Multi-fusion Recurrent Network for Argument Pair Extraction

Naixu He, Qingfeng Chen(✉), Qian Yu, and Zongzhao Han

School of Computer, Electronics and Information, Guangxi University, Nanning, China
2013301014@st.gxu.edu.cn, qingfeng@gxu.edu.cn

Abstract. Argument Pair Extraction (APE) is an extension of argument mining that focuses on identifying argument pairs from two passages that have an intrinsic interaction, such as peer review and rebuttal. Existing studies have divided this task into separate subtasks for argument mining and sentence relation classification, but they overlook the connection between the two subtasks, leading to the accumulation of errors in argument pair extraction. To address this issue, we propose the **M**ulti-fusion **C**ross-update **R**ecurrent **N**etwork (MCRN), which includes two cross-updated units: an argument mining unit and a sentence pairing unit. Specifically, we cross-update the sentence representations of both units to learn the interaction between them, allowing the acquired sentence features to contain both argumentation and sentence relation information. We also designed a recurrent structure to iteratively learn these two units, which improves the utilization of pre-trained features. To evaluate the performance of the model, we conducted extensive experiments on benchmark datasets, which demonstrated that MCRN significantly improves the APE task.

Keywords: argument pair extraction · argument mining · sentence paring · recurrent network

1 Introduction

Argument mining has significant applications in various fields, such as legal document analysis, student writing guidance, and sentiment analysis [12]. This task aims to extract structured argumentative inference components from unstructured text. Exiting studies can be divided into two categories: monological argumentation and dialogical argumentation. Monological argumentation identifies argumentative discourse structures in documents where there is only one speaker, such as persuasive essays [20,21], Wikipedia [13] and legal documents [18,19], while dialogical argumentation focuses on pairs of arguments with internal connections, which is particularly relevant in interactive texts such as online debates [10,22].

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14259, pp. 103–114, 2023.
https://doi.org/10.1007/978-3-031-44223-0_9

Argument pair extraction (APE), proposed by Cheng et al. [5], is a part of dialogical argumentation, which aims to extract argument pairs with interactivity from two argumentative passages. Argument pair extraction requires obtaining arguments in two documents and then composing the corresponding arguments into argument pairs. Early approches relied on a pipeline approach to solve the problem. Arguments are obtained by sequence labeling, and then a binary classification task is used to determine whether the arguments can form argument pairs. However, this overlooks the connection between the two tasks, which can mutually reinforce each other.

To exploit this information, recent studies have proposed some joint training models [1,6], which allow downstream tasks to obtain both sequence annotation and pair extraction results simultaneously.

Despite these advances, the APE still presents some challenges. Sequence labeling and relation classification task of APE are not two independent subtasks, and they actually depend on each other. If we want to identify argument pairs, we need to consider not only argument information but also the relation between arguments. Therefore, a significant challenge is to find appropriate ways to mutually reinforce these two tasks to improve the accuracy of predictions. Unlike most natural language processing tasks, the APE focuses on learning sentence representations with contextual information rather than word vectors or entity representations. And the learning of sentence vectors for specific downstream task is more challenging. An argument usually consists of several adjacent sentences. When determining whether a sentence belongs to an argument, we need to consider the current sentence and its neighbouring sentences only. Too much information may have a negative effect on argument prediction. How to properly introduce adjacency feature to facilitate argument extraction is another challenge.

This paper proposes a novel approach called the **M**ulti-fusion **C**ross-update **R**ecurrent **N**etwork (MCRN) to address the challenges associated with the argument pair extraction (APE). The proposed method employs a cross-updated argument mining unit and sentence pairing unit to simultaneously extract arguments and sentence relations. The cross-update mechanism allows for mutual reinforcement between argument mining and sentence relation learning. A local encoder is designed in the argument mining unit to extract argument features without introducing redundant noise. To enhance the semantic information contained in the sentence vectors, a recurrent network is incorporated to repeatedly fuse the BERT representations. Experimental results on benchmark datasets demonstrate MCRN significantly outperforms baseline methods.

2 Related Work

Argument mining aims to extract arguments from the text in order to provide structured data for the computational model of the argumentation and reasoning engine. A large amount of research in argument mining focuses on monological argumentation, such as argument component identification [14,17], argument retrieval [7,8], argument quality assessment [24], etc.

