Semantics-Aware Dual Graph Convolutional Networks for Argument Pair Extraction

Minzhao Guan[1], Zhixun Qiu[1], Fenghuan Li[2*], Yun Xue[1*]

[1]School of Electronics and Information Engineering, South China Normal University

[2]School of Computer Science and Technology, Guangdong University of Technology

LREC-COLING 2024

Task

Argument Pair Extraction (APE)

- Extracting interactive argument pairs from two passages of a discussion.
- In Figure 1, a review text and its accompanying response text are divided into argumentative and non-argumentative segments at sentence level. The arguments within the review can be paired with those in the response based on the topics they address, creating interactive argument pairs.

Arg	Review	Arg
Non-Arg	1. This paper proposes a new Q learning algorithm framework: maxmin Q-learning, to address the overestimation bias issue of Q learning. ... 5. I have two main concerns for this paper:	**Non-Arg**
α_1^{rev}	6. 1)When is your algorithm useful? 7. What's your criterion of picking the hyper-parameters (e.g. number of Q functions you want to learn).	α_1^{rev}
α_2^{rev}	8. 2) Comparison to more intriguing way for jointly update of multiple Q functions, like soft Q learning.	α_2^{rev}
...	...	...
Non-Arg	19. Overall, I believe the idea of the paper is novel and interesting, but further improvements should be added in order to improve the score the paper.	**Non-Arg**

Arg	Rebuttal	Arg
Non-Arg	1. We appreciate your feedback.	**Non-Arg**
α_1^{rep}	2. For the first concern, you are right, we cannot know for an unknown environment whether overestimation or underestimation will help. ... 13. We are not exactly sure what you mean by your comment that "a drift for Q learning (e.g.) has no effect on our policy". ... 18. If c is random, could you clarify further what you mean here?	α_1^{rep}
α_2^{rep}	19. You are right that our Maxmin Q-learning is a joint update scheme for different Q functions, and one of our contributions is that we provide a convergence proof for such a framework under reasonable assumptions. ... 29. N Q functions are learned with Q-learning (rather than say with the Maxmin update). ... 38. If you can further clarify why we should compare to SQL, we would be happy to respond further.	α_2^{rep}

AP1

AP2

Figure1： An example of APE.

Motivation

- Co-occurring words are crucial for pairing arguments, but relying solely on their count can result in incorrect pairings. Therefore, in addition to the number of co-occurring words, the semantic relevance of each argument must be taken into account.
- In Figure 1, the underlined word 'Q learning' appears in both the 'review' passage and the 'rebuttal', but arguments fail to form a pair. Relying solely on co-occurring words has limitations, while incorporating topic information helps in argument pairing, as shown by the green and blue passages relating to parameters and algorithm comparison of 'Q learning'.

Arg	Review	Arg
Non-Arg	1. This paper proposes a new Q learning algorithm framework: maxmin Q-learning, to address the overestimation bias issue of Q learning. ... 5. I have two main concerns for this paper:	Non-Arg
α_1^{rev}	6. 1)When is your algorithm useful? 7. What's your criterion of picking the hyper-parameters (e.g. number of Q functions you want to learn).	α_1^{rev}
α_2^{rev}	8. 2) Comparison to more intriguing way for jointly update of multiple Q functions, like soft Q learning.	α_2^{rev}
...	...	...
Non-Arg	19. Overall, I believe the idea of the paper is novel and interesting, but further improvements should be added in order to improve the score the paper.	Non-Arg

AP1

AP2

Arg	Rebuttal	Arg
Non-Arg	1. We appreciate your feedback.	Non-Arg
α_1^{rep}	2. For the first concern, you are right, we cannot know for an unknown environment whether overestimation or underestimation will help. ... 13. We are not exactly sure what you mean by your comment that "a drift for Q learning (e.g.) has no effect on our policy". ... 18. If c is random, could you clarify further what you mean here?	α_1^{rep}
α_2^{rep}	19. You are right that our Maxmin Q-learning is a joint update scheme for different Q functions, and one of our contributions is that we provide a convergence proof for such a framework under reasonable assumptions. ... 29. N Q functions are learned with Q-learning (rather than say with the Maxmin update). ... 38. If you can further clarify why we should compare to SQL, we would be happy to respond further.	α_2^{rep}

Figure1：An example of APE.

