Argument Mining with LLaMA 8B

Anonymous ACL submission

Abstract

An end-to-end argument mining (AM) pipeline takes a text as input and provides the argumentative structure of this text as output, by identifying and classifying the argument units and relations within it. In this work, we focus on LLM fine-tuning approach to AM. We model the three sub-tasks of the AM pipeline as text generation tasks. We fine-tune classical and quantized versions of LLaMA–3, the most capable open-source model available, on the benchmark Persuasive Essays (PE) dataset. We consider various contextual and structural fine-tuning modalities, where the AM sub-tasks are modeled either at the paragraph or at the essay level, with or without inclusion of additional markup tags. We achieve state-of-the-art results on all three sub-tasks, with significant improvements over previous benchmarks.

1 Introduction

Argument Mining (AM) involves automatically analysing and parsing of the argumentative structure of natural language texts from diverse sources (Palau and Moens, 2009; Cabrio and Villata, 2018). A complete AM pipeline takes a text as input, identifies and classifies the argument units and relations within it, and provides the text's argumentative structure as output. AM sub-tasks include: (1) identifying argument components in the text (ACS), (2) classifying argument components according to their argumentative roles (ACC), (3) identifying argument relations between argument components (ARI) and (4) classifying the stance of the argument relations (ARC) (Stab and Gurevych, 2017).

Initial approaches to AM utilized traditional supervised machine learning algorithms, such as Maximum Entropy Classifiers (Mochales and Moens, 2011), Logistic Regressions (Levy et al., 2014) and Support Vector Machines (Stab and Gurevych, 2017; Habernal and Gurevych, 2017). Subsequent studies employ more advanced neural network-based models, like Recurrent Neural Networks (RNNs) (Eger et al., 2017; Niculae et al., 2017) and LSTMs/BiLSTMs (Haddadan et al., 2019; Potash et al., 2017; Mayer et al., 2020; Kuribayashi et al., 2019). These investigations convey two core messages: (i) the centrality of incorporating additional task-specific contextual, structural, and syntactic features in the models, as the text of the argument units and relations alone is insufficient for accurately predicting their argumentative roles, and (ii) the importance of capturing the global sequentiality of the argumentative and discursive flow in the text.

Large Language Models (LLMs) are the dominant contemporary paradigm in NLP (Zhao et al., 2023). These models employ transformer-based architectures and undergo pre-training on vast amounts of data, which enables them to grasp general-purpose language patterns (Vaswani et al., 2017). LLMs have demonstrated outstanding performance across various NLP tasks and exhibited significant emergent capabilities (Wei et al., 2022).

Reflecting the popularity of transformer-based architectures, Mushtaq and Cabessa, 2022, 2023 present customized BERT-based models for ACC that incorporate contextual, structural, and syntactic features provided as text rather than numerically. Moreover, Bao et al., 2021 jointly model the ACC and ARI tasks using a transition-based BERT-BiLSTM architecture.

In the realm of generative LLMs, AM has been reframed as a text generation tasks. Pojoni et al., 2023 use GPT–4 for argument mining in transcribed podcasts using two specially designed prompt templates, one more fine-grained than the other. Similarly, Al Zubaer et al., 2023 approach ACC in the legal domain as text generation using GPT–3.5 and GPT–4. Liu et al., 2023 incorporate the chain of thought (CoT) technique to their BART-Base based 'AM as text generation' model. For every AM sub-task, in addition to the class label ('MajorClaim'/'Claim'/'Premise', 'Relation'/'No-Relation', 'Support'/'Attack'), their model also generates a path from the root component to the query component as demonstration of the model's reasoning.

