Weakly-supervised Argument Mining with Boundary Refinement and Relation Denoising

Anonymous ACL submission

Abstract

Argument mining (AM) involves extracting argument components and predicting relations between them to create argumentative graphs, which are essential for applications requiring argumentative comprehension. To automatically provide high-quality graphs, previous works require a large amount of human-annotated training samples to train AM models. Instead, we leverage a large language model (LLM) to assign pseudo-labels to training samples for reducing reliance on human-annotated training data. However, the training data weakly-labeled by the LLM are too noisy to develop an AM model with reliable performance. In this paper, to improve the model performance, we propose a center-based component detector that refines the boundaries of the detected components and a relation denoiser to deal with noise present in the pseudo-labels when classifying relations between detected components. Experimentally, our AM model improves the boundary detection obtained from the LLM by up to 16% in terms of IoU_{75} and of the relation classification obtained from the LLM by up to 12% in terms of macro-F1 score. Our AM model achieves new state-of-the-art performance in weakly-supervised AM, showing up to a 6% improvement over the state-of-the-art component detector and up to a 7% improvement over the state-of-the-art relation classifier. Additionally, our model uses less than 20% of human-annotated data to match the performance of state-of-the-art fully-supervised AM models.

1 Introduction

Argumentative graphs extracted from argumentative text can enhance users' understanding of the text (Palau and Moens, 2009; Lawrence and Reed, 2019). Consequently, argument mining (AM) techniques have widespread applications in various domains, including patient-generated content analysis (Mayer et al., 2020; Stylianou and Vlahavas, 2021; Yeginbergenova and Agerri, 2023), legal reasoning (Wyner et al., 2010; Poudyal et al., 2020), and opinion mining (Niculae et al., 2017).

Building an argumentative graph requires two models: (1) a component detector to identify and label the components of an argument, and (2) relation classifier that identifies argument relations between the found argument components and determines their head or tail function. Previous work has considered AM for different domains such as clinical trials (Mayer et al., 2020) and electronic rulemaking (Niculae et al., 2017). Moreover, it considered data on varying granularity such as documents (Stab and Gurevych, 2017; Poudyal et al., 2020) and paragraphs (Niculae et al., 2017; Mayer et al., 2020). Some works adopted plain text as input (Mayer et al., 2020; Stylianou and Vlahavas, 2021), while others (Niculae et al., 2017; Bao et al., 2021; Galassi et al., 2023) use argument components as input. In this paper, we follow the approach of predicting argumentative graphs from the plain text of a paragraph.

The state-of-the-art AM model for this line of work combines a BIO sequence tagger[1] (which detects argument components) and a text classifier (which classifies relations between components) (Mayer et al., 2020). However, this approach has two drawbacks. First, the BIO sequence tagger frequently mislabels B-tokens as I-tokens, leading to detection errors for the boundaries of argument components. We address this problem by designing a center-based argument detector that assigns probabilistic labels (as opposed to hard labels). Second, training a argument relation classifier often requires access to significant quantities of human annotated data. Unfortunately, using weak labels provided by an LLM are too noisy to solely rely on when training the relation classifier. Therefore, we propose a relation denoiser that further improves

[1] The BIO tagger assigns Beginning, Inside or Outside labels to the tokens (i.e., sub-words) of a sequence.

the relation classification obtained from the LLM. Specially, the relation denoiser dynamically adjusts the contributions between two weakly labeled training sets, one obtained by an LLM annotation and one by a model fine-tuned on the golden-annotated benchmark development data (Zhu et al., 2023). As a result, the combination of the boundary refinement of argument components and the relation denoising yields a weakly supervised approach that matches the performance of fully supervised AM. We evaluate the proposed methods on four standard, publicly available AM datasets (AbstRCT-neoplasm, AbstRCT-glaucoma, AbstRCT-mixied, and CDCP) (Niculae et al., 2017; Mayer et al., 2020; Bao et al., 2021; Galassi et al., 2023). Our contributions are the following.

- A novel weakly supervised AM model that matches state-of-the-art fully-supervised AM using under 20% human-annotated data.
- The novel center-based component detector refines argument components' boundaries by using soft probabilistic BIO labels rather than hard labels.
- The relation denoiser improves the performance of argument relation classification by blending two types of weakly labeled training data.

2 Related Work

Stab and Gurevych (2017) propose a feature-based Integer Linear Programming model to jointly predict extracted argument components' labels and the relations between them in persuasive essays and introduces a constraint unique to the persuasive essays dataset: the number of parents of each claim does not exceed one. Stab and Gurevych (2017) and Eger et al. (2017) design an end-to-end AM model to extract argumentative graphs in the persuasive essays dataset. However, Mayer et al. (2020) and Stylianou and Vlahavas (2021) point out that the TreeLSTM-based models used do not perform well on long texts, necessitating the imposing of distance constraints. The above models jointly learn argument component and argument relation identification and impose additional constraints on the shape of the argumentative graph, which we restrain from in our work. ResAttARG (Galassi et al., 2023) employs a multi-objective residual network to identify the labels of argument components and the argument relations between them assuming that both tasks rely on similar features, an assumption which might not always be correct.

As a pipeline model, Mayer et al. (2020) leverage transformer-based language models with a RNN to identify argument components from text, and a classifier predicts relations between components. This model is a baseline in our experiments. TransforMed (Stylianou and Vlahavas, 2021) is also a combination of a sequence tagger and a text classifier, but it implements a domain-specific mechanism for extracting external medical knowledge, so we exclude it for fair comparison.