As the ultimate purpose of arguments is to be used in debates, modeling dialogical argumentation has attracted increasing attention [2,16]. Wei et al. [26] collected a dataset from online debate forum and the argument behavior was analyzed on four subtasks. Ji et al. [10] proposed discrete variational autoencoders to identify interactive arguments pairs in two posts with opposite stances. Cheng et al. [5] proposed a more challenging task, argument pair extraction, as the data to be processed are two unstructured documents and the goal of the task is both to get the arguments in them and to make them form the correct argument pairs.

From an alternative point of view, the task of APE can be considered as a form of multi-task learning. Miwa and Sasaki [15] introduced a table representation for entities and relations detection. In their tables, diagonal squares are used to predict entity types and non-diagonal lines are used to predict relationships. To solve the joint entity and relation extraction task, Wang and Lu [25] use cross-updated table encoders and sequence encoders to fill the table of entities and relations. Chen et al. [3] proposed a synchronous double-channel recurrent network for aspect-opinion pair extraction, in which a recurrently updated opinion entity extraction unit and relation detection unit are used to simultaneously extract aspects, opinions and the relations between them. However, the APE task is more complex because its extraction targets are sentences rather than words, and sentence vectors are more challenging to learn than word vectors.

3 Model

In order to address the challenges posed by the Argument Pair Extraction (APE), we propose a novel **M**ulti-fusion **C**ross-update **R**ecurrent **N**etwork (MCRN) architecture, as illustrated in Fig. 1. Following previous work [5], we conduct argument pair extraction on peer review and rebuttal datasets. The goal of APE is to extract all argument pairs in the form of $(\mathbf{arg}^{rv}, \mathbf{arg}^{rv})$, where $\mathbf{arg}$ refers to a contiguous sequence of sentences representing an argument.

3.1 Sentence Embedder

The initial stage of our proposed model entails the acquisition of sentence embeddings via a sentence embedder. It is inspired by the successful application of BERT which is used to obtain contextual semantic embeddings. Specifically, given a sentence, we obtain the token embeddings through BERT. These token embeddings are subsequently passed through a BiLSTM to obtain original sentence embeddings. As a result, for a given passage X, we can obtain the corresponding passage embedding H, which comprises the original sentence embeddings.

Note that the same model structure is employed for the passages with interaction, but with different parameters, as well as subsequent sections. For example, given two passages, a review and a rebuttal, we can obtain their respective sentence embeddings, denoted as H_{rv} and H_{rb}, by applying the sentence embedder.

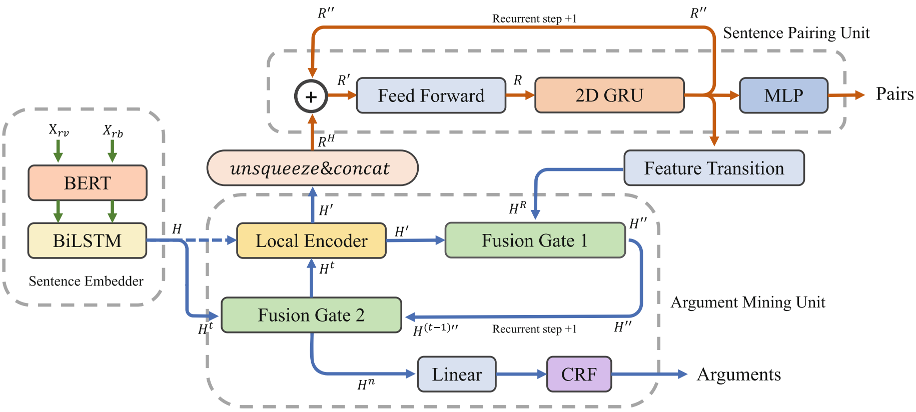


Fig. 1. The Multi-fusion Cross-update Recurrent Network - Model Architecture

3.2 Argument Mining Unit

The Argument Mining Unit (AMU) is responsible for updating sentence representations and predicting arguments. AMU, a recurrent structure, executes this task by sending updated sentence feature to the predictor while simultaneously re-fusing the feature with the original sentence feature. This integrated representation is then used as input for the next recurrent step of the argument mining process, making it an iterative and progressive process.