LLMs are commonly utilized for downstream tasks through two techniques: training-free, whereby the pre-trained LLM is used 'as is' for downstream tasks (typically using a prompt), and the more rigorous fine-tuning, whereby the pre-trained LLM is further trained on suitable task-specific data. In-Context Learning (ICL) is a training-free technique where the LLM is conditioned solve tasks by providing a few solved demonstration examples in the prompt, precluding the need for further fine-tuning. Interestingly, Nori et al., 2023 show that an ICL approach with GPT (OpenAI, 2023), LLaMA (Touvron et al., 2023) and Qwen (Bai et al., 2023) outperforms the fine-tuning approach for several NLP tasks. For Argument Mining, however, Cabessa et al., 2024 show that further LLM fine-tuning is required for optimally capturing the argumentative flow and sequentiality of argument components and relations.

Our work is situated within this LLM fine-tuning approach to AM. We model the ACC, ARI and ARC sub-tasks of the AM pipeline as text generation tasks using Meta's LLaMA 3, the most capable open-source model available. We fine-tune classical and quantized versions of LLaMA–3–8B on the benchmark Persuasive Essays (PE) dataset. We consider various contextual and structural fine-tuning modalities, where the AM tasks are modeled either at the paragraph or at the essay level, with or without inclusion of additional markup tags. We achieve state-of-the-art results on all three sub-tasks of the AM pipeline, with significant improvements over previous benchmarks. Our code is freely available on GitHub.

2 Methodology

2.1 Dataset

We use the benchmark Persuasive Essays (PE) dataset introduced by (Stab and Gurevych, 2017). The PE dataset consists of 402 structured essays on various topics. The train and test sets are composed of 322 and 80 essays, respectively. The statistics of the PE dataset are given in Table 1.

Corpus Statistics		Component Statistics	
Tokens	147,271	major claims	751
Sentence	7,116	claims	1,506
Paragraphs	1,833	premises	3,832
Essays	402	Total	6,089

Table 1: PE dataset statistics.

The ACC task consists of classifying each argument component (AC) as either 'MajorClaim', 'Claim' or 'Premise'. The ARI task involves classifying each argument relation (AR) of a paragraph as either 'Related' or 'Non-related'. For each paragraph, we consdier all ARs of the form (AC_i, AC_j) for classification. The ARC task consists of classifying each related argument relation (AC_i, AC_j) as either 'Support' or 'Attack'.

2.2 Fine-tuning modalities

Fine-tuning (FT) refers to the process of further training a pre-trained LLM on a specific downstream task. LLMs with billions of parameters can be fine-tuned efficiently using the QLoRA strategy, which employs a frozen n-bit quantized version of the pre-trained weights and trains rank decomposition matrices (low rank adapters) of the model's layers (Dettmers et al., 2023).

As in other works, we assume that the first task of the AM pipeline (ACS) has already been performed. As a result, the argument components are delimited by tags of the form <AC0>...</AC0>, <AC1>...</AC1>, <AC2>...</AC2>, etc. We address the subsequent ACC, ARI and ARC tasks using fine-tuned LLaMA–3 models. More specifically, the ACC, ARI and ARC tasks are reformulated as text generation tasks, where the list of argument component types (e.g. ['MajorClaim', 'Claim', 'Claim', ...]), the list of pairs of related argument components (e.g. $[(0, 1),(0, 2),(1, 2), \dots]$), and the list of argument relation types (e.g. ['Support', 'Support', 'Attack', ...]) are generated by LLaMA-3, respectively. The following fine-tuning modalities incorporating different contextual and structural information are considered:

- **Paragraph/Essay level:** The LLM is trained and tested on data samples consisting of either individual essay paragraphs or full essays, respectively.

- **With/Without structural tags:** Markup tags delimiting the topic (<topic>...</topic>), introduction (<para-intro>...</para-

intro>), body paragraphs (<para-body>. . . </para-body>) and conclusion (<para-conclusion>. . . </para-conclusion>) of the essays can be inserted in the train and test samples. For the ARI and ARC tasks, the argument components' types can further be given as tags of the form <ACn, MajorClaim>. . . </ACn, MajorClaim>, <ACn, Claim>. . . </ACn, Claim>, or <ACn, Premise>. . . </ACn, Premise>.