Although fully supervised AM models have been proposed, expensive manual annotation remains a challenge (Miller et al., 2019; Iskender et al., 2021). The semi-supervised AM model of Habernal and Gurevych (2015) assigns pseudo-labels to unlabeled data by determining the similarity between labeled data points and unlabeled samples, but does not focus on refining argument component boundaries neither on denoising the weak labels, as we propose.

3 Method

Fig. 1 shows the overall architecture of the proposed framework. Firstly, our novel center-based component detector refines the boundaries of the argument components (see 3.1). Secondly, the relation denoiser blends two weakly labeled training sets to improve accuracy of classifying the relations between the detected arguments(see 3.2).

3.1 Center-based Component Detector

Given N sentences of the text, the LLM generates weakly labeled argument components where the k_{th} sentence with m words is denoted as $X_k = \{x_{k,1}, \cdots, x_{k,m}\}$. Fig. 2 shows the working principle of the center-based component detector. We utilize a Gaussian Kernel to generate a mask over the sentence. The peaks of the mask are the center points of argument components. Similarly, we generate a mask for argument component's boundaries. We then classify the found argument components into pre-defined argumentative labels.

More specifically, let $\tilde{x}_{k,left}$ be the argument component's left boundary index and $\tilde{x}_{k,right}$ be its right boundary index in the input text (obtained by the LLM). The coordinate of the center point of this argument component is $\tilde{x}_{k,center} = \frac{\tilde{x}_{k,left}+\tilde{x}_{k,right}}{2}$ and we round-down the $\tilde{x}_{k,center}$

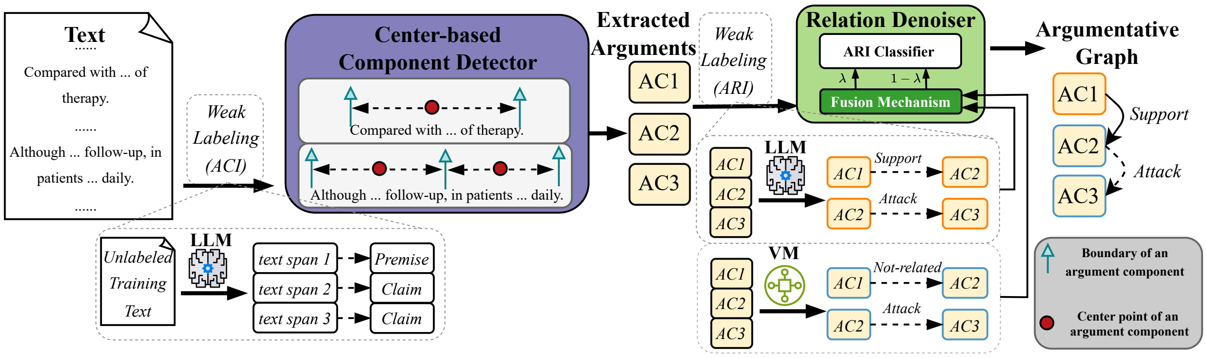


Figure 1: The overall architecture of our proposed framework. First, a LLM identifies argument components in text (where "AC" refers to the argument components). The center-based component detector then refines these boundaries to provide better component detection. Next, a LLM weakly labels pairs of argument components to provide weakly labeled argument relation identification data. The relation denoiser enhances the performance of the relation classifier by combining two weakly labeled training sets: LLM annotation and model annotation from the relation classifier trained on the gold-standard benchmark development set (the latter model is called "VM").

into an integer. We use a Gaussian kernel $Y = \exp\left(-\frac{\tilde{x}_{k,j}-\tilde{x}_{k,center}}{2\sigma^2}\right)$, where $\{1 \leq \tilde{x}_{k,j} \leq m\}$, σ is $\frac{\tilde{x}_{k,right}-\tilde{x}_{k,left}}{\zeta}$ and ζ is the shape coefficient that controls the shape of the mask. Similarly, the masks for the boundaries of an argument component are generated. Thus, we get the mask for argument components' center points $G_k = \{G_{k,1}, \cdots, G_{k,m}\}$ and the boundary mask $S_k = \{S_{k,1}, \cdots, S_{k,m}\}$. If two masks overlap, we select the maximum value at each location.

Following (Mayer et al., 2020), we use SciBERT (Beltagy et al., 2019) as text encoder. After sub-word tokenization, the input sentence composed of m words is represented with d tokens, $\mathbf{x}'_k = \{x'_{k,1}, \cdots, x'_{k,d}\}$. The mask vector of the argument components' center points is $\mathbf{g}_k = \{g'_{k,1}, \cdots, g'_{k,d}\}$, the mask vector of the argument components' boundaries is $\mathbf{s}_k = \{s'_{k,1}, \cdots, s'_{k,d}\}$, and the argumentative label vector is $\mathbf{c}_k = \{c'_{k,1}, \cdots, c'_{k,d}\}$. We encode the input vector, and linear layers predict the mask of the argument components' center points $\hat{\mathbf{g}'}_k = \{\hat{g}'_{k,1}, \cdots, \hat{g}'_{k,d}\}$, the mask of the argument components' boundaries $\hat{\mathbf{s}'}_k = \{\hat{s}'_{k,1}, \cdots, \hat{s}'_{k,d}\}$, and the argumentative labels $\hat{\mathbf{c}'}_k = \{\hat{c}'_{k,1}, \cdots, \hat{c}'_{k,d}\}$. Because tokenization of the encoder could distort the shape of masks, it becomes challenging to extract peaks from the predictions. Therefore, following (Wang et al., 2020), we first generate an ignore mask and then design a masked MSE loss function to learn the model to predict the label of a word's first token.