Local Encoder. In the context of the APE task, arguments typically consist of consecutive sentences. As a result, determining the membership of a given sentence within an argument requires only local information pertaining to itself and its immediate neighbors. For this purpose, a multihead self-attention encoder [23] with a masking mechanism is employed as a local encoder to extract the relevant features of adjacent sentences. The original sentence features, denoted as H, are initially fed as input to the local encoder during the first recurrent step. This allows us to obtain a sentence embedding H' that aggregates information from adjacent sentences.

The updated sentence features serve two primary functions: generating relation features for the sentence pairing unit, and fusing relation features for argument mining. The relation features R^H are obtained by concatenating every pair of updated sentence features H' from different passages. For instance, given review features H'_{rv} and rebuttal features H'_{rb} containing m and n sentences, respectively. The relation features are constructed by the following.

$$R^H = \{[\mathbf{h}'_{rv,i}; \mathbf{h}'_{rb,j}] | i \in \{1, 2, \ldots, m\}, j \in \{1, 2, \ldots, n\}\} \tag{1}$$

Fusion Gate. Our proposed model incorporates two fusion gates with identical structures. Specifically, a fusion gate is introduced after the local encoder to incorporate information from the other passage. This is achieved by combining the sentence features H' with H^R. Here, the sentence features H^R is described in Sect. 3.3. The fusion gate operates as follows:

$$gate = \sigma([H'; H^R]W_{gate} + b_{gate}) \tag{2}$$

$$H'' = H' \odot gate + H^R \odot (1 - gate) \tag{3}$$

where $W_{gate} \in \mathbb{R}^{2d \times d}$ and b_{gate} are learnable, and σ denotes sigmoid function.

The fused features H'' and original sentence features H will go through another fusion gate with the same structure as above to obtain the input features H^{t+1} for the argument mining unit of next recurrent step. Additionally, the sentence embeddings H^n of last recurrent will then be used to predict the arguments.

Arguments Predictor. To address the challenge of argument mining, we frame it as a sequence labeling task. Specifically, we use a conditional random field (CRF) approach [11] to perform this task. The CRF model involves a state transition matrix $T \in \mathbb{R}^{K \times K}$ and a state score matrix $S \in \mathbb{R}^{N \times K}$, where N is the number of sentences and K is the label dimension. The sentence features H'' will be used to calculate the state score S by a linear layer. Given a passage $X = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$, we predict a sequence of labels $\hat{Y} = \{\hat{y}_1, \hat{y}_1, \ldots, \hat{y}_N\}$, and its score can be defined as:

$$score(X, \hat{Y}) = \sum_{i=0}^{N} T_{\hat{y}_i, \hat{y}_{i+1}} + \sum_{i=1}^{N} S_{x_i, \hat{y}_i} \tag{4}$$

Then the probability of sequence Y will be calculated by the follow:

$$p(Y|X) = \frac{exp(score(X, Y))}{\sum_{\tilde{Y}} exp(score(X, \tilde{Y}))} \tag{5}$$

where $\tilde{Y} \in Y_X$, and Y_X denotes all possible label sequences.

3.3 Sentence Pairing Unit

Sentence Pairing Unit (SPU) is a component designed to detect the relationship between two sentences within interacted passages. The SPU is a recurrent structure that utilizes its own output features R'' and attention-based AMU features R^H to form input relation features R'. These two types of information are integrated using an element-wise summation to ensure that both forms of information are considered in the detection of sentence relationships.

In the first recurrent, $R'' \in \mathbb{R}^{|rv| \times |rb| \times 2d}$ is initialized by xavier normal [9]. Then we update the relation features R' to R using a feed-forward network:

$$R = R' + (max(0, R'W + b))W' + b' \tag{6}$$

Next, we need to incorporate contextual semantic information between different passages into the relation features. The relation feature matrix $R \in \mathbb{R}^{|rv| \times |rb| \times 2d}$ comprises $|rv|$ review sentences and $|rb|$ rebuttal sentences. The relation between the i-th review sentence and the j-th rebuttal sentence at the t-th recurrent step is denoted as $R_{i,j,t}$. Note that t here belongs to the GRU and not the MCRN. We update $R_{i,j,t}$ using a 2D-GRU [25] as follows:

$$R''_{i,j,t} = 2DGRU(R_{i,j,t-1}, R_{i-1,j,t}, R_{i,j-1,t}) \tag{7}$$

The relation feature R'' of the final recurrent step is fed into a multilayer perceptron (MLP) with three linear layers and two ReLU layers. The MLP computes the probability $p(\hat{y}^{pair}|\mathbf{s}_{rv}, \mathbf{s}_{rb})$ indicating whether sentence $\mathbf{s}_{rv}$ and sentence $\mathbf{s}_{rb}$ form a sentence pair.