Several examples of dataset samples at the paragraph or essay levels, with or without tags, are provided in Appendix B. Implementations detailed are given in Appendix A.

3 Results

We present the detailed results of our experiments in Table 3. We compare our results with the common baselines in AM as well as with the state-of-the-art (SOTA) models (see Table 2).

Argument Component Classification (ACC): We achieve a state-of-the-art result on this task with a macro F1 of 89.5, compared to the previous SOTA score of 89.2 (see Table 2). We also ran the bigger quantized model Llama–3–70b–bnb–4bit and obtained a macro F1 of 89.2, which is on par with SOTA.

Previous results indicate that capturing the argumentative sequentiality of ACs is essential for achieving good performance on the ACC task. Both paragraph and essay modalities enable the grasping of this sequentiality, though at different scales. There is no clear pattern indicating which contextual scale performs best. We conjecture that the essay level is beneficial for this task, as the argumentative flow extends throughout the entire essay.

Generally, the consideration of structural features helps in predicting the AC types (e.g., major claims tend to appear more frequently in introduction and conclusion paragraphs). Here, the injection of markup tags seems to improve results at the paragraph level, but not at the essay level. The structural information conveyed by the tags would thus be able to boost performance in the case of limited contextual scale.

Argument Relation Identification (ARI): In its original formulation, the ARI task involves the identification of argument relations within paragraphs, by identifying the related pairs of ACs among all possible ones. Naturally, even if rephrased as a text generation task, solving the ARI task at the global essay level remains more challenging than at the more local paragraph level. These considerations explain the significantly lower scores obtained at the essay level.

At the paragraph level, the addition of markup tags drastically improves the results. An ablation study has revealed the importance of structural tags (e.g., <para-intro>. . . </para-intro>) and AC type tags (e.g., <AC0, MajorClaim>. . . </AC0, MajorClaim>). First, we note that the injection of structural tags alone is sufficient to achieve SOTA results (83.5). In this case, the models most probably learned that introduction, body, and conclusion paragraphs are associated to different patterns of argument relations, and was able to leverage this information to improve its performance. Secondly, the injection of AC type tags alone drastically boosts the results (92.8). Clearly, the related/non-related nature of ARs strongly depends on the types of their constituent ACs, and the model was able to learn and exploit this information. In a real-life AM pipeline, these true AC types, which are unknown, could be replaced by the predictions of the previous ACC task to enhance the model's performance. Finally, the combination of both tags further improves the results to 93.7. Note that this score represents a drastic improvement over previous SOTA result (82.7, see Table 2).

Argument Relation Classification (ARC): The ARC task is also modelled by definition at the paragraph level, which explains the significantly poorer results obtained at the essay level.

At the paragraph level, we achieve state-of-the-art results on this task too, with an F1 score of 89.6, which represents a drastic improvement over the previous SOTA of 81.0 (see Table 2). In this case, there is no clear evidence indicating whether the addition of tags improves the results. We nevertheless conjecture that tag injection plays a marginal role. Indeed, once ARs are identified, their supporting or attacking nature primarily depends on the textual content of their constituent ACs, and less significantly on the types of these ACs or the types of paragraphs in which they are located.

Joint ACC–ARI–ARC Task Since the ARI and ARC tasks are modeled by definition at the paragraph level, and reinforced by the low accuracy obtained for these tasks at the essay level, we evaluated the joint ACC–ARI–ARC task at the para-

Model	ACC	ARI	ARC
SVM-ILP (Stab and Gurevych, 2017)	82.6	75.1	68.0
Joint-PN (Potash et al., 2017)	84.9	76.7	-
BiLSTM-MINUS (single task) (Kuribayashi et al., 2019)	85.6	78.3	79.6
BiLSTM-MINUS (joint tasks) (Kuribayashi et al., 2019)	87.3	81.1	79.0
BERT-Trans (Bao et al., 2021)	88.4	82.5	**81.0**
BERT-MINUS-FeaTxt (Mushtaq and Cabessa, 2023)	83.1	–	–
GPT–4 In-Context Learning (ICL) (Cabessa et al., 2024)	83.6	–	–
MRC-GEN (Liu et al., 2023)	**89.2**	**82.7**	78.2

Table 2: Macro F1 scores of ACC, ARI and ARC tasks obtained by previous baselines and benchmark models on the PE dataset. The state-of-the-art results (before our study) are highlighted in boldface.