The ignore mask $\mathbf{ig}'$ is created by setting the first

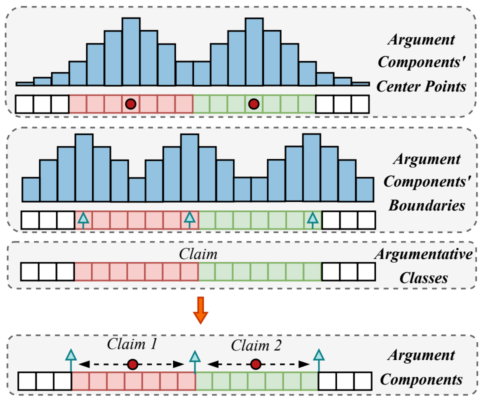


Figure 2: The figure shows the working principle of the center-based component detector. We locate argument components based on the peaks of the mask of center points. Similarly, we determine the boundaries of the argument components. Next, we segment the argument components from the text based on the predicted masks of center points and boundaries. After assigning argumentative labels to the detected components, we obtain the segmented argument components with their corresponding labels.

token in a word to 1 and all its remaining tokens to 0. The center loss $\mathcal{L}_{ce}$, boundary loss $\mathcal{L}_{bd}$, and class loss $\mathcal{L}_{cl}$ are:

$$\mathcal{L}_{ce} = \frac{1}{Nd}\sum_{k=1}^{N}\sum_{u=1}^{d}\left[(\hat{g'}_{k,u} - g_{k,u})^2 ig_{k,u}{}'\right], \quad (1)$$

$$\mathcal{L}_{bd} = \frac{1}{Nd}\sum_{k=1}^{N}\sum_{u=1}^{d}\left[(\hat{s'}_{k,u} - s_{k,u})^2 ig_{k,u}{}'\right], \quad (2)$$

$$\mathcal{L}_{cl} = -\frac{1}{Nd}\sum_{k=1}^{N}\sum_{u=1}^{d}\Big[c_{k,u}\log(\hat{c}_{k,u}) ig_{k,u}{}'\Big]. \quad (3)$$

We train three sub-models separately on the data weakly labeled by the LLM with Continuous Fine-Tuning (CFT) (Zhu et al., 2023). CFT first fine-tunes the model with the weakly annotated training data and then further fine-tunes the model with the golden-annotated benchmark development set.

During inference, we identify the argument components' center points and boundaries based on the peaks of their predicted masks. Finally, we predict the argumentative labels of found argument components.[2]

3.2 Relation Denoiser

We build the set of M argument component pairs. The LLM generates the weak relation labels for each pair (r pre-defined argument relation labels). The weak labels produced by the LLM are too noisy to rely on solely for training the relation classifier. Therefore, we create an additional weakly-labeled dataset by training the relation classifier using the small golden-annotated benchmark development set and using it to weakly annotate the training data. We apply the fusion mechanism to dynamically blend the weight assigned to each weakly labeled dataset.

The weak labels of the LLM annotation and of the model annotation - the latter trained on golden-annotated benchmark development data - are denoted as label vectors $\mathbf{y}^{llm} \in \mathbb{R}^M$ and $\mathbf{y}^{vc} \in \mathbb{R}^M$, respectively. We utilize Sci-BERT as our encoder and employ a linear layer as the classifier. The logits of the relation classifier are represented by the vector $\hat{\mathbf{y}}$. The fusion mechanism dynamically controls the contributions of the two weakly labeled training data and its workflow is shown in Algorithm 1. Line 4 in the algorithm states the prediction $\hat{\mathbf{y}}^p$. To calculate the overlapping labels of two vectors, we define a element-wise comparison function $\mathcal{H}(.)$, i.e., if two scalars are the same, the function outputs 1; otherwise 0. Line 5 represents the overlapping labels between $\mathbf{y}^{vc}$ and $\mathbf{y}^{llm}$, and line 6 the overlapping labels between $\mathbf{y}^{vc}$ and $\hat{\mathbf{y}}^p$. Line 7 states the logical conjunction between two one-hot vectors. We obtain the score τ in line 8.

[2] If the detector predicts the "None" label for a given component, that component is considered as non-argumentative.

Algorithm 1 Algorithm for Fusion Mechanism

1: **Input:** Logits $\hat{\mathbf{y}} \in \mathbb{R}^{M\times d}$, Label vectors $\mathbf{y}^{llm}$, $\mathbf{y}^{vc}$; Fusion confidence T; Maximum Epochs E; Model Parameters $\mathbf{\Theta}$; Learning rate η
2:
3: **while** $ep \leq E$ **do**
4: $\quad \hat{\mathbf{y}}^p \in \mathbb{R}^M \leftarrow \arg\max(\sigma(\hat{\mathbf{y}}))$
5: $\quad \mathbf{h}^{om} \in \mathbb{R}^M \leftarrow \mathcal{H}(\mathbf{y}^{vc}, \mathbf{y}^{llm})$
6: $\quad \mathbf{h}^{omp} \in \mathbb{R}^M \leftarrow \mathcal{H}(\mathbf{y}^{vc}, \hat{\mathbf{y}}^p)$
7: $\quad \mathbf{h}^{rm} \in \mathbb{R}^M \leftarrow \mathbf{h}^{om} \circ \mathbf{h}^{omp}$
8: $\quad \tau \leftarrow \frac{1}{M}\sum_{i=1}^{M}(h_i^{rm})$
9:
10: $\quad$ **if** $\tau < T$ **then**
11: $\qquad \mathcal{L} = -\frac{1}{Md}\sum_{i=1}^{M} h_i^{om}\sum_{j=1}^{d}[y_{i,j}^{vc}\log(\hat{y}_{i,j})]$
12: $\quad$ **else**
13: $\qquad \lambda \leftarrow \frac{1}{M}\sum_{i=1}^{M}(h_i^{omp})$
14: $\qquad \mathcal{L} = -\frac{1}{Md}\sum_{i=1}^{M}\sum_{j=1}^{d}\{\lambda[y_{i,j}^{vc}\log(\hat{y}_{i,j})]$ $+(1-\lambda)[y_{i,j}^{llm}\log(\hat{y}_{i,j})]\}$
15: $\quad$ **end if**
16: $ep = ep + 1$
17: $\mathbf{\Theta} = \mathbf{\Theta} - \eta\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta})$
18: **end while**
19: **Output:** $\mathbf{\Theta}$