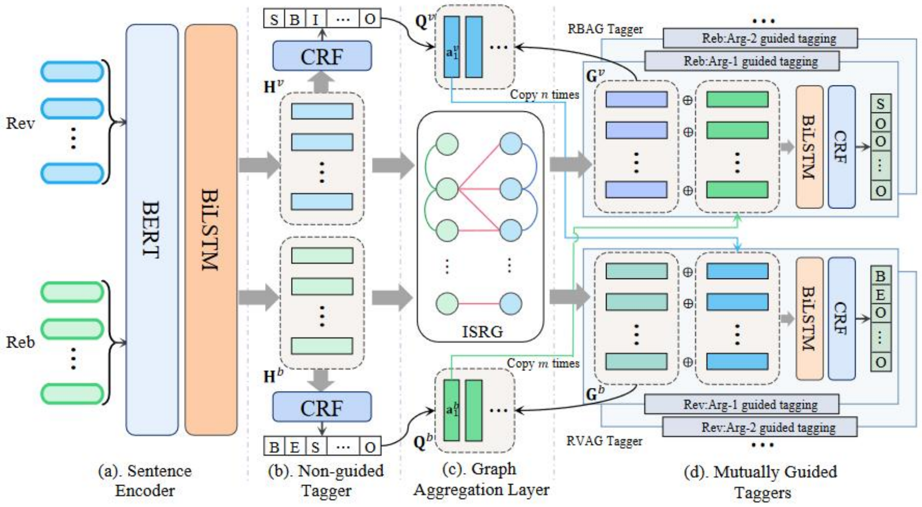

Figure2：The approach of [Bao et al., 2021].

Contribution

- We devise a Rouge-guided Co-occurring word Graph Convolutional Network (RCGCN), which deals with the semantic relevance of argument pairs containing co-occurring words.
- A topic-related graph is constructed in line with the topic probability distribution of neural topic model and topic embeddings, which is further encoded by GCN for argument topic interaction. Lastly, a gating unit is performed to fuse the co-occurring word information and topic information for APE.
- Experiments are carried out on publicly available benchmarks to evaluate the working performance of our model. Experimental results indicate that the proposed model outperforms the state-of-the-art(SOTA) by 6.56% in F1 score. More tests are also conducted to validate the effectiveness of components in the proposed model.

Task Definition

- Input:
  - A review passage $D^{rev} = \{s_1^{rev}, \cdots, s_n^{rev}\}$ and a rebuttal passage $D^{reb} = \{s_1^{reb}, \cdots, s_n^{reb}\}$.
- Ouput:
  - A review argument spans set $X^{rev} = \{\alpha_1^{rev}, \alpha_2^{rev}, \cdots\}$ and a rebuttal argument spans set $X^{reb} = \{\alpha_1^{reb}, \alpha_2^{reb}, \cdots\}$.
  - A set of interactive argument pairs $P = \{p_1, p_2, \cdots\}$, where $p_i \in X^{rev} \times X^{reb}$ is an interactive argument pair.

Arg	Review	Arg
Non-Arg	1. This paper proposes a new Q learning algorithm framework: maxmin Q-learning, to address the overestimation bias issue of Q learning. ... 5. I have two main concerns for this paper:	Non-Arg
α_1^{rev}	6. 1)When is your algorithm useful? 7. What's your criterion of picking the hyper-parameters (e.g. number of Q functions you want to learn).	α_1^{rev}
α_2^{rev}	8. 2) Comparison to more intriguing way for jointly update of multiple Q functions, like soft Q learning.	α_2^{rev}
...	...	...
Non-Arg	19. Overall, I believe the idea of the paper is novel and interesting, but further improvements should be added in order to improve the score the paper.	Non-Arg

