Turiya at DialAM-2024: Inference Anchoring Theory Based LLM Parsers

Sougata Saha and Rohini Srihari
State University of New York at Buffalo
Department of Computer Science and Engineering
{sougatas, rohini}@buffalo.edu

Abstract

Representing discourse as argument graphs facilitates robust analysis. Although computational frameworks for constructing graphs from monologues exist, there is a lack of frameworks for parsing dialogue. Inference Anchoring Theory (IAT) is a theoretical framework for extracting graphical argument structures and relationships from dialogues. Here, we introduce computational models for implementing the IAT framework for parsing dialogues. We experiment with a classification-based biaffine parser and Large Language Model (LLM)-based generative methods and compare them. Our results demonstrate the utility of finetuning LLMs for constructing IAT-based argument graphs from dialogues, which is a nuanced task.

1 Introduction

Argumentation is prevalent in our daily verbal communication and represents chains of thought patterns and reasoning, making it an integral mode of persuasion (Saha et al., 2022a). Although argument mining (AM) (Stab and Gurevych, 2014a,b; Persing and Ng, 2016; Stab and Gurevych, 2017; Nguyen and Litman, 2018; Eger et al., 2017; Mirko et al., 2020; Morio et al., 2020; Lawrence and Reed, 2020; Ye and Teufel, 2021; Bao et al., 2021; Saha et al., 2022a) from monologues is well studied, formal models for parsing dialogues are lacking (Saha et al., 2022b). DialAM-2024 (Ruiz-Dolz et al., 2024) introduced the first shared task in dialogue argument mining, where argumentation and dialogue information are modeled jointly in the domain-independent IAT framework (Budzynska et al., 2014, 2016; Janier et al., 2014). The framework represents dialogues as a graph where the nodes comprise (i) Locutions (*L-nodes*)-the Argumentative Discourse Units (ADUs) from each speaker turn. (ii) Propositions (*I-nodes*)-reconstructed *L-nodes* with resolved anaphora, pronouns, and deixis, making them independently coherent. The edges comprise (i) Default Transitions (*TAs*) between *L-nodes*. (ii) *S-nodes* that connect propositions (*I-nodes*) and can be of types RA (default inference), MA (default rephrase), or CA (default conflict). (iii) *YA-nodes* that connect *L-nodes* with *I-nodes*, *TAs* with *S-nodes*, or *TAs* with *I-nodes*.

Here, we compare generative approaches against classification-based approaches for implementing the IAT framework. Since LLMs (Chang et al., 2023; Min et al., 2023; Hadi et al., 2023) attain superior results on several tasks, we test their utility in dialogical argument mining and compare them against a biaffine-parsing-based implementation (Dozat and Manning, 2016, 2018). We ask the following research questions: **(i) Can LLMs be used for parsing dialogues in the IAT framework?** We experiment with Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) and present dialAM as a generative task where the *L-nodes*, *I-nodes*, and *TA-nodes* are the context of the LLM, and the task comprises determining the propositional (Task A) and illocutionary (Task B) relations. **(ii) How do LLMs compare against simpler classification-based dialogue parsers?** We compare the LLM parser against a biaffine-parsing-based parser that predicts the relationship and type between nodes.

2 Proposed Method

2.1 Classification-Based Model

As illustrated in Figure 1, the classifier is Roberta-based (Liu et al., 2019) and contains two biaffine layers, each comprising two biaffine heads, which predict the relationships and their types. Biaffine classifiers (Dozat and Manning, 2016, 2018) are generalizations of linear classifiers, which include multiplicative interactions between two vectors. The first biaffine layer determines the *S-nodes* and labels the relationships between *L - I-nodes* and *TA - I-nodes*. The second layer determines and labels relationships between the *TA* and *S-nodes*.