Model	Mode		AM tasks			
	context	tags	ACC	ARI	ARC	ACC – ARI – ARC
Llama-3-8b-bnb-4bit	paragraph	0	87.7	81.2	80.0	87.5 – 80.7 – 82.9
Llama-3-8b	paragraph	0	86.3	81.0	**89.6**	88.3 – 80.9 – 79.9
Llama-3-8b-bnb-4bit	paragraph	1	88.2	**83.5** / 92.8 / 93.5	86.8	87.9 – 80.6 – 79.9
Llama-3-8b	paragraph	1	87.3	83.0 / 92.5 / **93.7**	89.1	88.1 – 80.6 – 77.2
Llama-3-8b-bnb-4bit	essay	0	86.8	65.6	64.0	–
Llama-3-8b	essay	0	**89.5**	49.7	64.0	–
Llama-3-8b-bnb-4bit	essay	1	87.0	77.0	71.5	–
Llama-3-8b	essay	1	86.3	77.1	64.3	–

Table 3: Results of the ACC, ARI and ARC tasks obtained by Llama–3–8B and it's 4-bit quantized version, with various fine-tune modes. For ARI task with mode 'tags=1', the results $x/y/z$ correspond to the three ablation settings where: only the paragraph tags are provided, only the AC type tags are provided or both the paragraph and the AC type tags are provided, respectively. State-of-the-art results are highlighted in boldface.

graph level. In line with Kuribayashi et al., 2019, Bao et al., 2021 and Liu et al., 2023, the joint task modelling does not yield any significant improvements and generally harms the individual task performance. For ACC, the joint task approach either performs on par with or slightly improves over single-task modeling in equivalent modalities. For ARI and ARC, on the other hand, the joint task modeling achieves significantly lower performance compared to their single task counterparts. Overall, these results reflect more closely the performance of a real life AM pipeline than the single task setting. Note that the evaluation of ARC is over overestimated in this joint task generative setting (see Section 4 for further explanations).

4 Conclusion

In this work, we address the three main tasks of the AM pipeline using several interesting fine-tuning modalities. These fine-tuning modalities are designed to capture contextual information at different scales (paragraph level or essay level) as well as structural information (paragraph types and AC types) in the form of markup tags. We use Llama–3–8B and its 4-bit quantized version to achieve state-of-the-art results on all three tasks. For the ARI and ARC tasks, our results represent a major improvement over the previous ones. Overall, our study demonstrates the strong abilities of LLMs to capture argumentative discourse and reasoning patterns in natural texts.

For future work, we plan to investigate the AM sub-tasks using other popular LLMs to better understand the extent to which the model's size influences task performance. A thorough investigation of the models' attention heads could also shed light on the precise role that contextual information and structural tags play in the results. Finally, to better understand the internal reasoning process of LLMs, we plan to study the ability of LLMs to generate complete argumentative structures using Chain-of-Thought (CoT) techniques.

Limitations

We obtained state-of-the-art results on all three AM sub-tasks, with strong improvements over previous benchmarks. However, for obvious computational limitation reasons, we haven't run repeated sets of experiments for each task to examine the means and standard deviations of the models' performance. We also experimented with the bigger 4-bit quantized llama-70B model, and we couldn't establish a clear pattern relating the model's size and performance. Therefore, we cannot assert how increasing the LLM size correlates with the obtained performance.