During training, in the early stages (line 10, 11), we treat the overlapping labels of the two weakly annotated data as the correct labels to train a relation classifier. The relation classifier is initially trained on these labels using a masking tensor $\mathbf{h}^{rm}$ to ignore irrelevant labels. Once the relation classifier achieves a high score τ on the assumed correct labels, we allow the relation classifier to adjust the fusion parameter (λ) for the two weakly labeled training data. λ and $1-\lambda$ are the contributions of two weakly labeled datasets, and the λ is dynamically updated in the algorithm. At inference time, we use the trained relation classifier to provide predictions.

4 Experiments

In this section, we evaluate our AM model using four AM datasets, perform an ablation study, and conduct an in-depth analysis of the proposed methods.

4.1 Evaluation Datasets

AbstRCT is divided into three datasets based on disease category: neoplasm, glaucoma, and mixed (Mayer et al., 2020) The **neoplasm** dataset contains 350 documents for training, 50 for development, and 100 for testing. The neoplasm train set is utilized as the training set for the **glaucoma** and **mixed** datasets, each comprising 100 instances for testing. The argument component identification labels for the AbstRCT dataset are "Premise" and "Claim" and argument relation identification labels are "Support", "Attack" and "Not-related". The **CDCP** dataset includes 731 user comments about consumer debt collection practices from an eRulemaking website, with 581 examples for training and 150 for testing. We selected 100 samples from the training set for development. The argument component identification labels for the CDCP dataset are "Value", "Policy", "Testimony", "Fact" and "Reference" and the processed argument relation identification labels are "Related" and "Not-related" (following (Bao et al., 2021; Wei et al., 2024)).

4.2 Metrics and Parameter Setting

We evaluate the identified argument components with the IoU_{75} (Wei et al., 2023; Guan et al., 2023) metric and at the token-level by the macro-averaged F1 (F1) and micro-average F1 (indicated as f1 in the Tables). Following (Liu et al., 2020; Law and Deng, 2018), we set the IoU threshold as 0.75. The IoU measures the normalized overlap between the tokens of a ground truth component and the tokens of the prediction of that component with maximum overlap. Argument relations are evaluated with the macro-average F1 (F1) and micro-average F1 (f1) (3). F1 scores and their variance are computed with 5 different seeds. All models are trained on an NVIDIA GeForce RTX 3090 GPU. The AdamW optimizer (Loshchilov and Hutter, 2019) has a learning scheduler initialized at 2×10^{-5} and linearly decreased to 0. Hyperparameters T and ζ are selected by using grid search on the development set. The batch size is set to 8.

4.3 Baselines

All weakly supervised AM baselines utilize the weakly labeled AM datasets annotated by the ChatGPT (using the same prompt defined in Section A.1) and then are further fine-tuned on the golden-annotated benchmark development set. Fully supervised baselines utilize the golden-annotated training set. All weakly-supervised component detection baselines and relation classification baselines leverage the **C**ontinuous **F**ine-**T**uning (**CFT**) technique, i.e., further fine-tune baselines on golden-standard benchmark development sets, for fair comparisons.

BioBERT$_{\text{mlp}}$ (Mayer et al., 2020) uses BioBERT (Lee et al., 2020) as text encoder and subsequently applies a linear layer to predict token-level labels for argument component identification.

SciBERT$_{\text{mlp}}$ (Mayer et al., 2020) leverages SciBERT as text encoder and then applies a linear layer for argument component identification.

BioBERT$_{\text{gru-crf}}$ (Mayer et al., 2020) encodes text using BioBERT, followed by a GRU network. A Conditional Random Field (CRF) layer decodes the outputs from the GRU network into argument components.

SciBERT$_{\text{gru-crf}}$ (Mayer et al., 2020) replaces the encoder of the BioBERT-GRU-CRF by SciBERT and then predicts argument components from textual inputs.

ChatGPT addresses both argument component identification and argument relation identification tasks through in-context learning.

SciBERT$_{\text{senf}}$ (Mayer et al., 2020) uses the SciBERT model to encode pairwise argument components, which constitute the outputs of the SciBERT-GRU-CRF model. Subsequently, a linear layer decodes the outputs into argument relations.

RoBERTa$_{\text{senf}}$ (Mayer et al., 2020) replaces the SciBERT-Senf model's encoder by a RoBERTa model to predict argument relations.

SNet$_{\text{jtl}}$, inspired by (Zeng et al., 2019), conducts the joint-learning over two weakly labeled data where the contributions of the two weakly labeled data are equal, i.e., λ is fixed and $\lambda = 0.5$.

