DFKI-MLST at DialAM-2024 Shared Task: System Description

Arne Binder[*] **Tatiana Anikina**[*] **Leonhard Hennig** **Simon Ostermann**
German Research Center for Artificial Intelligence (DFKI)
{*arne.binder, tatiana.anikina, leonhard.hennig, simon.ostermann*}*@dfki.de*

Abstract

This paper presents the `dfki-mlst` submission for the DialAM shared task (Ruiz-Dolz et al., 2024) on identification of argumentative and illocutionary relations in dialogue. Our model achieves the best results in the global setting: 48.25 F1 at the focused level when looking only at the related arguments/locutions and 67.05 F1 at the general level when evaluating the complete argument maps. We describe our implementation of the data pre-processing pipeline, relation encoding and classification, evaluating 11 different base models and performing experiments with, e.g., node text combination and data augmentation. Our source code is publicly available.[1]

1 Introduction

DialAM 2024 (Ruiz-Dolz et al., 2024) is the first shared task in dialogue argument mining. It uses the Inference Anchoring Theory (IAT) framework (Budzynska et al., 2014) as data schema. IAT describes argument structure as graphs of propositions that are derived from the argumentative discourse units (ADUs; the basic units of argumentative analysis). The shared task focuses on the detection and classification of the relations that (1) argumentatively link these propositions with each other (ARI) and that (2) anchor them in the corresponding ADUs (ILO).

The DialAM dataset is based on the QT30 corpus (Hautli-Janisz et al., 2022), which is a collection of 30 episodes of the show Question Time by the BBC. The dataset includes transcriptions of dialogues between a moderator and several panelists and audience members annotated according to the IAT. Figure 1 visualizes the structure of the data. In simplified terms, IAT models argumentation information as a bipartite graph.[2] One side consists of the ADUs as they occur in the original text, called **location (`L`)** nodes, and **transition (`TA`)** nodes that link them in the direction of the dialog flow. The other side consists of **information (`I`)** nodes which encode the propositions derived from the locutions and the **argumentative relation (`S`)** nodes (e.g., `Inference`, `Conflict`, or `Rephrase`) that connect them in the direction of argumentative reference. Finally, `I` and `S` nodes are anchored by **illocutionary relation (`YA`)** nodes in `L` and `TA` nodes, respectively, i.e. they encode from which `L` and `TA` nodes they are derived. The relation nodes connect to their arguments via two distinct roles: `incoming` (edges point towards the relation node) and `outgoing` (edges point away from it). The shared task data are organized in nodesets where each nodeset is a collection of annotated nodes and edges in Argument Interchange Format (Rahwan and Reed, 2009) extracted from an episode.

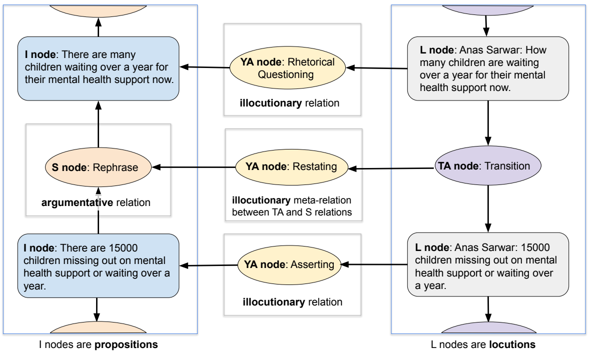

I node: There are many children waiting over a year for their mental health support now.
YA node: Rhetorical Questioning
illocutionary relation
L node: Anas Sarwar: How many children are waiting over a year for their mental health support now.
S node: Rephrase
argumentative relation
YA node: Restating
illocutionary meta-relation between TA and S relations
TA node: Transition
I node: There are 15000 children missing out on mental health support or waiting over a year.
YA node: Asserting
illocutionary relation
L node: Anas Sarwar: 15000 children missing out on mental health support or waiting over a year.
I nodes are **propositions**
L nodes are **locutions**

Figure 1: Extract of an example DialAM data point. Argumentative structure (left side; `I` and `S` nodes) is anchored in the dialogue structure (right side; `L` and `TA`) by illocutionary relations (middle; `YA` nodes) that are based on speech acts. The DialAM 2024 shared task requires identification as well as classification of (1) `S` node relations (ARI) and (2) `YA` node relations (ILO).