AP1

Arg	Rebuttal	Arg
Non-Arg	1. We appreciate your feedback.	Non-Arg
α_1^{rep}	2. For the first concern, you are right, we cannot know for an unknown environment whether overestimation or underestimation will help. ... 13. We are not exactly sure what you mean by your comment that "a drift for Q learning (e.g.) has no effect on our policy". ... 18. If c is random, could you clarify further what you mean here?	α_1^{rep}
α_2^{rep}	19. You are right that our Maxmin Q-learning is a joint update scheme for different Q functions, and one of our contributions is that we provide a convergence proof for such a framework under reasonable assumptions. ... 29. N Q functions are learned with Q-learning (rather than say with the Maxmin update). ... 38. If you can further clarify why we should compare to SQL, we would be happy to respond further.	α_2^{rep}

AP2

Figure1：An example of APE.

Our Approach

- We propose a Semantics-Aware Dual Graph Convolutional Networks (SADGCN).
- Our approach aims to determine the relationship between argument pairs by using co-occurring word information and topic information.

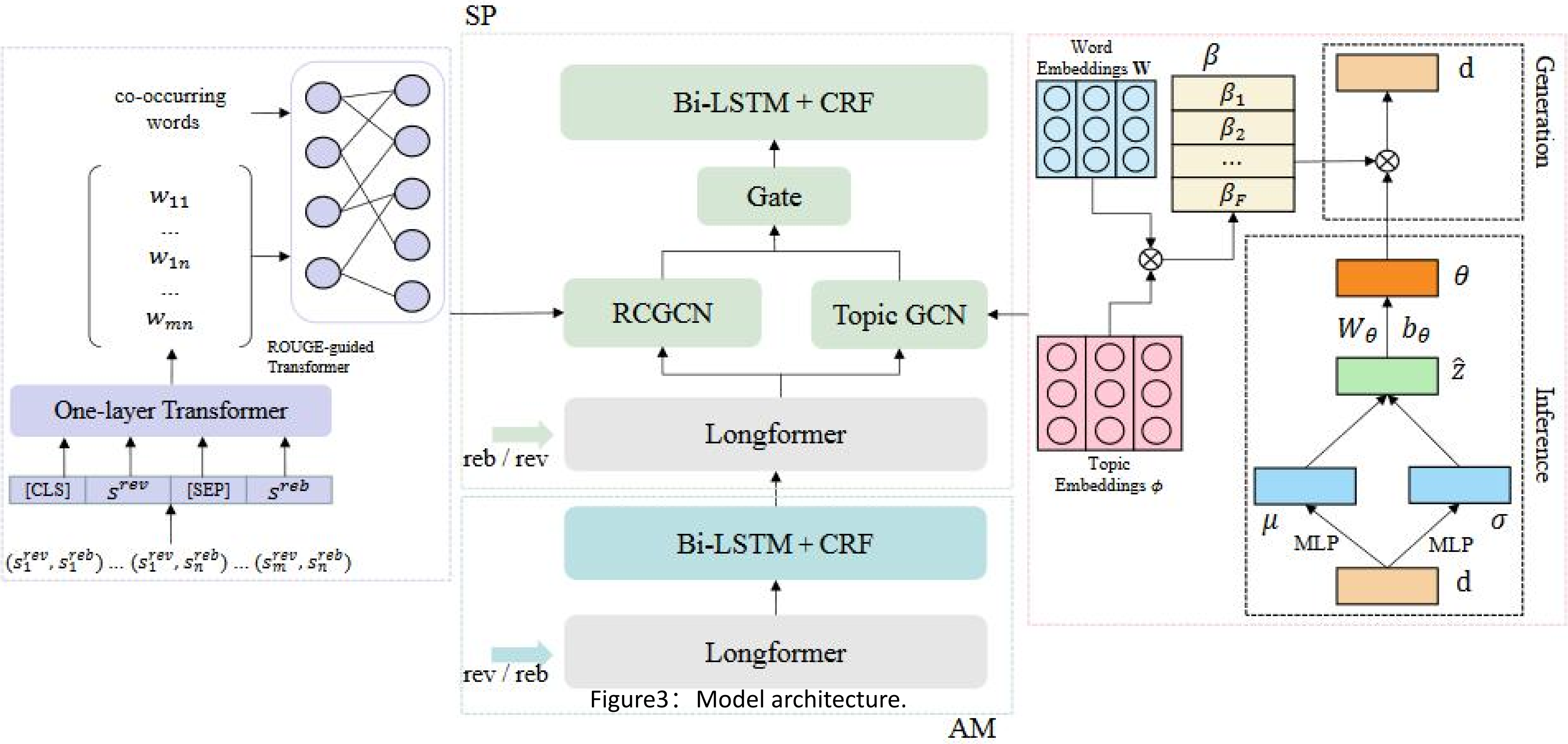

Figure3：Model architecture.

Our Approach

- Argument Mining Input