4.4 Results

Tab. 1 and Tab. 2 display the results for the argument component identification and argument relation identification tasks, respectively. Each table shows the model performance in two supervision settings: fully-supervised and weakly-supervised, across four datasets. To facilitate readability, we abbreviate the names of the four datasets as "Neo" for AbstRCT-neoplasm, "Gla" for AbstRCT-glaucoma, "Mix" for AbstRCT-mixed, and "CDCP" for CDCP. Upon analyzing the tables, we observe that:

(1) In Tab 1, our center-based component detector outperforms all baseline models on four datasets in both fully-supervised and weakly-supervised modes. In the fully-supervised setting, when compared with the state-of-the-art model SciBERT-

Models	Neoplasm			Glaucoma			Mixed			CDCP		
	f1	F1	IoU_{75}	f1	F1	IoU_{75}	f1	F1	IoU_{75}	f1	F1	IoU_{75}
Fully Supervised ACI												
$BioBERT_{mlp}$	89.10	84.95	79.03	91.04	89.71	84.15	90.17	87.31	82.17	74.01	52.43	76.16
$SciBERT_{mlp}$	89.48	85.74	81.22	90.12	89.41	83.53	89.09	86.21	80.02	75.14	55.80	80.38
$BioBERT_{gru\text{-}crf}$	89.38	86.15	80.34	91.97	90.56	84.86	91.64	88.97	82.98	73.07	51.67	75.07
$SciBERT_{gru\text{-}crf}$	89.63	86.77	81.70	91.03	89.62	83.93	89.86	86.98	80.26	75.28	55.95	80.89
$Ours_{(BioBERT_{mlp})}$	**90.77** ±0.22	88.00 ±0.34	**85.07** ±0.51	**92.15** ±0.16	**90.83** ±0.27	88.83 ±0.55	**91.88** ±0.12	**89.61** ±0.20	**85.33** ±0.51	75.03 ±0.37	54.76 ±0.45	**84.09** ±0.69
$Ours_{(SciBERT_{mlp})}$	90.83 ±0.31	**88.43** ±0.37	84.80 ±0.53	91.95 ±0.22	90.66 ±0.24	**89.06** ±0.40	91.00 ±0.11	88.58 ±0.17	84.58 ±0.47	**76.58** ±0.42	**56.64** ±0.47	83.63 ±0.67
Weak ACI labels												
ChatGPT	69.56	69.95	64.49	76.72	76.63	71.10	68.12	69.01	68.46	54.93	44.94	72.64
Weakly Supervised ACI												
$BioBERT_{mlp}$	87.03	83.83	73.33	90.26	88.60	82.10	88.71	85.98	76.56	68.44	51.51	69.69
$SciBERT_{mlp}$	87.84	68.45	73.57	89.83	88.04	81.63	88.94	86.21	77.58	65.92	56.32	66.92
$BioBERT_{gru\text{-}crf}$	88.57	85.67	74.63	90.35	89.04	82.54	89.20	86.69	77.78	68.97	52.28	73.03
$SciBERT_{gru\text{-}crf}$	88.16	85.30	73.15	90.04	88.28	79.84	89.03	86.58	75.23	70.04	59.33	77.26
$Ours_{(BioBERT_{mlp})}$	**89.13** ±0.33	**86.01** ±0.47	**80.29** ±0.54	**91.58** ±0.21	89.56 ±0.35	84.75 ±0.47	**89.74** ±0.18	87.02 ±0.23	81.88 ±0.32	71.20 ±0.51	60.49 ±0.63	**80.26** ±0.87
$Ours_{(SciBERT_{mlp})}$	88.91 ±0.29	85.94 ±0.35	79.56 ±0.46	90.81 ±0.33	**89.75** ±0.37	**85.31** ±0.51	89.25 ±0.21	**87.17** ±0.27	**82.21** ±0.43	**71.60** ±0.44	**60.55** ±0.59	79.77 ±0.79

Table 1: Results in terms of micro-averaged F1 (f1), macro-average F1 (F1), and IoU_{75} for the supervised and weakly-supervised argument component identification (ACI) task obtained on four datasets.

GRU-CRF, our approach achieves improvements of 3.10, 5.13, 2.35, and 4.32 percentage points in IoU_{75} scores on Neo, Gla, Mix, and CDCP datasets, respectively. In the weakly-supervised setting, our detector promote the IoU_{75} scores by 6.41, 5.53, 6.98, and 2.51 percentage points on Neo, Gla, Mix, and CDCP datasets, respectively. The results indicate a good refinement of the argument components' boundaries.

Models	Neo	Gla	Mix	CDCP
Fully Supervised ARI				
$SciBERT_{senf}$	60.78	56.21	61.88	55.21
$RoBERTa_{senf}$	61.19	55.13	60.23	54.72
Weak ARI labels				
ChatGPT	44.29	47.16	46.76	51.95
Weakly Supervised ARI				
$SciBERT_{senf}$	48.85	52.23	49.52	52.62
$RoBERTa_{senf}$	49.23	51.73	50.23	52.17
$SNet_{jtl}$	49.20	53.59	54.06	53.52
$Our_{(SciBERT_{Senf})}$	**56.75** ±1.86	**57.55** ±1.15	**58.19** ±1.58	**54.95** ± 0.78

Table 2: Results in terms of macro-F1 for supervised and weakly-supervised argument relation identification (ARI) obtained on four datasets.