We experiment with the benchmark Persuasive Essays dataset, which consists of reasonably well-structured text. While we believe that other domains with similar textual modalities, such as legal texts, will also benefit from our approach, we are curious about how LLMs will generalize to less structured domains like news articles, speeches, and social media content.

On a broader, philosophical level, we find it fitting to comment on the emerging research trends in AI and Machine Learning. Fine-tuning increasingly larger generative models appears to outperform complex, well-designed, and richer yet smaller architectures. With the ongoing 'arms race' among AI companies to produce ever larger models, it seems natural to ask: will compute power overtake pure research?

As a final technical remark, note that our evaluation of ARC in the joint task setting overestimates the score for this task. More precisely, assuming that the ground truth lists of tagged ARs is

$$[[0, 1, \text{`Support'}], [0, 2, \text{`Support'}], [1, 2, \text{`Attack'}]]$$

and that the models generated the corresponding list

$$[[0, 3, \text{`Support'}], [0, 4, \text{`Support'}], [1, 2, \text{`Support'}]]$$

then we counted the two first predictions 'Support', 'Support' as correct, although they are related to incorrect ARs. We invite the reader to examine the code to understand how we handled cases where the ground truth and predicted lists are of different lengths.

References

Abdullah Al Zubaer, Michael Granitzer, and Jelena Mitrović. 2023. Performance analysis of large language models in the domain of legal argument mining. *Frontiers in Artificial Intelligence*, 6.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6354–6364. Association for Computational Linguistics.

Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. 2024. In-context learning and fine-tuning gpt for argument mining. *Preprint*, arXiv:2406.06699.

Elena Cabrio and Serena Villata. 2018. Five years of argument mining: A data-driven analysis. In *Proceedings of IJCAI 2018*, IJCAI'18, pages 5427–5433. AAAI Press.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of ACL 2017*, pages 11–22, Vancouver, Canada. ACL.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. Yes, we can! mining arguments in 50 years of US presidential campaign debates. In *Proceedings of ACL 2019*, pages 4684–4690, Florence, Italy. ACL.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In *Proceedings of ACL 2019*, pages 4691–4698, Florence, Italy. ACL.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *ICCL*.

Boyang Liu, Viktor Schlegel, Riza Batista-Navarro, and Sophia Ananiadou. 2023. Argument mining as a multi-hop generative machine reading comprehension task. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10846–10858. Association for Computational Linguistics.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *Proceedings of ECAI 2020*, volume 325 of *FAIA*, pages 2108–2115. IOS Press.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Umer Mushtaq and Jérémie Cabessa. 2022. Argument classification with BERT plus contextual, structural and syntactic features as text. In *Proceedings of ICONIP 2022*, volume 1791 of *CCIS*, pages 622–633. Springer.

Umer Mushtaq and Jérémie Cabessa. 2023. Argument mining with modular BERT and transfer learning. In *Proceedings of IJCNN 2023*, pages 1–8. IEEE.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of ACL 2017*, pages 985–995, Vancouver, Canada. ACL.

H. Nori et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *CoRR*, abs/2311.16452.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of ICAIL 2019*, ICAIL '09, pages 98–107, New York, NY, USA. ACM.

Mircea-Luchian Pojoni, Lorik Dumani, and Ralf Schenkel. 2023. Argument-mining from podcasts using chatgpt. In *Proceedings of ICCBR-WS 2023*, volume 3438 of *CEUR Workshop Proceedings*, pages 129–144. CEUR-WS.org.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of EMNLP 2017*, pages 1364–1373. ACL.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS 2017*, pages 5998–6008.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Implementation details

All experiment were carried out using the LLaMA-Factory python library (Zheng et al., 2024). We trained the models llama-3-8b-Instruct-bnb-4bit, llama-3-8b-Instruct and llama-3-70b-Instruct-bnb-4bit freely available from Hugging Face. We used the default hyper-parameters of the LLaMA-Factory (no hyperparameter tuning) and trained the models for 10 epochs on a single NVIDIA RTX A6000 (48GB) GPU. The average training and inference time of the PE dataset at the paragraph and essay levels was approximately 2 and 1.5 hours, respectively. Our code is freely available on GitHub.