Feature Transition. To enable the use of relational features in AMU, it is imperative to transform them into sentence embeddings denoted as H^R. This transformation requires reducing the dimensions of the features. To minimize computational costs, we employ an average method, where we obtain the review sentence embeddings by averaging the feature R'' row-wise, and the rebuttal sentence embeddings by averaging it column-wise. These sentence features are then updated using a linear layer, followed by layer normalization to ensure that the dimensions of H^R are consistent with the features in AMU.

3.4 Joint Learning

To establish a mutually reinforcing relationship between the argument mining unit and sentence pairing unit, we aim to maximize the probability of golden sequence $p(Y|X)$ and the probability of golden pair $p(y^{pair}|X_{rv}, X_{rb})$ simultaneously. For argument mining unit, we minimize the negative logarithm of the conditional probability of the golden label as the loss function:

$$\mathcal{L}_A = -\log p(Y|X) = \log \sum_{\tilde{Y}} \exp(score(X, \tilde{Y})) - score(X, Y) \tag{8}$$

For sentence pairing unit, the binary cross-entropy loss function is used for sentence pairing:

$$\begin{aligned} \mathcal{L}_R = -\sum_{i,j} (&y^{pair} \log p(y^{pair} = 1|\mathbf{s}_{rv,i}, \mathbf{s}_{rb,j}) \\ &+ (1 - y^{pair}) \log p(y^{pair} = 0|\mathbf{s}_{rv,i}, \mathbf{s}_{rb,j})) \end{aligned} \tag{9}$$

where $\mathbf{s}_{rv,i}$ and $\mathbf{s}_{rb,j}$ are sentences from the passages X_{rv} and X_{rb}, respectively.

Then, the overall loss function of the model is expressed as:

$$\mathcal{L} = \mathcal{L}_A + \lambda \cdot \mathcal{L}_R \tag{10}$$

where λ is the weight of sentence pairing loss.

3.5 Inference

To extract arguments and argument pairs, an additional inference module is introduced. The sequence labeling result with the highest conditional probability, obtained through the CRF algorithm, is taken as the labeling prediction result for argument extraction.

$$\hat{Y} = \arg\max_{\tilde{Y}} p(\tilde{Y}|X) \tag{11}$$

After obtaining the sets of review arguments $A^{rv} = \{\mathbf{arg}_1^{rv}, \mathbf{arg}_2^{rv}, \dots\}$ and rebuttal arguments $A^{rb} = \{\mathbf{arg}_1^{rb}, \mathbf{arg}_2^{rb}, \dots\}$, we compute the scores $\hat{\delta}$ for each argument pair. Let $\mathbf{arg}_i^{rv}$ and $\mathbf{arg}_j^{rb}$ denote a review argument and a rebuttal argument, respectively. The score $\hat{\delta}$ is calculated as follows:

$$\hat{\delta}_{i,j} = \frac{\sum_{\mathbf{s}^{rv} \in \mathbf{arg}_i^{rv}} \sum_{\mathbf{s}^{rb} \in \mathbf{arg}_j^{rb}} \mathbb{1}_{p(\tilde{y}^{pair}=1|\mathbf{s}^{rv},\mathbf{s}^{rb})>0.5}}{|\mathbf{arg}_i^{rv}| * |\mathbf{arg}_i^{rb}|} \tag{12}$$

where $|\cdot|$ is the function to calculate the number of sentences. Given a threshold score δ, we consider two arguments $\mathbf{arg}_i^{rv}$ and $\mathbf{arg}_j^{rb}$ as an argument pair if $\hat{\delta}_{i,j} > \delta$. According to this approach, we obtain a set of argument pairs $C = \{(\mathbf{arg}_1^{rv}, \mathbf{arg}_1^{rb}), (\mathbf{arg}_2^{rv}, \mathbf{arg}_2^{rb}), \dots\}$.