    - We use a special token "[AM]" as the query for argument min word information and topic information. During Argument Mining process, the inputs are the concatenation of "[AM]" and the document, i.e.,
    $$I^{AM} = ([s], [AM], [/s], [s], s_1, \cdots, s_{num}, [/s])$$
- Sentence Pairing Input
    - In the Argument Mining process, we obtained $X^{rev} = \{\alpha_1^{rev}, \cdots\}$ and $X^{reb} = \{\alpha_1^{reb}, \cdots\}$. Taking each argument from X^{rev}/X^{reb} as the query for Sentence Pairing, we concatenate $\alpha_t^{rev}/\alpha_t^{reb}$ with D^{reb}/D^{rev}, to generate the input sequence, which are:
    $$I_{rev\to reb,t}^{SP} = ([s], \alpha_t^{rev}, [/s], [s], s_1^{reb}, \cdots, s_m^{reb}, [/s])$$
    $$I_{reb\to rev,t}^{SP} = ([s], \alpha_t^{reb}, [/s], [s], s_1^{rev}, \cdots, s_n^{rev}, [/s])$$
- Afterwards, the input sequences are fed into the Longformer model to extract the hidden representations of individual tokens.

Our Approach

- Argument Mining
  - The input data is processed by Longformer model, in order to derive the sentence representations. Subsequently, these sentence representations are fed into a Bi-LSTM+CRF architecture to yield the final output. Mathematically, the process of argument mining (AM) can be expressed as follows:

$$Y^{rev/reb} = CRF(Bi - LSTM(Longformer(I^{AM})))$$

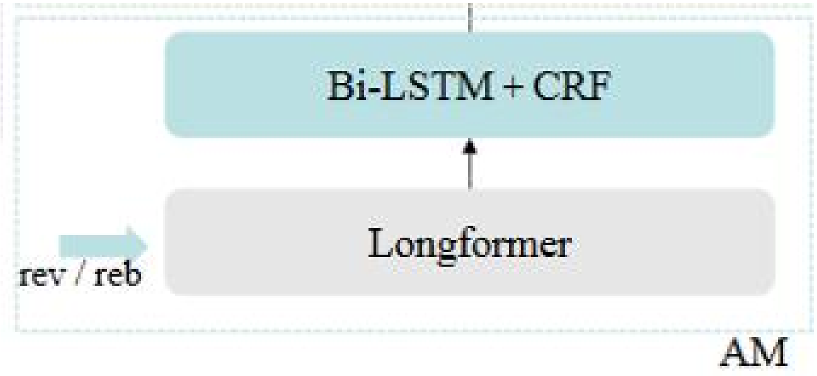

Figure4： AM process of Model .