B Prompts

We provide test prompts of different modalities (paragraph/essay level, with/without tags) used for each sub-tasks ACC, ARI and ARC. The training samples are of the same kind, but with the answers to the task added at the end.

Example 1. ACC task, essay level, with tags.

### You are an expert in Argument Mining. You are given an essay which contains numbered argument components enclosed by <AC></AC> tags. Your task is to classify each argument components in the essay as either 'MajorClaim', 'Claim' or 'Premise'. You must return a list of argument component types in following JSON format: 'component_types': [component_type (str), component_type (str), . . . , component_type (str)]

### Here is the essay text: <topic> Should students be taught to compete or to cooperate ? </topic><para-intro> It is always said that competition can effectively promote the development of economy . In order to survive in the competition , companies continue to improve their products and service , and as a result , the whole society prospers . However , when we discuss the issue of competition or cooperation , what we are concerned about is not the whole society , but the development of an individual ' s whole life . From this point of view , I firmly believe that <AC0> we should attach more importance to cooperation during primary education </AC0> . </para-intro><para-body> First of all , <AC1> through cooperation , children can learn about interpersonal skills which are significant in the future life of all students </AC1> . <AC2> What we acquired from team work is not only how to achieve the same goal with others but more importantly , how to get along with others </AC2> . <AC3> During the process of cooperation , children can learn about how to listen to opinions of others , how to communicate with others , how to think comprehensively , and even how to compromise with other team members when conflicts occurred </AC3> . <AC4> All of these skills help them to get on well with other people and will benefit them for the whole life </AC4> . </para-body><para-body> On the other hand , <AC5> the significance of competition is that how to become more excellence to gain the victory </AC5> . Hence it is always said that <AC6> competition makes the society more effective </AC6> . However , <AC7> when we consider about the question that how to win the game , we always find that we need the cooperation </AC7> . The greater our goal is , the more competition we need . <AC8> Take Olympic games which is a form of competition for instance , it is hard to imagine how an athlete could win the game without the training of his or her coach , and the help of other professional staffs such as the people who take care of his diet , and those who are in charge of the medical care </AC8> . The winner is the athlete but the success belongs to the whole team . Therefore <AC9> without the cooperation , there would be no victory of competition </AC9> . </para-body><para-conclusion> Consequently , no matter from the view of individual development or the relationship between competition and cooperation we can receive the same conclusion that <AC10> a more cooperative attitudes towards life is more profitable in one ' s success </AC10> . </para-conclusion>

Example 2. ARI task, paragraph level, without tags. Note that the ACs are still delimited by tags (<ACn>. . . <ACn>) as result of the first segmentation task ACS.

### You are an expert in Argument Mining. You are given a paragraph which contains argument components enclosed by <AC></AC> tags. Your task is to identify argument relations between argument components in the paragraph. You must return a list of argument component pairs in following JSON format: 'list_argument_relations': [[target AC (int), source AC (int)], . . . , [target AC (int), source AC (int)]]

### Here is the paragraph text: First of all , <AC0> to obtain information , using the internet is quicker and more convenient than reading newspapers </AC0> . <AC1> Contrary to the past when people had to wait long hours to take a daily newspaper , nowadays , they can acquire latest news updated every second through their mobile phones or computers connected to the internet , everywhere and at anytime </AC1> . <AC2> As can be seen , these devices and machines are very common in all parts of the world , making it easier for people to read a number of things that newspapers can not provide in only some pages </AC2> . Hence , <AC3> the print media has failed to keep its important role in the provision of information </AC3>.

Example 3. ARC task, essay level, with tags.