4 Experiments

4.1 Dataset

To evaluate the effectiveness of MCRN, we conduct experiments on the benchmark Review-Rebuttal dataset (RR dataset) proposed by [5]. The dataset consists of 4764 pairs of peer reviews and author rebuttals collected from ICLR 2013 to ICLR 2020. The dataset has two versions: RR-passage-v1 and RR-Submission-v2. They are both divided into training, development and test sets in the ratio of 8:1:1.

4.2 Parameter Configuration

We adopt the pre-trained **BERT-base** with dimension 768 as the token embedder and we freeze it during the training process. During training, we use the Adam optimizer with a learning rate of 2e−4 and train the model for 20 epochs. The hyperparameters of our model are set as follows: batch size = 2, dropout rate = 0.5, model dimension $d = 256$, local encoder head number $N_{head} = 2$, relation loss weight $\lambda = 0.4$, argument pair threshold $\delta = 0.5$. We tuned these hyperparameters primarily based on the RR-submission-v2 dataset. To measure the performance of MCRN, we report *Precision*, *Recall* and F_1*-score* to evaluate the results on three tasks, including argument mining, sentence pairing and APE. We take the results on the test set when the model achieves optimal results on the development set.

4.3 Baselines

To demonstrate the performance of MCRN, we compare it with the following baselines:

- **PL-H-LSTM-CRF** [5] is a pipeline model that trains argument mining and sentence pair detection separately and then integrates their results for extracting argument pairs.
- **MT-H-LSTM-CRF** [5] trains two subtasks simultaneously in a multi-tasks framework, where a CRF is used to solve the argument mining task and a linear layer is used to solve the sentence matching task.
- **MLMC** [6] solves the sentence pairing problem using an attention-guided multi-cross encoding-based model. The main MLMC architecture consists of multi-layer multi-cross encoder layer.

In addition to the above three baseline models, we construct an additional baseline model, **MCRN-ML**. Which differs from MCRN in that it does not use a recurrent structure but a multi-layer stacking structure.

4.4 Experiment Results

Main Results. The comparison of our model and the baselines on argument pair extraction is shown in Table 1 and Table 2. It can be seen that the experimental results of MCRN are better than all baselines on both datasets. According to Table 1, MCRN outperforms **PL-H-LSTM-CRF** by 14.81% F_1 score on argument pair extraction, but both the pipeline model and the joint learning model exhibit comparable performance on the argument mining subtask. This indicates that there is a significant error accumulation in the pipeline model, while the joint learning model can avoid the problem as much as possible.

We also compare the MCRN with the joint learning model. According to Table 2, the results of **MCRN** are 8.81% and 2.61% inferior to **MT-H-LSTM-CRF** and **MLMC**. **MT-H-LSTM-CRF** only uses a linear layer for the sentence pairing problem, which prevents it from fully learning the relation features of sentence pairs. The results of MLMC and our approach illustrate the importance of explicitly modeling sentence pair relations. In particular, the F_1 score of **MCRN-ML** is only comparable to **MT-H-LSTM-CRF** in argument pair extraction and significantly lower than other methods in sentence pairing. This indicates that MCRN can efficiently utilize the feature representation of the pre-trained model through recurrent structure to improve the ability of learning sentence relations.

To exhibit the efficacy of our proposed approach, we also conducted experiments on the claim-evidence pair extraction (CEPE) dataset [4]. Notably, since the CEPE dataset is substantially smaller in size compared to the RR dataset, we did not implement the mask in the local encoder. Our experimental results, as demonstrated in Table 3, exhibit a performance enhancement of 1.68 when compared to the previous state-of-the-art method. Furthermore, we conducted an investigation into the influence of recurrent steps on model performance. Our

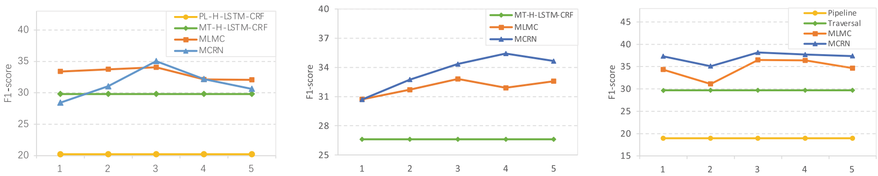
Table 1. Experiment results on RR-Passage-v1.