Proceedings of the 11th Workshop on Argument Mining, pages 124–129
August 15, 2024 ©2024 Association for Computational Linguistics

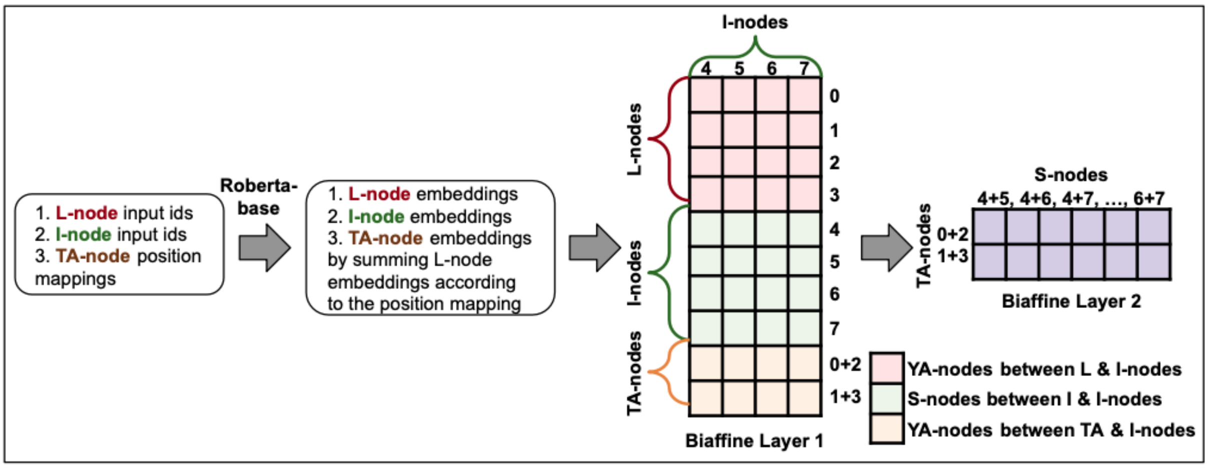


Figure 1: Architecture of the Roberta-based biaffine classifier.

The model inputs the *L*, *I-node* input ids, and the *TA-node* position mappings- A set of pairs of *L-node* indexes that form a *TA-node*. First, the parser independently encodes all *L* and *I-nodes* and then performs multi-headed attention between the embeddings (sum-pooled representation of the transformer last layer). The *TA-node* position mappings are sum-pooled to yield the *TA-node* embeddings. Then, the *L*, *I*, and *TA-node* embeddings are passed through a single-layer feed-forward neural network (FF) to generate the source representation of the biaffine heads of the first biaffine layer. The FF layer reduces the input representation from 768 dimensions to 600. Another single-layered FF computes a 600-dimensional representation of the *I-node* embeddings, and are the targets of the biaffine heads.

Since *S-node* prediction is a pre-requisite for determining relationships between the TA and *S-nodes*, two subsequent biaffine heads determine and label their relationship. We generate pairs of all possible *I-nodes*, sum-pool their embeddings, and weigh them by the predicted logits from the *S-node* relationship biaffine head. A single-layered FF computes the final 600-dimensional representation, which is the target of the biaffine heads. The source of the biaffine heads is the prior computed 600-dimensional representation of the *TA-nodes*. During inference, we only consider relationship labels with a predicted probability > 0.1 and persist the highest scored relationships such that a node is referenced only once.