DialAM poses some unique challenges because it requires three different types of relations to be extracted (see Figure 1): argumentative relations

[*] Equal contribution.

[1] github.com/ArneBinder/dialam-2024-shared-task

[2] There are several exclusions to this, e.g., reported speech.

between propositions (**S nodes; subtask 1**), illocutionary relations modeling speech acts (**YA:L-to-I nodes; subtask 2.1**) and, relations between argumentative relations and dialogue turn transitions (**YA:TA-to-S nodes; subtask 2.2**). Note that all relations have at least one incoming and outgoing edge, but argumentative relations (S nodes) such as Inference may have more than one incoming edge. Thus, subtask 1 is an instance of n-ary relation extraction. Furthermore, YA:TA-to-S relations link TA and S nodes which are both relation nodes, so this is a meta-relation. Both aspects circumvent usual relation extraction approaches that assume binary relations connecting spans over text. There are 25 relation labels in total with a very imbalanced distribution (see Appendix A and E.2).

Previous approaches to dialogue argument mining, such as Ruiz-Dolz et al. (2021), have shown that Transformer-based models work well on the argument relation identification task, with RoBERTa (Liu et al., 2019) significantly outperforming BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), DistilBERT (Sanh et al., 2019) and ALBERT (Lan et al., 2020). They have found that in many cases misclassified relations were due to the lack of context or multiple valid interpretations of a relation. However, Ruiz-Dolz et al. (2021) address a simpler task compared to the DialAM setup because they classify only propositional relations while DialAM involves n-ary relations between different types of nodes (propositions, locutions and meta-relations).

Our contributions are as follows: (1) we introduce a unified approach towards dialogue argument mining based on n-ary relation classification and train a single model that can handle all three types of relations to get the most out of the data, (2) our dfki-mlst submission achieves the best scores in the global setting of the shared task, and (3) we conduct a comparative analysis of different types of base models, explore data augmentation, weighted loss and node text combination.

2 System Architecture

We handle all three subtasks by framing them as n-ary relation classification. Let $r^A = \{(l,a)|l \in L, a \in A\}$ be a n-ary relation with L the set of possible argument roles and A the set of possible relation arguments such as the set $S_t = \{(i,j,l)\}$ of labeled spans over a text t with i and j start and end indices with respect to t and l the label. We define n-ary relation classification as assigning a class $c \in C$ to r where C is the set of possible classes. In the following we explain how we construct relations r^{S_t}, i.e. sets of argument-role – text-span pairs over a single text, and the relation classes c from the individual relation nodes. In detail, we describe how we normalize the data (2.1), encode each task data as n-ary relations (2.2), and, finally, how we implement the relation classification (2.3).

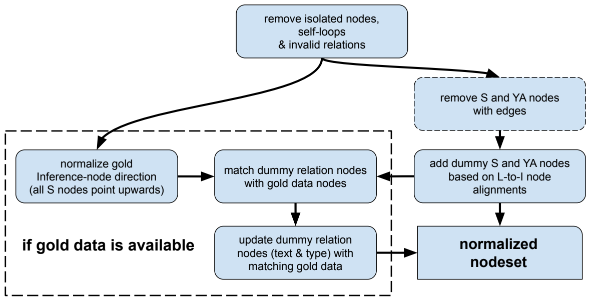

remove isolated nodes, self-loops & invalid relations
remove S and YA nodes with edges
normalize gold Inference-node direction (all S nodes point upwards)
match dummy relation nodes with gold data nodes
add dummy S and YA nodes based on L-to-I node alignments
if gold data is available
update dummy relation nodes (text & type) with matching gold data
normalized nodeset

Figure 2: Nodeset normalization. Dashed boxes indicate steps that are only relevant for model training.