Our Approach

- **Sentence Pairing**
  - At this stage, one or more arguments are derived from the argument mining task, which is taken as queries for SP to obtain the sentence representations. To facilitate the description, we tend to present the search for rebuttal argument in D^{reb} using review argument, i.e., $\mathrm{rev} \rightarrow \mathrm{reb}$. The process of $\mathrm{reb} \rightarrow \mathrm{rev}$ is implemented in the same manner but in a reversed direction. We take the union set of both directions as the final extraction results.
  - **RCGCN**
    - **ROT**
      - We use a Rouge-guided One-layer Transformer (ROT) to analyze the semantic relevance between related sentences. Each sentence from D^{rev} is matched with each sentence from D^{reb} to create $\mathrm{n} \times \mathrm{m}$ pairs of sentences. The Rouge-2 score is then calculated for each pair.

$$z_{rot} = Transformer([CLS]s^{rev}[SEP]s^{reb})$$

      - The optimization of the ROT is performed by minimizing the mean-squared error between precision and recall of ROUGE-2, which is given by:

$$\widehat{R}(s^{rev}, s^{reb}) = MLP(z_{rot}([CLS]))$$
$$L_R = ||\widehat{R}(s^{rev}, s^{reb}) - R(s^{rev}, s^{reb})||_2^2 - \lambda_R||\Delta\theta||_2^2$$

Our Approach

- **Co-occurring Word Graph Construction**
  - A graph is constructed to model the co-occurring word relation of the mined argument with the other document. The weights at the diagonal positions are set to 1 and other positions to 0 for A^{co}. Only if the i^{th} sentence of argument shares an oc-curring word with the j^{th} sentence of passage, can the edge between the two sentences be established in A^{co}. The weight value is computed as:

$$R_i = ROT([CLS]s_i[SEP])$$
$$D_{ij}^{co} = |R_i([CLS]) - R_j([CLS])|_1$$
$$A_{ij}^{co} = 1 - \frac{D_{ij}^{co} - min(D^{co})}{max(D^{co}) - min(D^{co})}$$

- **Co-occurring Word Graph Convolution**
  - The graph convolutional network (GCN) is typically used for information exchange between nodes in a graph, which are as follow:

$$H_0^{co} = [H_t^{rev}; H^{reb}]$$
$$H_{l+1}^{co} = \sigma(A^{co} H_l^{co} W_l^{co})$$

Our Approach

- **Topic GCN**
  - **NTM**
    - Neural topic model contains inference and generation. Formally, we build an inference network to infer the document-topic distribution θ. A bag-of-words representation d is sent to two neural networks to generate $\mu(d)$ and $\sigma(d)$, together with the parameterization of $q(z|d) = \mathrm{N}\ (\mu(d), \sigma^2(d))$. We then re-parameterize the $q(z|d)$ to extract $\hat{z} = \mu(d) + \varepsilon \cdot \sigma(d)$. The topic distribution θ is expressed as:

$$\theta = softmax(W_\theta \hat{z} + b_\theta)$$

$$\beta = softmax(\frac{\Phi \cdot W^T}{\sqrt{M}})$$

$$d = \theta \cdot \beta$$

    - The loss function of the neural topic model is given by:

$$L_{NTM} = KL[q(z|d)||p(z)] - E_{q(z|d)}[logp(d|\theta, \beta)]$$

Our Approach

- **Topic Graph Construction**
  - The topic embedding of the NTM is used to construct a topic graph, which models the topic relation between the mined argument and the other document. Similar to the construction of the co-occurring word graph, for a z-sentence argument α_t^{rev}, a topic-relevant matrix A^{topic} is established.

$$e^{topic} = \theta \cdot \Phi$$

$$D_{ij}^{topic} = |e_i^{topic} - e_j^{topic}|_2$$

$$W_{ij}^{topic} = 1 - \frac{D_{ij}^{topic} - min(D^{topic})}{max(D^{topic}) - min(D^{topic})}$$

$$A_i^{topic} = topk(W_i^{topic})$$

- **Topic Graph Convolution**
  - Likewise, GCN is employed for topic information interaction between nodes.