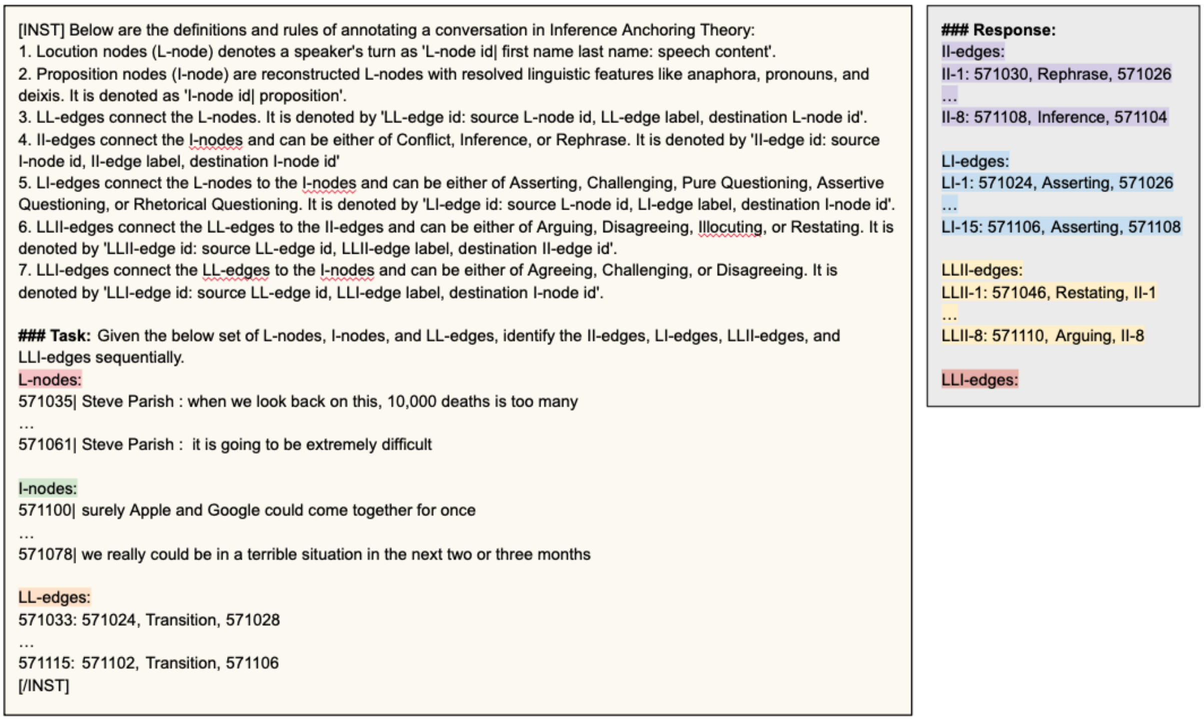
2.2 Generative Model

To determine the utility of using LLMs for argument mining, we experiment with Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) using two types of instructions. We pose the problem as a generation task where the model is presented with a description of the IAT annotation framework, the *L*, *I*, and *TA-nodes* and tasked to identify the *S* and *YA-nodes* sequentially. The ordering of the *L* and *I-nodes* are randomized to prevent the model from learning spurious ordering-based associations. Figures 2 and 3 illustrate Type 1 and 2 instructions with the model-generated responses. Compared to Type 1, the Type 2 instructions explicitly include all the relation labels in the response text.

2.3 Training Details

The dataset comprised 1,477 IAT annotated conversations from the QT30 corpus (Hautli-Janisz et al., 2022). We preprocess the corpus and preserve relations that strictly adhere to the IAT relationship definitions and discard others. Although training on the reduced examples might hamper performance, it eases answering the research questions- our primary objective. We randomly split the dataset into 1,327 training and 150 testing examples for our experiments. The classification-based model used Roberta-base as the base model and was trained for 15 epochs, processing 32 batches at a time, using a 5e-5 learning rate, and accumulating gradients for 16 steps. The LLM-based models were trained for two epochs using LoRA (Hu et al., 2021), a parameter-efficient fine-tuning method (Mangrulkar et al., 2022). The LoRA r and alpha were set to 16 and 32 and trained the q, v, k, o, gate, up, and down projection modules of the attention heads and the LM head using a 2.5e-5 learning rate. We use greedy decoding during inference and limit the number of new tokens to 1,000.

While evaluating the results of Type 1 and 2 in-



[INST] Below are the definitions and rules of annotating a conversation in Inference Anchoring Theory:
1. Locution nodes (L-node) denotes a speaker's turn as 'L-node id| first name last name: speech content'.
2. Proposition nodes (I-node) are reconstructed L-nodes with resolved linguistic features like anaphora, pronouns, and deixis. It is denoted as 'I-node id| proposition'.
3. LL-edges connect the L-nodes. It is denoted by 'LL-edge id: source L-node id, LL-edge label, destination L-node id'.
4. II-edges connect the I-nodes and can be either of Conflict, Inference, or Rephrase. It is denoted by 'II-edge id: source I-node id, II-edge label, destination I-node id'
5. LI-edges connect the L-nodes to the I-nodes and can be either of Asserting, Challenging, Pure Questioning, Assertive Questioning, or Rhetorical Questioning. It is denoted by 'LI-edge id: source L-node id, LI-edge label, destination I-node id'.
6. LLII-edges connect the LL-edges to the II-edges and can be either of Arguing, Disagreeing, Illocuting, or Restating. It is denoted by 'LLII-edge id: source LL-edge id, LLII-edge label, destination II-edge id'.
7. LLI-edges connect the LL-edges to the I-nodes and can be either of Agreeing, Challenging, or Disagreeing. It is denoted by 'LLI-edge id: source LL-edge id, LLI-edge label, destination I-node id'.