(2) In Tab. 2, our relation denoiser outperforms all baseline relation classifier on four datasets in the weakly-supervised setting.[3] Compared with the state-of-the-art model, $Denoiser_{jtl}$, our approach achieves improvements of 7.55, 3.96, 4.13, and 2.33 percentage points in macro-F1 scores on Neo, Gla, Mix, and CDCP datasets, respectively.

(3) Our weakly-supervised AM model achieves performance very close to those of the previous fully supervised AM model. In the argument component identification task (Tab. 1), the evaluation results of our detector (in the weakly-supervised setting) are only 1.41, −0.45, 0.77, and 0.63 percentage points less than the fully supervised state-of-the-art model in terms of IoU_{75} on the Neo, Gla, Mix, and CDCP datasets, respectively. For the argument relation identification task (Tab. 2), the fully-supervised state-of-the-art model outperforms our relation denoiser (in the weakly-supervised setting) by only 5.59, −0.94, 5.01, and 0.26 percentage points in terms of macro-F1. on the Neo, Gla, Mix, and CDCP datasets, respectively. Moreover, our weakly-supervised AM model uses only 12.5%, 12.5%, 12.5% and 17.1% of the human-annotated

[3]Errors in the component detectors propagate to the relation classifiers.

samples in the Neo, Gla, Mix, and CDCP datasets, respectively. These numbers reflect the sizes of the development sets of these benchmark datasets.

Models	**Neo**	**Gla**	**Mix**	**CDCP**
Fully Supervised ARI				
SciBERT$_{senf}$	91.33	91.73	91.66	47.13
RoBERTa$_{senf}$	92.65	92.17	92.63	94.57
Weak ARI labels				
ChatGPT	89.69	89.46	88.53	94.71
Weakly Supervised ARI				
SciBERT$_{senf}$	92.32	90.71	90.70	94.47
RoBERTa$_{senf}$	92.14	90.12	90.46	94.76
SNet$_{jtl}$	92.17	91.13	91.16	94.22
Our$_{(SciBERT_{Senf})}$	91.10	91.87	91.95	97.08

Table 3: Micro-F1 scores for supervised and weakly-supervised ARI tasks on four datasets.

Models	**Neo**	**Gla**	**Mix**	**CDCP**
Weakly-supervised ACI				
Ours	**80.29**	**84.75**	**81.88**	**80.26**
- CoD (center)	79.57	84.32	81.38	79.56
- CoD	73.33	81.63	77.58	66.92
Weakly-supervised ARI				
Ours	**56.75**	**57.55**	**58.19**	**54.95**
- FM (LLM)	52.91	51.32	51.81	52.78
- FM (VC)	44.83	47.23	46.47	51.95

Table 4: Ablation results for weakly-supervised AM obtained on four datasets. “FM (LLM)” is the relation denoiser with only the LLM branch. “- FM (VC)” represents the relation denoiser with only the VC branch. We represent the center-based component detector as the “CoD”. “CoD (center)” is the CoD only with center-point branch. "ACI" and "ARI" stand for the argument component identification task and argument relation identification task, respectively.

4.5 Ablation Study

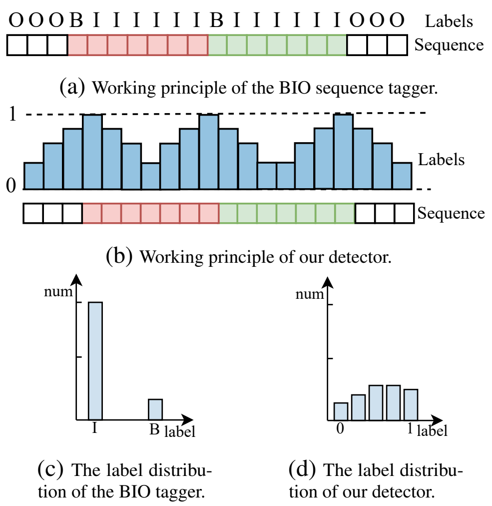
This section studies how the center-based component detector and relation denoiser affect model performance, respectively (results are in Tab. 4).

(1) When not employing our center-based component detector and instead using the previous method (Mayer et al., 2020; Stylianou and Vlahavas, 2021; Yeginbergenova and Agerri, 2023), the IoU of the model experiences a notable drop by 5.20, 4.43, 4.30, and 7.91 percentage points on the Neo, Gla, Mix, and CDCP datasets, respectively. This result indicates that our center-based component detector effectively contributes to improving argument boundary detection. To study the effectiveness of the center-point branch in our detector, we only use the predicted boundary masks to extract arguments. The performance drops across the four datasets ranges from 0.43 $\sim$ 0.72 in term of the IoU$_{75}$ score.

(2) For the weakly-supervised argument relation identification task, the removal of the LLM branch or VC branch in the fusion mechanism (FM) leads up to 6 and 12 points performance reduction in the macro-F1 scores across all datasets, respectively. The mechanism learns to blend the two branches, making them complementary to achieve better results on the argument relation identification task.

4.6 Analysis

This section studies the properties of the proposed models.



(a) Working principle of the BIO sequence tagger.

(b) Working principle of our detector.

(c) The label distribution of the BIO tagger.

(d) The label distribution of our detector.

Figure 3: (a) Illustrates the working principle of the BIO tagger. (b) Shows the working principle of our detector (c) Demonstrates the imbalanced label distribution obtained by the BIO tagger. (d) Shows a better balanced label distribution obtained by our method.