$$H_0^{topic} = [H_t^{rev}; H^{reb}]$$

$$H_{l+1}^{topic} = \sigma(A^{topic} H_l^{topic} W_l^{topic})$$

Our Approach

- **Gated Fusion**
  - The co-occurring word graph representation H_{reb}^{co} and the topic graph representation H_{reb}^{topic} are integrated via gating mechanism:

$$\alpha^{gate} = \sigma(W_1^{gate} \cdot H_{reb}^{co} + W_2^{gate} \cdot H_{reb}^{topic})$$
$$H^{gate} = \alpha^{gate} \cdot H_{reb}^{co} + (1 - \alpha^{gate}) \cdot H_{reb}^{topic}$$

  - The outcome H^{gate} is fed into LSTM to obtain its contextual representation, which is further sent to the CRF sequence tagger.

Our Approach

- **Training**
  - Three losses are added up as the training objective of our model:

$$L_{AM} = logp(\widehat{Y}^{rev}|D^{rev}) + logp(\widehat{Y}^{reb}|D^{reb})$$

$$L_{SP} = \sum_i logp(\widehat{Y}^{pair}_{rev\to reb,i}|D^{reb}, X^{rev}) + \sum_i logp(\widehat{Y}^{pair}_{reb\to rev,i}|D^{rev}, X^{reb})$$

$$L = L_{AM} + L_{NTM} + L_{Sp}$$

- **Inference**
  - In the inference phase, prediction results of two reversed directions are fused for sentence pairing. The argument from D^{reb} that is in pair with α_t^{rev} is extracted. The rebuttal argument span is written as $X^{reb}_{\mathrm{rev\to reb}} = \{\alpha_1^{reb}, \cdots\}$. Accordingly, the argument pair set derived from $Y^{\mathrm{pair}}_{\mathrm{rev\to reb,t}}$ can be $P_{\mathrm{rev\to reb,t}} = \{[\alpha_t^{rev}, \alpha_1^{reb}], ...\}$. Following this process, all argument pairs in the direction of rev → reb can be predicted as $P_{\mathrm{rev\to reb}} = \cup_t P_{\mathrm{rev\to reb,t}}$. In the same way, all argument pairs in the direction of reb → rev can also be maintained. We shall take the predictions of both directions as the final argument pair prediction, i.e., $P = P_{\mathrm{rev\to reb}} \cup P_{\mathrm{reb\to rev}}$.

Experiments

- We carry out our experiments on the Review-Rebuttal(RR) dataset (Cheng et al., 2020).
- Under the condition that our model is comparable with MRC-APE in AM task, the proposed model substantially outperforms MRC-APE in APE task. The performance gaps on F1 against MRC-APE are 6.56% and 4.68% on RR-submission and RR-passage.
- With the elimination of error propagation caused by AM, our model outperforms MRC-APE by 4.30% on RR-submission and 4.14% on RR-passage of F1 score.
- The results of these experiments indicate that LLMs exhibit inadequate performance when it comes to address-ing the task at hand

Data	Method	Argument Mining			Sentence Pairing			Argument Pair Extraction		
		Pre.	Rec.	F_1	Pre.	Rec.	F_1	Pre.	Rec.	F_1
RR-submission	PL-H-LSTM-CRF	67.63	68.51	68.06	50.05	47.15	48.56	19.86	19.94	19.90
	MT-H-LSTM-CRF	70.09	70.14	70.12	53.44	42.71	47.48	26.69	26.24	26.46
	MLMC	69.53	**73.27**	71.35	60.81	47.14	53.11	37.15	29.38	32.81
	MGF	70.40	71.87	71.13	44.99	51.94	$48.22^\natural$	34.23	34.57	34.40
	MRC-APE	71.83	73.05	72.43	56.80	59.58	$58.16^\natural$	41.83	38.17	39.92
	GPT-3.5	57.83	63.31	60.45	65.64	50.57	57.13	25.02	28.57	26.68
	GPT-4	67.38	69.71	68.53	**67.33**	55.63	60.92	37.63	39.12	38.36
	Our SADGCN	**73.18**	72.88	**73.03**	59.16	**66.15**	**62.46**	**45.67**	**47.32**	**46.48**
RR-passage	PL-H-LSTM-CRF	73.10	67.65	70.27	51.34	42.08	46.25	21.24	19.30	20.22
	MT-H-LSTM-CRF	71.85	71.01	71.43	54.28	43.24	48.13	30.08	29.55	29.81
	MLMC	66.79	72.17	69.38	**61.29**	45.94	52.52	40.27	29.53	34.07
	MGF	73.62	70.88	72.22	42.45	54.00	$47.53^\natural$	38.03	35.68	36.82
	MRC-APE	**76.39**	70.62	73.39	52.22	63.11	$57.15^\natural$	37.70	44.00	40.61
	GPT-3.5	64.40	66.43	65.40	58.34	50.53	54.15	28.90	30.23	29.55
	GPT-4	68.52	71.75	70.10	59.43	56.86	58.12	38.38	40.71	39.51
	Our SADGCN	73.31	**73.69**	**73.50**	56.21	**67.37**	**61.29**	**43.25**	**47.53**	**45.29**

