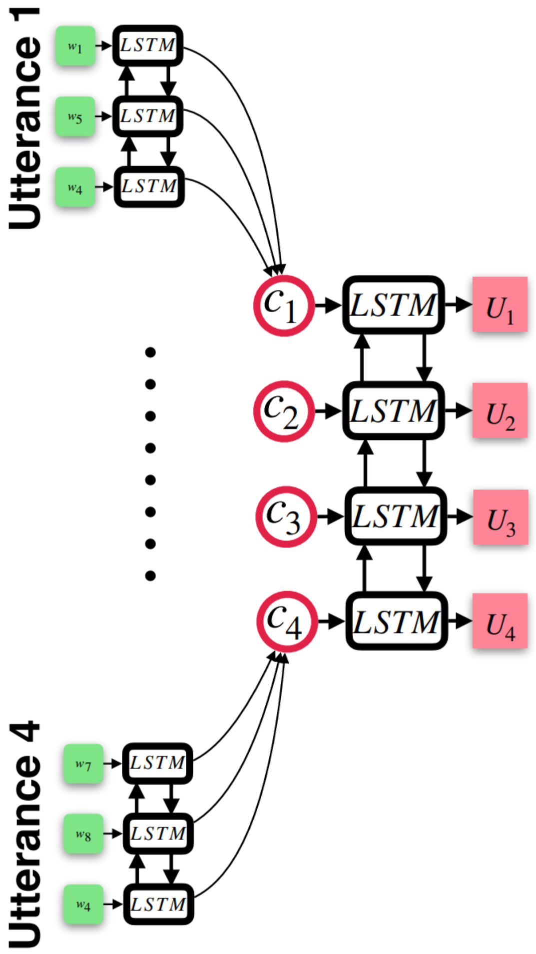
Graph Construction

- Utterances are connected to their constituent words, to the speakers that uttered these uterrances.
- Arguments are connected to the utterances that they appear in, and to their entity types.
- Entity types are connected to constituent words of corresponding arguments.