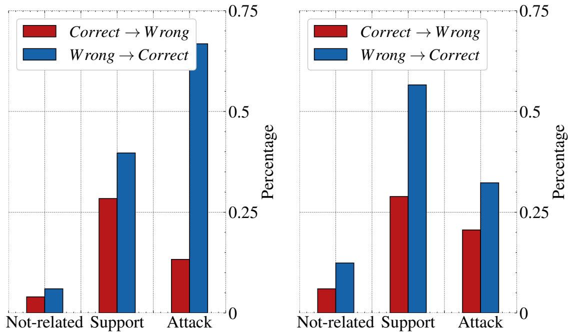
(1) First, we make comparison between our novel center-based component detector and the state-of-the-art detector in terms of the statistics of the obtained BIO labels. Fig 3 shows how our approach can obtain a better balanced label distributions compared to the BIO sequence tagger's label distribution. In order to demonstrate it, we compute the imbalance ratio (Thabtah et al., 2020), i.e., ratio of the number of samples in the majority class to the number of samples in the minority class, to measure the imbalance between argument

component B- and I-tokens on the AbstRCT-Neo dataset. This needs a conversion of the masks of center points and boundaries into B- and I-tokens. We regard a token as an I-token if the value of the predicted mask of center point in this position is higher than its boundary's value; otherwise this token is referred as a B-token. The imbalance ratio of the BIO tagger is 22.38, and the ratio of our detector is 1.189. Thus, the label distribution of our detector is better balanced compared with the distribution of the BIO tagger. To visualize the argumentative boundary detection of our detector, we provide an example to make a comparison between our detector and the state-of-the-art baseline, i.e., SciBERT$_{\text{gru-crf}}$ (Mayer et al., 2020), in the Tab. 5 Our approach successfully segments the input into two argument components, whereas the baseline wrongly identifies the whole text as one argument.

Baseline: Although further studies may need to confirm these data on a larger sample and to evaluate the side effect of increased iris pigmentation on long-term follow-up, in patients with pigmentary glaucoma, 0.005% latanoprost taken once daily was well tolerated and more effective in reducing IOP than 0.5% timolol taken twice daily.
Ours: Although further studies may need to confirm these data on a larger sample and to evaluate the side effect of increased iris pigmentation on long-term follow-up, in patients with pigmentary glaucoma, 0.005% latanoprost taken once daily was well tolerated and more effective in reducing IOP than 0.5% timolol taken twice daily.
GT: Although further studies may need to confirm these data on a larger sample and to evaluate the side effect of increased iris pigmentation on long-term follow-up, in patients with pigmentary glaucoma, 0.005% latanoprost taken once daily was well tolerated and more effective in reducing IOP than 0.5% timolol taken twice daily.

Table 5: The example shows the argumentative boundary detection abilities of our method and the baseline. Highlighted text with different color indicates different argument components.

(2) Second, we explore the correspondences and differences between predictions of the relation denoiser and the two weakly labeled data. Figure 4a shows the changes of predictions from the labels of LLM annotation to the predictions of our denoiser model. Figure 4b presents the changes of predictions from the labels of VC annotation to the predictions of our denoiser model. The flow from correct pseudo label predictions to incorrect predictions (red bars in both figures) helps us understand if the denoising model introduces errors even when the initial pseudo labels were correct. The flow from incorrect pseudo label predictions to correct predictions (blue bars in both figures) shows how well the model improves the correctness of incorrect pseudo labels. In both figures we observe that the flows of predictions from wrong to correct are stronger than the flows from correct to wrong. This shows our denoiser performs better by reducing errors and label noise from pseudo labels assigned by the LLM or VC annotation.



(a) Prediction changes of LLM annotation and our denoiser.

(b) Prediction changes of VC annotation and our denoiser.

Figure 4: The figures illustrate the prediction changes between the labels of weakly annotated resources (LLM or VC) and after applying the relation denoiser.

5 Conclusion

In this paper, we propose a novel weakly-supervised AM model to achieve performance comparable to fully-supervised AM models by leveraging limited human-annotated data. We leverage a LLM to provide weak labels for training samples of the argument component identification task and the argument relation identification task. Considering that weak labels generated by the LLM are noisy, we introduce two novel methods: a center-based component detector and a relation denoiser, to refine both the weak identification and weak labeling provided by the LLM. The center-based component detector refines the argument components' boundaries, and the relation denoiser reduces the noise in weakly labeled argument relation identification data. Experimental results on four widely used datasets indicate that our weakly supervised AM framework achieves new state-of-the-art performance in both AM tasks and significantly narrows the gap with fully supervised models. We believe our approach can be applied to other tasks, such as medical image segmentation (Wang et al., 2022) or nested named entity recognition (Lu et al., 2022), that require accurate boundary detection or face high annotation costs.

Limitation

The limitations of our paper are reflected as follows:

(1) Our models rely on the the weak labels provided by a LLM. We assume that for detecting the argumentative graph of a long document these labels might be too noisy to start from (Poudyal et al., 2020; Stab and Gurevych, 2017). In the future, we plan to explore methods to enhance the LLM's ability to provide effective weak labels for AM samples when dealing with document-level argumentative text.

(2) We only used few-shot in-context learning to obtain weak labels. In future work, we will employ more advanced ICL methods, such as CoT (Wei et al., 2022), PS-CoT (Wang et al., 2023a), and ToT (Yao et al., 2023), to obtain higher quality weak labels.

Ethics Statement

The datasets utilized in this paper are publicly available, anonymized, and devoid of sensitive information. An ethical concern arises from our dependence on large language models to provide weak labels for argument component and relation identification. These models, trained on extensive corpora, may potentially generate problematic or biased outputs.