Experiments

- **Ablation Study**
  - The most significant modules for our model are co-occurring word GCN and topic GCN.
  - The application of ROT benefits the exploiting of semantic relevance in APE.

Models	Argument Pair Extraction		
	Pre.	Rec.	F_1
Our SADGCN	**45.67**	**47.32**	**46.48**
W/o ROT weight	45.30	43.08	44.16
W/o RCGCN	40.65	44.03	42.27
W/o Topic GCN	45.54	40.27	42.74
W/o RCGCN & Topic GCN	36.61	38.41	37.25
W/o $D^{reb} \rightarrow D^{rev}$	45.87	41.47	43.56
W/o $D^{rev} \rightarrow D^{reb}$	41.48	39.03	40.22

- **Execution Speed Comparison**
  - The execution speed of LLMs tends to be suboptimal because they encompass a substantially larger quantity of parameters.
  - Compared to MRC models that also utilize Longformer encoding, our model engages in more interactions during the matching process. This leads to a marginally slower execution speed, but concurrently, superior performance on F1-score of our model.

Dataset	Method	Execution Speed
RR-submission	PL-H-LSTM-CRF	1m12s
	MT-H-LSTM-CRF	1m03s
	MLMC	2m13s
	MGF	50s
	MRC-APE	3m21s
	GPT-3.5	53m
	GPT-4	48m
	Our SADGCN	5m25s
RR-passage	PL-H-LSTM-CRF	1m13s
	MT-H-LSTM-CRF	1m03s
	MLMC	2m14s
	MGF	52s
	MRC-APE	3m24s
	GPT-3.5	51m
	GPT-4	47m
	Our SADGCN	5m31s

Experiments

- **Effectiveness of Co-occurring Word Graph and Topic Graph**
  - In D^{rev} , the argument span set is $X^{rev} = \{\alpha_1^{rev}, \alpha_2^{rev}, \alpha_3^{rev}, \alpha_4^{rev}\} = \{(5, 6), (7, 7), (8, 13), (14, 17)\}$. Similarly, the argument span set in D^{reb} is $X^{reb} = \{\alpha_1^{reb}, \alpha_2^{reb}\} = \{(1, 17), (18, 37)\}$. There are four argument pairs in this example: $\mathrm{P} = \{p_1, p_2, p_3, p_4\} = \{[\alpha_1^{rev}, \alpha_1^{reb}], [\alpha_2^{rev}, \alpha_2^{reb}], [\alpha_3^{rev}, \alpha_1^{reb}], [\alpha_4^{rev}, \alpha_2^{reb}]\}$.
  - In the co-occurring word graph, sent-6 in the α_1^{rev} is isolated, while sent-7 is connected to six sentences, with only one belonging to α_1^{reb}. However, in the topic graph, both sent-6 and sent-7 are linked to sentences from α_1^{reb}. Sent-7 captures the co-occurring word 'Q functions'. The discussion on 'Q functions' and 'Q learning' in α_2^{reb} relies solely on co-occurring word information. Conversely, in α_2^{rev}, connections with sentences from α_2^{reb} exist in the co-occurring word graph, while connections with sentences from α_1^{reb} are established in the topic graph. Our model integrates topic and co-occurring word information using a gating mechanism to enhance information transfer between sentences.