2.1 Nodeset Normalization

To encode the data as relations, we use some heuristics to construct a full nodeset from the provided nodes (L, I, and TA). This will include already all edges, but we assign a dummy NONE label to all relation nodes that we add. We achieve this by exploiting the following observations.

First, each I node is usually anchored by exactly one L node. Since the I node text is derived from the corresponding L node, their text contents are very similar. We use this to find an alignment of L and I nodes by computing their textual similarity using longest common substring and calculate a pairwaise assignment. This alignment allows us to construct the YA:L-to-I nodes.

Second, the incoming and outgoing edges of the S nodes usually mirror their counterparts at the anchoring TA node, but in reversed direction (i.e. outgoing edges of the TA nodes mirror incoming edges of the S nodes and the other way around). This allows us to construct S nodes by reversing the TA arguments and mirroring them to I nodes by following the L-I-alignment. However, there is one prominent edge case. The S nodes with label Inference may point in the opposite direction. We can normalize that by swapping the incoming and outgoing edges of all such Inference nodes in the gold data and assigning a special node label Inference-rev(ersed) to maintain the original semantics. We can determine if an Inference node needs to be swapped by looking at the direction of the anchoring TA node.[3]

Finally, we assign gold labels to all constructed relation nodes for which we find matching gold nodes by considering only the arguments and their roles.

We found several issues with the data (e.g. isolated nodes, self-loops, relations with invalid combinations of arguments) that we fix before applying the normalization steps. Also, there are valid cases that contradict the above assumptions (I nodes of reported speech may have no directly anchoring L node; I and S nodes with multiple anchors; etc.), but since they are very rare we discard such nodes. Figure 2 visualizes the full normalization pipeline.

2.2 Encoding as Relations

To encode all task relevant relation nodes as relations r^{S_t}, we first convert them to n-ary relations over locutions (L nodes) r^L and then construct a common base text t from all locutions.

We encode the YA:L-to-I nodes (subtask 2.1) as unary relation classification where we use the anchoring L node as single argument with its role (outgoing). For YA:TA-to-S nodes (subtask 2.2), we use the arguments of the related TA relation with their respective roles. We encode the S nodes (subtask 1) by using the L nodes that anchor their arguments, but with the S node roles. In all three cases, we use the relation node label as label, but prefix it as well as the argument roles (incoming and outgoing) with the respective task identifier (S, YA:L-to-I, or YA:TA-to-S).

To get a contiguous base text t, we concatenate all locutions in the direction of the dialogue flow.[4] Note that the L nodes do not form a sequence, but a directed graph. Since there are no reliable time stamps, we linearize this graph in such a way that the ordering of the nodes is preserved.[5] We use the start and end offsets of the L node texts in t to construct r^{S_t} from r^L.

Using distinct roles and a common base text allows us to use a single model to solve all subtasks.

2.3 Classification Model

We use a deep learning based text classification model consisting of a contextual text encoder and a one layer classification head implemented within the PyTorch-IE framework (Binder et al., 2024). First, role specific begin- and end-marker tokens are inserted into the base text for all arguments of the relation to classify. Then, the modified text is classified by the model.

We use the cross entropy loss and the Adam optimizer to train it. The source code is publicly available.[1]

3 Experiments and Results

With the relation classification approach described in Section 2 we train our model on the DialAM data. We split the original training set into training (1259 nodesets) and validation (140 nodesets) partitions and repeat the training procedure three times with different seeds. The best model is selected based on the validation set performance. Our dfki-mlst submission uses DeBERTa-v3[6] (He et al., 2021) as text encoder trained with a learning rate 1e-4 and a window size of 512 tokens[7] for 20 epochs on a single GPU NVIDIA H100 80GB HBM3. We evaluate our model with the official script that outputs precision, recall and F1 scores for the ARI and ILO tasks, and the GLOBAL metrics represent the combined scores. All scores are calculated at two levels: focused (only related arguments/locutions) and general (complete argument maps).