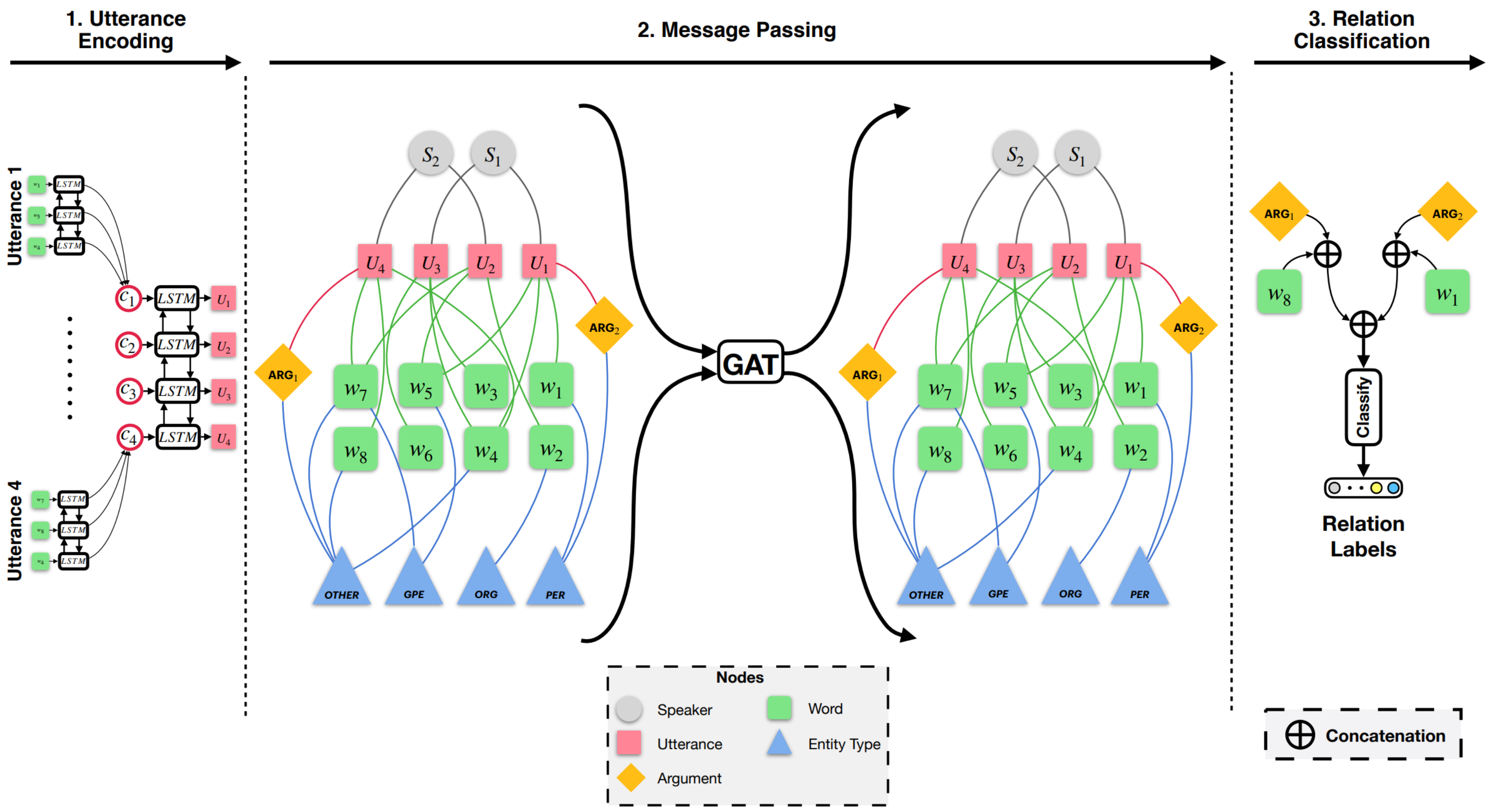


Input encoder

- Glove for word representations.
- LSTMs for utterance representations.
- Randomly-initialized vectors for speaker, argument, and entity type representations.



Meta-Path for Message Passing



Meta-Path for Message Passing

- Updates for a node i w.r.t a neighbor node j:

$$\mathcal{F}(h_i, h_j) = \text{LeakyReLU}(\mathbf{a}^T(\mathbf{W}_i h_i; \mathbf{W}_j h_j; \mathbf{E}_{ij}) \quad (7)$$

$$\alpha_{ij} = \text{softmax}(\mathcal{F}(h_i, h_j)) = \frac{\exp(\mathcal{F}(h_i, h_j))}{\sum_k \exp(\mathcal{F}(h_i, h_k))} \quad (8)$$

$$h_i' = ||_{k=1}^{K} \sigma(\sum_j \alpha_{ij}^k \mathbf{W}_q^k h_j) \quad (9)$$

- $\mathbf{E}_{ij}$ is the randomly-initialized embedding vector, that depending on the type of edge between node i and node j (utterance-word, utterance-argument, …).

Meta-Path for Message Passing

- Updates for the nodes are *not done simultaneously*.
- For hetergenous graphs, Meta-path (2011) has been used as a general structure to capture different semantics in the graphs.
- They propose a particular meta-path for the updates for the nodes at each layer of GAT network. The order of the meta-path is validated by their ablation study.

Utterances -> {words, speakers, arguments} -> entity types -> {words, speakers, arguments} -> Utterances -> {words, speakers, arguments}

Relation Classifier

- From the output representations of the multi-layer message passing with GAT, select the argument nodes τ_x , τ_y and their constitutent word nodes e_x, e_y to make prediction on the relation.

$$
\begin{aligned}
e'_x &= [\mathrm{maxpool}(\tau_x); \mathrm{maxpool}(e_x)] \\
e'_y &= [\mathrm{maxpool}(\tau_y); \mathrm{maxpool}(e_y)] \\
e' &= [e'_x; e'_y] \\
P(r|e_x, e_y) &= \sigma(\mathbf{W}_e e' + b_e)_r
\end{aligned}
$$

Results

Model	#params	Dev $F1$	Dev $F1_c$	Test $F1$	Test $F1_c$
Majority (Yu et al. 2020)	-	38.9	38.7	35.8	35.8
CNN (Yu et al. 2020)	-	46.1	43.7	48.0	45.0
LSTM (Yu et al. 2020)	-	46.7	44.2	47.4	44.9
BiLSTM (Yu et al. 2020)	4.1M	48.1	44.3	48.6	45.0
AGGCN (Guo, Zhang, and Lu 2019)	3.7M	46.6	40.5	46.2	39.5
LSR (Nan et al. 2020)	20.5M	44.5	-	44.4	-
DHGAT(Ours)	4.0M	**57.7**	**52.7**	**56.1**	**50.7**