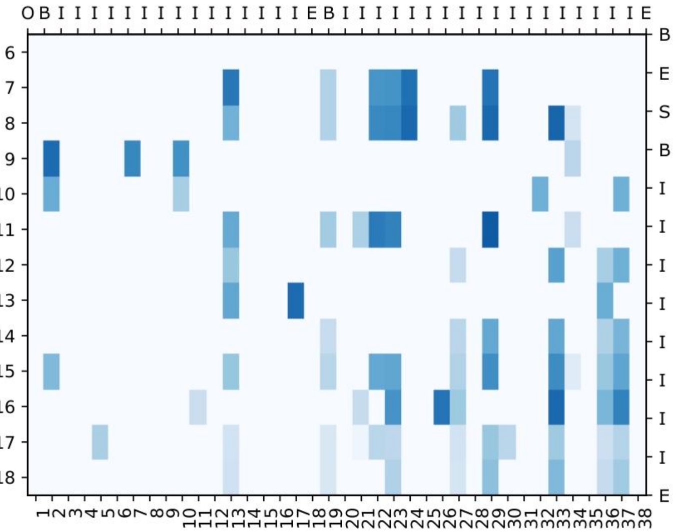

Figure5： Co-occurring word graph.

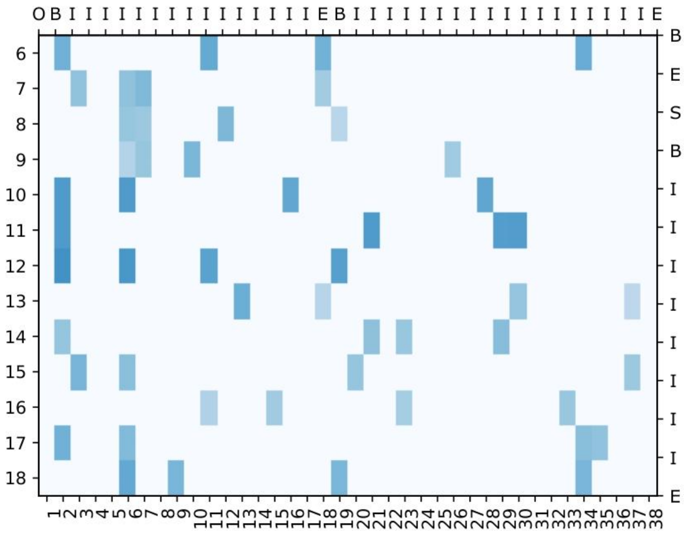

Figure6： Topic graph.

Conclusion

- In this work, a SADGCN model is developed to improve the APE task. The model incorporates co-occurring words and topic information to enhance the reliability of argument pairing.

- By considering the lexical and semantic relevance of arguments, the built RCGCN mitigates unreliable pairings caused by the number of co-occurring words. Additionally, a topic graph characterizes sentence relations of the same topic, enabling deeper sentence comprehension and reducing reliance on co-occurring words alone.

- The integration of these two types of information facilitates the extraction of argument pairs, resulting in SOTA performance on the benchmark dataset.

Limitation

- Based on the empirical study, our model accurately extracts only 40% of arguments consisting of more than 10 sentences. This could be because AM is viewed as a task focused on annotating sentence-level sequences, making it difficult to differentiate and identify the diverse argument spans.

- For another, errors generated in AM can cause the unreliability of SP results. In our work, a minor focus is to eliminate the issue of error propagation. Comparing the working performances of AM and SP, error propagation results in performance degradation of at least 15%.

- In general, our model is less effective in dealing with arguments of complicated sentences. Besides, the mitigation of error propagation is still in suspense, which can be addressed in future work.

Thank you!