### You are an expert in Argument Mining. You are given a paragraph which contains argument components enclosed by <AC></AC> tags. You are also given a list of pairs of related argument components in the form: [(target AC (int), source AC (int)), (target AC (int), source AC (int)), . . . , (target AC (int), source AC (int))]. Your task is to classify each pair of related argument components in the list as either 'Support' or 'Attack'. You must return a list of relation types in following JSON format: 'relation_types': [relation_type (str), relation_type (str), . . . , relation_type (str)]

### Here is the paragraph text: <topic> Should students be taught to compete or to cooperate ? </topic><para-intro> It is always said that competition can effectively promote the development of economy . In order to survive in the competition , companies continue to improve their products and service , and as a result , the whole society prospers . However , when we discuss the issue of competition or cooperation , what we are concerned about is not the whole society , but the development of an individual ' s whole life . From this point of view , I firmly believe that <AC0, MajorClaim> we should attach more importance to cooperation during primary education </AC0, MajorClaim> . </para-intro><para-body> First of all , <AC1, Claim> through cooperation , children can learn about interpersonal skills which are significant in the future life of all students </AC1, Claim> . <AC2, Premise> What we acquired from team work is not only how to achieve the same goal with others but more importantly , how to get along with others </AC2, Premise> . <AC3, Premise> During the process of cooperation , children can learn about how to listen to opinions of others , how to communicate with others , how to think comprehensively , and even how to compromise with other team members when conflicts occurred </AC3, Premise> . <AC4, Premise> All of these skills help them to get on well with other people and will benefit them for the whole life </AC4, Premise> . </para-body><para-body> On the other hand , <AC5, Premise> the significance of competition is that how to become more excellence to gain the victory </AC5, Premise> . Hence it is always said that <AC6, Claim> competition makes the society more effective </AC6, Claim> . However , <AC7, Premise> when we consider about the question that how to win the game , we always find that we need the cooperation </AC7, Premise> . The greater our goal is , the more competition we need . <AC8, Premise> Take Olympic games which is a form of competition for instance , it is hard to imagine how an athlete could win the game without the training of his or her coach , and the help of other professional staffs such as the people who take care of his diet , and those who are in charge of the medical care </AC8, Premise> . The winner is the athlete but the success belongs to the whole team . Therefore <AC9, Claim> without the cooperation , there would be no victory of competition </AC9, Claim> . </para-body><para-conclusion> Consequently , no matter from the view of individual development or the relationship between competition and cooperation we can receive the same conclusion that <AC10, MajorClaim> a more cooperative attitudes towards life is more profitable in one ' s success </AC10, MajorClaim> . </para-conclusion>

###Here is the list of pairs of related argument components in this paragraph: [(0, 1), (0, 2), (0, 3), (1, 0), (4, 2), (4, 3)]

Example 4. Joint ACC-ARI-ARC task, paragraph level, with tags.

### You are an expert in Argument Mining. You are given a paragraph which contains argument components enclosed by <AC></AC> tags. Your task is to classify the argument components as well as to identify and classify argument relations between argument components in the paragraph. For each argument component, its AC type (str) is either 'MajorClaim', 'Claim' or 'Premise'. For each argument relation (target AC (int), source AC (int)), its link type (str) is either 'Support' or 'Attack'. You must return two lists in following JSON format: "list_component_types": [AC type (str), ..., AC type (str)], "list_argument_relations_and_types": [[target AC (int), source AC (int), link type (str)], ..., [target AC (int), source AC (int), link type (str)]]

### Here is the paragraph text: <para-body> <AC0> Taking care of thousands of citizens who suffer from disease or illiteracy is more urgent and pragmatic than building theaters or sports stadiums </AC0> . As a matter of fact , <AC1> an uneducated person may barely appreciate musicals , whereas a physical damaged person , resulting from the lack of medical treatment , may no longer participate in any sports games </AC1> . Therefore , <AC2> providing education and medical care is more essential and prioritized to the government </AC2> . </para-body>