References

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. 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)*, pages 6354–6364, Online. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2023. Multi-task attentive residual networks for argument mining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1877–1892.

Yong Guan, Jiaoyan Chen, Freddy Lecue, Jeff Pan, Juanzi Li, and Ru Li. 2023. Trigger-argument based explanation for event detection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5046–5058, Toronto, Canada. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2015. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2127–2137, Lisbon, Portugal. Association for Computational Linguistics.

Neslihan Iskender, Robin Schaefer, Tim Polzehl, and Sebastian Möller. 2021. Argument mining in tweets: Comparing crowd and expert annotations for automated claim and evidence detection. In *International Conference on Applications of Natural Language to Information Systems*, pages 275–288. Springer.

Hei Law and Jia Deng. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Wei Liu, Irtiza Hasan, and Shengcai Liao. 2020. Centerand scale prediction: A box-free approach for pedestrianand face detection. In *Computer Vision and Pattern Recognition*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020*, pages 2108–2115. IOS Press.

Tristan Miller, Maria Sukhareva, and Iryna Gurevych. 2019. A streamlined method for sourcing discourse-level argumentation annotations from the crowd. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1790–1796, Minneapolis, Minnesota. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver, Canada. Association for Computational Linguistics.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107.

Prakash Poudyal, Jaromír Šavelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. Echr: legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Nikolaos Stylianou and Ioannis Vlahavas. 2021. Transformed: End-to-end transformers for evidence-based medicine and argument mining in medical literature. *Journal of Biomedical Informatics*, 117:103767.

Fadi Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda Gonsalves. 2020. Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513:429–441.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Risheng Wang, Tao Lei, Ruixia Cui, Bingtao Zhang, Hongying Meng, and Asoke K Nandi. 2022. Medical image segmentation using deep learning: A survey. *IET Image Processing*, 16(5):1243–1267.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. 2020. Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1692–1705, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Kaiwen Wei, Yiran Yang, Li Jin, Xian Sun, Zequn Zhang, Jingyuan Zhang, Xiao Li, Linhao Zhang, Jintao Liu, and Guo Zhi. 2023. Guide the many-to-one assignment: Open information extraction via IoU-aware optimal transport. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4971–4984, Toronto, Canada. Association for Computational Linguistics.

Sun Wei, Mingxiao Li, Jingyuan Sun, Jesse Davis, and Marie-Francine Moens. 2024. DMON: A simple yet effective approach for argument structure learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5109–5118, Torino, Italia. ELRA and ICCL.

Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. *Approaches to Text Mining Arguments from Legal Cases*, pages 60–79. Springer Berlin Heidelberg, Berlin, Heidelberg.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Anar Yeginbergenova and Rodrigo Agerri. 2023. Cross-lingual argument mining in the medical domain. *arXiv preprint arXiv:2301.10527*.

Yu Zeng, Yunzhi Zhuge, Huchuan Lu, Lihe Zhang, Mingyang Qian, and Yizhou Yu. 2019. Multi-source weak supervision for saliency detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6074–6083.

Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023. Weaker than you think: A critical look at weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14229–14253, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Prompt Construction

The prompt is constructed in three parts: the system prompt, demonstration examples, and inputs.

(1) System Prompt The system prompts, denoted as p_{sys}, vary across different datasets. We consider an AM task with a label space for the argument component identification sub-task consisting of {"Claim", "Premise"}, and a label space for the argument relation identification sub-task consisting of {"Support", "Attack"}.

Argument Component Identification task description: *You are an AM system for argument detection. Find argument and classify them into, Claim, or Premise. Below are several examples:*

Argument Relation Identification task description: *You are an AM system for argument relation classification. Classify relations between arguments into, Support or Attack. Below are several examples:*

(2) Demonstration Prompts: Demonstration prompts p_{demo} consists of n annotated samples:

$$\{(p_{demo_1}, q_{demo_1}), \cdots, (p_{demo_n}, q_{demo_n})\},$$

where q_{demo_i} represents the ground-truth label for the i_{th} demonstration example. Both p_{demo_i} and q_{demo_i} vary across different tasks. Specifically, in the argument component identification task, p_{demo_i} is plain text, while q_{demo_i} consists of extracted argument components. Building on prior research (Wang et al., 2023b), we employ " @@" as the text separator to differentiate between various argument components within q_{demo_i}, as denoted by:

$$@@AC_i^1 \setminus n@@AC_i^2 \setminus n \cdots$$

where $\setminus n$ is the newline character. In argument relation identification task, p_{demo_i} is the extracted argument components and q_{demo_i} is a pairwise argument relation. q_{demo_i} is referred to:

$$@@AC_i^1@@ < relation > @@AC_i^2 \setminus n \cdots,$$

where $< relation >$ represents the argument relation between AC_i^1 and AC_i^2.

We select demonstration examples from a golden-annotated benchmark development set. Regarding the criterion for example selection, we adhere to the methodology outlined in previous work (Min et al., 2022) and choose demonstration examples whose label space encompasses that of the test set. To ensure similarity, we represent the i-th demonstration example as the string <demo>$_i$:

$$\{\setminus n; \text{Input: } p_{demo_i}; \setminus n; \text{Output: } q_{demo_i}; \setminus n\},$$

(3) Input: Input for LLMs p_{input} are the concatenation of corresponding system prompt p_{sys}, demonstration prompts p_{demo}, and test sequence p_{test}. The input sequence is:

$$\{p_{sys}; \text{<demo>}_1; \ldots; \text{<demo>}_n; p_{test}\}.$$