### Task: Given the below set of L-nodes, I-nodes, and LL-edges, identify the II-edges, LI-edges, LLII-edges, and LLI-edges sequentially.
L-nodes:
571035| Steve Parish : when we look back on this, 10,000 deaths is too many
...
571061| Steve Parish : it is going to be extremely difficult

I-nodes:
571100| surely Apple and Google could come together for once
...
571078| we really could be in a terrible situation in the next two or three months

LL-edges:
571033: 571024, Transition, 571028
...
571115: 571102, Transition, 571106
[/INST]

### Response:
II-edges:
II-1: 571030, Rephrase, 571026
...
II-8: 571108, Inference, 571104

LI-edges:
LI-1: 571024, Asserting, 571026
...
LI-15: 571106, Asserting, 571108

LLII-edges:
LLII-1: 571046, Restating, II-1
...
LLII-8: 571110, Arguing, II-8

LLI-edges:

Figure 2: Mistral Type 1 instruction with generated response.

structions, we observed that the model often leads to incomplete annotations. For example, for the Type 1 instructions, the model stops generating past the *S-nodes*. Similarly, for the Type 2 instructions, the model frequently does not generate all the distinct types of *YA-nodes*. Hence, we implement an iterative decoding approach by re-prompting the model with partially generated annotations until it predicts all relationship types. For the Type 1 instructions, we first pass the IAT definitions and task details (left side of Figure 2) as prompt and generate the *S-nodes* (II-nodes in Figure 2). We iteratively append the generated output (highlighted in purple) to the prompt and re-prompt the model to generate the *YA-nodes* (LI-edges in Figure 2). We follow this approach until the model identifies all types of relationships (LLII and LLI-edges in Figure 2). We follow a similar approach for Type 2 instructions by re-prompting the model incrementally with the highlighted sections in Figure 3.