Table 1 shows the comparison of F1 scores on the gold test data between the official RoBERTa baseline, our dfki-mlst submission and the best-performing competitor model in each setting. Our approach shows overall strong performance in the GLOBAL setting when complete argument maps are taken into account (+0.8% F1 in GLOBAL-Focused and +3.35% in GLOBAL-General). dfki-mlst also outperforms other models on the ARI-General task for propositional relations (+9.11% F1).

3.1 Error Analysis

Since the nodeset normalization plays a major role in our setup, we evaluate its impact based on our validation set with 140 nodesets. To make the normalized nodesets comparable with the original data,

[3] We can use the L-I-alignment to get all anchoring L nodes for the arguments of the S node at hand. Then, we can check if there is a TA node with these anchor nodes as arguments or with the swapped arguments.

[4] This means, that we completely ignore I node text.

[5] i.e. for all node pairs (x, y) where there is a path from x to y, x must occur before y in the linearized nodes.

[6] huggingface.co/microsoft/deberta-v3-large

[7] For each relation classification pass, the window is centered at the minimal span covering all its relation arguments.

o	ARI		ILO		GLOBAL	
	Focused	*General*	*Focused*	*General*	*Focused*	*General*
baseline	22.80	26.46	**72.09**	45.75	47.45	36.10
best-competitor	**35.89**	46.22	69.95	**81.17**	45.23	63.70
dfki-mlst (ours)	30.40	**55.33**	66.10	78.78	**48.25**	**67.05**

Table 1: F1 scores of the official baseline, best competitor model, and `dfki-mlst` (ours) per task.

we reverse all `Inference-rev` relations back as well as remove the `NONE` nodes. We observe a lower performance, i.e. higher impact, on ARI (78.61 focused F1, 93.04 general F1) when compared to ILO (83.05 focused F1, 95.61 general F1). See Table 2 in Appendix C.1 for the complete results.

We also evaluate `dfki-mlst` performance per label based on our validation set (see Figure 5 in Appendix C for statistics). Unsurprisingly, the most common `YA` node relation `Asserting` achieves the highest F1 score (99%) since this label is also well-represented in the training set (see Appendix A for label distribution). We also observe that `NONE` relation between different types of nodes can be classified reliably in most cases. We found that some classes are distinctive and easy to classify. E.g., `Pure Questioning` between `I` and `L` nodes with the support of 120, and 1.86% representation in the training data, has 81% F1. Other categories are more challenging and result in worse scores even when they have more training samples, e.g., `Default Inference` constitutes 3.85% of the training set with the support of 246 but the classifier achieves only 43% F1.

3.2 Base Model Comparison

We explore different LLMs as text encoders in our classification model and evaluate them on the released gold test data. The results show that although DeBERTa-v3 is the best-performing model on the validation set (+0.85% on GLOBAL-General and +0.5% on GLOBAL-Focused compared to the second best model DeBERTa-v1), it shows slightly worse performance than DeBERTa-v1 on the test data. Interestingly, BART (Lewis et al., 2019) demonstrates the best F1 scores in both ILO-Focused (72.28 F1) and ILO-General (83.68 F1) settings. However, its performance on the argument relation identification task is considerably worse (-7.98% F1 in Focused and -9.84 % F1 in General). Also, models such as Mistral (Jiang et al., 2023) and Llama (Touvron et al., 2023) do not achieve very good results when fine-tuned on the relation classification task (see Appendix B for the training details). We compare Mistral and Llama fine-tuning to the setting where we freeze the base model and fine-tune only the adapter weights with LoRA (Hu et al., 2022). In both cases LoRA outperforms the fine-tuned models but still underperforms DeBERTa. The results of the full analysis are shown in Tables 3 and 4 in Appendix D.