Models	Argument Mining			Sentence Pairing			APE		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
PL-H-LSTM-CRF	**73.10**	67.65	70.27	51.34	42.08	46.25	21.24	19.30	20.23
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**	69.37	**62.69**	42.33	50.53	**40.27**	29.53	34.07
MCRN-ML	69.92	71.71	70.80	58.63	38.29	46.33	37.82	24.84	29.99
MCRN	69.27	71.39	70.32	58.50	**49.47**	**53.61**	38.20	**32.37**	**35.04**

Table 2. Experiment results on RR-Submission-v2.

Models	Argument Mining			Sentence Pairing			APE		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
MT-H-LSTM-CRF	**70.74**	69.46	70.09	52.05	46.74	49.25	27.24	26.00	26.61
MLMC	69.53	**73.27**	**71.35**	60.01	46.82	52.60	37.15	29.38	32.81
MCRN-ML	69.67	69.10	69.38	59.59	38.81	47.00	36.15	24.72	29.36
MCRN	70.52	69.53	70.02	**60.13**	**48.23**	**53.53**	**39.62**	**32.03**	**35.42**

results, illustrated in Fig. 2, indicate that although there is some variability in model performance across the three datasets, overall, increasing the depth of the model leads to improved performance. Specifically, our findings demonstrate that the optimal level of performance is achieved at 3 to 4 recurrent steps, beyond which there is a decline in performance, likely due to overfitting of the model.



(a) RR-passage-v1 (b) RR-submission-v2 (c) CEPE

Fig. 2. Performance Comparison of Different Layer/Recurrent on Three Datasets

4.5 Ablation Study

To evaluate the effectiveness of the various modules in our proposed MCRN model, we conducted ablation experiments on the RR-submission-v2 dataset, and present the results in Table 4. Specifically, we examined the impact of removing the Local Encoder mask and replacing the Fusion Gates with mean function, as well as sharing the local encoder and CRF layers between the review and rebuttal data. The results demonstrate that the removal of Fusion Gate 1 has a marginal effect on

Table 3. Experiments on CEPE

Models	Pre.	Rec.	F_1
Pipeline	16.58	22.11	18.95
Traversal	24.06	**38.74**	29.69
MLMC*	48.92	29.08	36.48
MCRN	**54.25**	29.43	**38.16**

Table 4. Ablation Experiments

Model Settings	APE F_1	$\Delta(F_1)$
MCRN	35.42	-
w/o Fusion Gate 1	33.65	-1.77
w/o Fusion Gate 2	32.10	-3.32
w/o Local Encoder	31.81	-3.61
sharing Local Encoder	33.51	-1.91
sharing CRF	33.52	-1.90
sharing both	33.91	-1.51

performance, while the absence of Fusion Gate 2 and Local Encoder mask significantly impair the performance of our APE model. These results imply that proper fusion of pre-trained features can enhance experimental performance. Additionally, our study reveals that global attention adds excessive noise and that distant sentences are unhelpful in feature learning of target sentences. Moreover, our findings indicate that sharing the Local Encoder and CRF layers between the review and rebuttal data results in a loss of F_1 score of -1.91 and -1.90, respectively. This could be attributed to the differing data distributions between the two sets. However, sharing both modules yields a lesser F_1 score drop than sharing just one, indicating that common feature learning through sharing improves the model's performance to some extent. Nevertheless, our results suggest that customizing different encoding layers for different data distributions is more effective, as long as the data distribution is significantly different.

5 Conclusion

In this paper, we focused on argument pair extraction and proposed Multi-fusion Cross-update Recurrent Network. The argument mining unit and sentence pairing unit are designed to extract arguments and sentence pairs simultaneously. The two units cross update their features in a recurrent network. The results of argument and sentence pairs allow us to obtain argument pairs that combine information from the above two units. Extensive experiments on benchmark datasets show that MCRN has a significant improvement in contrast to baseline methods.

Acknowledgement. The work reported in this paper was partially supported by a National Natural Science Foundation of China project 61963004.