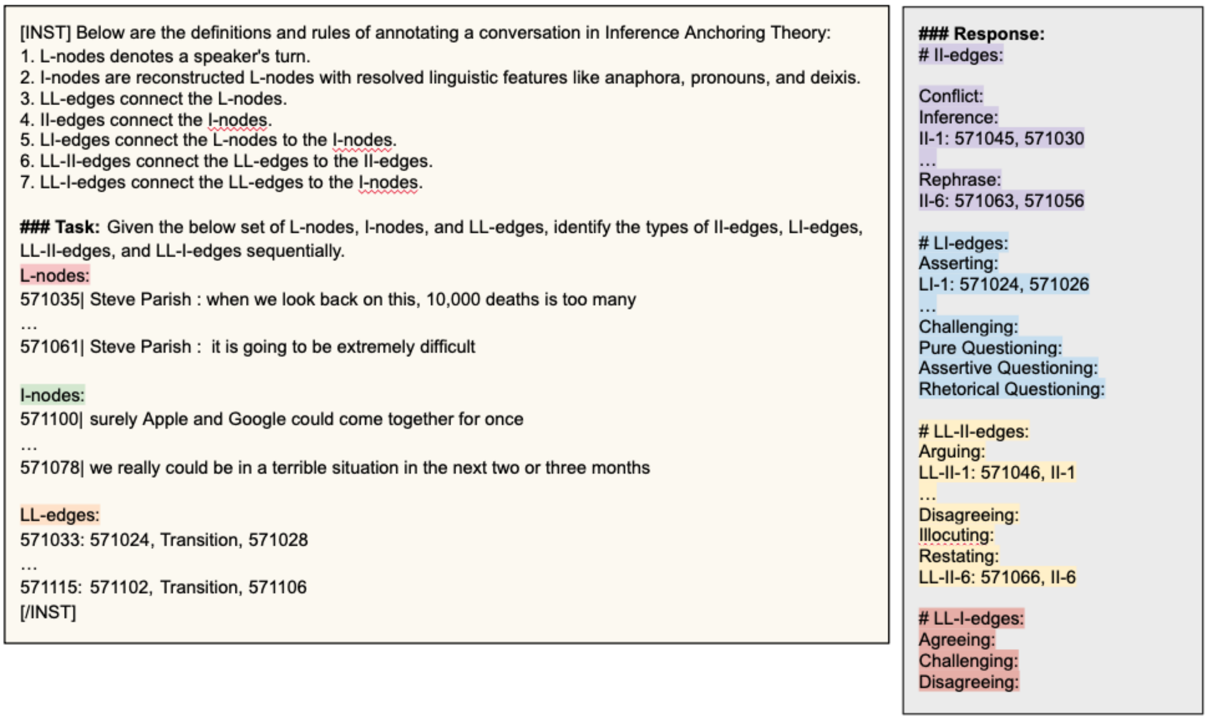
2.4 Results and Observations

Although our iterative decoding approach for the generative models facilitates better annotations, they are computationally expensive. Compared to regular decoding, they are approximately 4x more expensive for the Type 1 instructions and approximately 4-15x costlier for the Type 2 instructions. Hence, we internally compare the three model variants on a random sample of 10 examples from the test set and share the results in Table 1. We use the original task evaluation script, which computes **Pr**ecision, **Re**call, and **F1** scores at *Focused* and *General* levels. *Focused* evaluates the performance of the systems by looking at the related propositions/locutions in the evaluation files only, excluding all the non-related cases. *General* looks at the whole map, including the non-related class. High performance in *General* but low in *Focused* represents over-reliability on the non-related nodes, and vice-versa for *Focused*.

		General			Focused		
Id	Model	Pr	Re	F1	Pr	Re	F1
1	Biaff	68.7	68.6	68.6	60.0	36.1	41.0
2	LLM (Type-1)	**82.4**	85.5	**83.8**	59.0	55.8	57.3
3	LLM (Type-2)	81.7	73.4	75.5	49.5	37.0	40.4
4	Biaff + LLM (Type-1)	75.2	**89.3**	80.1	**68.9**	**67.1**	**67.9**
5	Biaff + LLM (Type-2)	69.4	80.1	73.1	61.2	54.7	56.1
6	LLM (Type-1 + Type-2)	77.6	80.4	78.7	60.5	56.6	58.1
7	Biaff + LLM (Type-1 + Type-2)	68.8	83.2	73.6	67.1	65.0	65.4

Table 1: Model performance on internal test set.

We also ensemble the three model variants and report results in Table 1 (lower half). We observe the following: (i) For all model variants, the F1 scores at *General* level are higher than *Focused*,



[INST] Below are the definitions and rules of annotating a conversation in Inference Anchoring Theory:
1. L-nodes denotes a speaker's turn.
2. I-nodes are reconstructed L-nodes with resolved linguistic features like anaphora, pronouns, and deixis.
3. LL-edges connect the L-nodes.
4. II-edges connect the I-nodes.
5. LI-edges connect the L-nodes to the I-nodes.
6. LL-II-edges connect the LL-edges to the II-edges.
7. LL-I-edges connect the LL-edges to the I-nodes.

### Task: Given the below set of L-nodes, I-nodes, and LL-edges, identify the types of II-edges, LI-edges, LL-II-edges, and LL-I-edges sequentially.
L-nodes:
571035| Steve Parish : when we look back on this, 10,000 deaths is too many
...
571061| Steve Parish : it is going to be extremely difficult

I-nodes:
571100| surely Apple and Google could come together for once
...
571078| we really could be in a terrible situation in the next two or three months

LL-edges:
571033: 571024, Transition, 571028
...
571115: 571102, Transition, 571106
[/INST]

### Response:
II-edges:

Conflict:
Inference:
II-1: 571045, 571030
...
Rephrase:
II-6: 571063, 571056

LI-edges:
Asserting:
LI-1: 571024, 571026
...
Challenging:
Pure Questioning:
Assertive Questioning:
Rhetorical Questioning:

LL-II-edges:
Arguing:
LL-II-1: 571046, II-1
...
Disagreeing:
Illocuting:
Restating:
LL-II-6: 571066, II-6

LL-I-edges:
Agreeing:
Challenging:
Disagreeing:

Figure 3: Mistral Type 2 instruction with generated response.

denoting that the models does not relate all propositions/locutions. This is expected as the models are trained on reduced relations (discussed in Section 2.3). (ii) Comparing the F1 score, the LLMs outperforms the biaffine classifier at both levels. Furthermore, the LLM trained on the Type 1 instruction outperforms the Type 2 instruction-trained model at both levels of evaluation. (iii) Ensembling the biaffine and Type 1 instruction-based LLM yields the best Precision, Recall, and F1 scores when evaluated at the *Focused* level. An ensemble comprising all three models performs the second best.

		General			Focused		
Task	Model	Pr	Re	F1	Pr	Re	F1
Task A	Majority-BL	28.8	30.3	29.5	0.0	0.0	0.0
	RoBERTa-BL	28.6	**34.7**	26.5	**37.1**	**18.4**	**22.8**
	Ours	**30.8**	31.5	**30.8**	19.0	4.2	6.7
Task B	Majority-BL	34.7	35.9	35.3	0.0	0.0	0.0
	RoBERTa-BL	39.1	**62.1**	45.8	**73.1**	**72.6**	**72.1**
	Ours	**51.4**	57.1	**53.3**	43.8	26.1	30.4
Global	Majority-BL	31.8	33.1	32.4	0.0	0.0	0.0
	RoBERTa-BL	33.9	**48.4**	36.1	**55.1**	**45.5**	**47.5**
	Ours	**41.1**	44.3	**42.0**	31.4	15.2	18.5

Table 2: Model performance on official test set.

Following our internal results, we use the ensembled biaffine and Type 1 instruction-based LLM-*Biaff + LLM (Type 1)* to parse the official test set samples and share our official test set results in Table 2. The table compares our implementation against majority-based and Roberta-based baselines for tasks A and B. It also shares global-level evaluations by looking at the complete argument maps. We observe the following: (i) Across all tasks, our implementation attains the best F1 score at the *General* level, whereas the Roberta baseline attains the best score at the *Focused* level. This observation is warranted as the *Focused* evaluates only the types of relationships prevalent in the dialogue and ignores all other classes. Our iterative decoding approach explicitly prompts the LLM to generate annotations for all relationship types, which can lead to spurious predictions by promoting recall. (ii) Similar to the baseline, our model performs Task B better than Task A.

3 Conclusion

Here, we computationally implement the theoretical IAT framework using classification and LLM-based models. We question the viability of leveraging LLMs, which are generative models, for such a nuanced task and compare them against simpler classifiers (non-generative) such as biaffine parsers. Our results indicate that posing the graph construction problem as a generative task and finetuning LLMs outperforms biaffine classifiers. Furthermore, ensembling the generative and classification-based approaches yields the best results.

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.

Katarzyna Budzynska, Mathilde Janier, Juyeon Kang, Chris Reed, Patrick Saint-Dizier, Manfred Stede, and Olena Yaskorska. 2014. Towards argument mining from dialogue. In *Computational Models of Argument*, pages 185–196. IOS Press.

Katarzyna Budzynska, Mathilde Janier, Chris Reed, and Patrick Saint-Dizier. 2016. Theoretical foundations for illocutionary structure parsing. *Argument & Computation*, 7(1):91–108.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *Preprint*, arXiv:1611.01734.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. 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.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Annette Hautli-Janisz, Zlata Kikteva, Wassiliki Siskou, Kamila Gorska, Ray Becker, and Chris Reed. 2022. Qt30: A corpus of argument and conflict in broadcast debate. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 3291–3300. European Language Resources Association (ELRA).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mathilde Janier, John Lawrence, and Chris Reed. 2014. Ova+: An argument analysis interface. *Frontiers in Artificial Intelligence and Applications*, 266:463–464.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

LENZ Mirko, Premtim Sahitaj, Sean Kallenberg, Christopher Coors, Lorik Dumani, Ralf Schenkel, and Ralph Bergmann. 2020. Towards an argument mining pipeline transforming texts to argument graphs. *Computational Models of Argument: Proceedings of COMMA 2020*, 326:263.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 2020. Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266, Online. Association for Computational Linguistics.

Huy V. Nguyen and Diane J. Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. In *AAAI*.

Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Ramon Ruiz-Dolz, John Lawrence, Ella Schad, and Chris Reed. 2024. Overview of DialAM-2024: Argument Mining in Natural Language Dialogues. In *Proceedings of the 11th Workshop on Argument Mining*, Thailand. Association for Computational Linguistics.

Sougata Saha, Souvik Das, and Rohini Srihari. 2022a. EDU-AP: Elementary discourse unit based argument parser. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 183–192, Edinburgh, UK. Association for Computational Linguistics.

Sougata Saha, Souvik Das, and Rohini K. Srihari. 2022b. Dialo-AP: A dependency parsing based argument parser for dialogues. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 887–901, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014a. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014b. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56.

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

Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 669–678, Online. Association for Computational Linguistics.