References

1. Bao, J., Liang, B., Sun, J., Zhang, Y., Yang, M., Xu, R.: Argument pair extraction with mutual guidance and inter-sentence relation graph. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3923–3934 (2021)

2. Chalaguine, L.A., Hunter, A., Potts, H., Hamilton, F.: Impact of argument type and concerns in argumentation with a chatbot. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1557–1562 (2019). https://doi.org/10.1109/ICTAI.2019.00224
3. Chen, S., Liu, J., Wang, Y., Zhang, W., Chi, Z.: Synchronous double-channel recurrent network for aspect-opinion pair extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6515–6524. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.582. http://www.aclanthology.org/2020.acl-main.582
4. Cheng, L., Bing, L., He, R., Yu, Q., Zhang, Y., Si, L.: IAM: a comprehensive and large-scale dataset for integrated argument mining tasks. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2277–2287 (2022)
5. Cheng, L., Bing, L., Yu, Q., Lu, W., Si, L.: APE: argument pair extraction from peer review and rebuttal via multi-task learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7000–7011 (2020)
6. Cheng, L., Wu, T., Bing, L., Si, L.: Argument pair extraction via attention-guided multi-layer multi-cross encoding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 6341–6353 (2021)
7. Dumani, L., Neumann, P.J., Schenkel, R.: A framework for argument retrieval. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 431–445. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_29
8. Ein-Dor, L., et al.: Corpus wide argument mining-a working solution. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 7683–7691 (2020)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
10. Ji, L., Wei, Z., Li, J., Zhang, Q., Huang, X.J.: Discrete argument representation learning for interactive argument pair identification. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5467–5478 (2021)
11. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of 18th International Conference on Machine Learning, pp. 282–289 (2001)
12. Lawrence, J., Reed, C.: Argument mining: a survey. Comput. Linguist. **45**(4), 765–818 (2020)
13. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., Slonim, N.: Context dependent claim detection. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 1489–1500 (2014)
14. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., Slonim, N.: Context dependent claim detection. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, pp. 1489–1500. Dublin City University and Association for Computational Linguistics (2014). http://www.aclanthology.org/C14-1141

15. Miwa, M., Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1858–1869 (2014)
16. Persing, I., Ng, V.: Why can't you convince me? Modeling weaknesses in unpersuasive arguments. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, pp. 4082–4088 (2017). https://doi.org/10.24963/ijcai.2017/570
17. P Petasis, G., Karkaletsis, V.: Identifying argument components through TextRank. In: Proceedings of the Third Workshop on Argument Mining (ArgMining2016), Berlin, Germany, pp. 94–102. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/W16-2811. http://www.aclanthology.org/W16-2811
18. Poudyal, P.: A machine learning approach to argument mining in legal documents. In: Pagallo, U., Palmirani, M., Casanovas, P., Sartor, G., Villata, S. (eds.) AICOL 2015-2017. LNCS (LNAI), vol. 10791, pp. 443–450. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00178-0_30
19. Poudyal, P., Šavelka, J., Ieven, A., Moens, M.F., Goncalves, T., Quaresma, P.: ECHR: legal corpus for argument mining. In: Proceedings of the 7th Workshop on Argument Mining, pp. 67–75 (2020)
20. Song, W., Song, Z., Fu, R., Liu, L., Cheng, M., Liu, T.: Discourse self-attention for discourse element identification in argumentative student essays. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2820–2830 (2020)
21. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 46–56 (2014)
22. Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., Lee, L.: Winning arguments: interaction dynamics and persuasion strategies in good-faith online discussions. In: Proceedings of the 25th International Conference on World Wide Web, pp. 613–624 (2016)
23. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
24. Wachsmuth, H., et al.: Computational argumentation quality assessment in natural language. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain, pp. 176–187. Association for Computational Linguistics (2017). http://www.aclanthology.org/E17-1017
25. Wang, J., Lu, W.: Two are better than one: joint entity and relation extraction with table-sequence encoders. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1706–1721 (2020)
26. Wei, Z., et al.: A preliminary study of disputation behavior in online debating forum. In: Proceedings of the Third Workshop on Argument Mining (ArgMining2016), Berlin, Germany, pp. 166–171. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/W16-2820. http://www.aclanthology.org/W16-2820