3.3 Experiments with Input Data Modification and Weighted Loss

Although our `dfki-mlst` submission uses only `L` node texts we experimented with combining both `L` and `I` node texts when encoding relations and this setup achieves the best scores in the GLOBAL setting and also improves our performance on the ILO task compared to the original submission. Further details can be found in Section E.3 in Appendix.

After nodeset cleaning and normalization we were left with only 1259 documents (compared to the original 1478). Hence, we decided to experiment with data augmentation to increase the amount of available data and train a more robust model. We modify `L` node texts using two different approaches: paraphrase-based data augmentation and token-level perturbations based on Easy Data Augmentation (EDA) (Wei and Zou, 2019). Combining EDA-augmented and original data improves F1 scores for ARI-Focused and ARI-General tasks but results in worse performance on the ILO task. More details can be found in Section E.1.

Given that the dataset has imbalanced distribution, we also experimented with weighted loss (see Section E.2) and found that with this approach we get some improvements on the ARI task but overall worse performance compared to vanilla DeBERTa.

4 Conclusion

This paper introduces the `dfki-mlst` submission that achieves the best scores in the global evaluation setting of the DialAM shared task. We describe our nodeset pre-processing pipeline and the system architecture. We also present the comparison of different base models (DeBERTa, BART, Mistral etc.) as well as our experiments with data augmentation, class distribution and node text combination. We observe that some models (e.g., DeBERTa) demonstrate better performance on the argument relation task while other models (e.g., BART) are better at detecting illocutionary relations.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback on our work. This work has been supported by the German Ministry of Education and Research (BMBF) as part of the project TRAILS.

References

Arne Binder, Leonhard Hennig, and Christoph Alt. 2024. Pytorch-ie: Fast and reproducible prototyping for information extraction. *Preprint*, arXiv:2406.00007.

Katarzyna Budzynska, Mathilde Janier, Chris Reed, Patrick Saint-Dizier, Manfred Stede, and Olena Yaskorska. 2014. A model for processing illocutionary structures and argumentation in debates. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 917–924. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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 Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 3291–3300. European Language Resources Association.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

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. *CoRR*, abs/1907.11692.

Vukosi Marivate and Tshephisho Sefara. 2019. Improving short text classification through global augmentation methods. *CoRR*, abs/1907.03752.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Iyad Rahwan and Chris Reed. 2009. The argument interchange format. In Guillermo Ricardo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 383–402. Springer.

Ramon Ruiz-Dolz, José Alemany, Stella Heras Barberá, and Ana García-Fornes. 2021. Transformer-based models for automatic identification of argument relations: A cross-domain evaluation. *IEEE Intell. Syst.*, 36(6):62–70.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

A Relation Label Distribution

Figure 3 and 4 show the overall and per relation node type label distribution in the train data.

B Training Details for Large Models

Since LLMs such as Mistral and LLama have a large number of parameters and fine-tuning all of them would require a lot of GPU memory, we freeze the first 30 layers and fine-tune only the last two layers together with the classification head (see Mistral-30 and Llama-30 in Tables 3 and 4).

C Error Analysis

C.1 Impact of Preprocessing

Experimental results regrading the impact of the nodeset normalization measured on the validation data can be found in Table 2.

C.2 Performance per Label

Figure 5 compares the amount of support with the per label performance.

D Model Comparison

Figure 3 and 4 show the focused as well as the general metric scores for all analysed models on the test data.

E Additional Experiments

E.1 Data Augmentation Experiments

Our experiments with data augmentation do not modify the original relations and nodeset structure, we change only the L node text by either paraphrasing it with a T5-based model trained on Chat-GPT paraphrases[8] or using an Easy Data Augmentation (EDA) (Wei and Zou, 2019) approach based on textaugment (Marivate and Sefara, 2019). In case of EDA we randomly choose whether to replace a token with a synonym from WordNet (Miller, 1995), delete it, add a new token, or swap one token with another. Ideally, such changes introduce surface perturbations without changing the original meaning, therefore annotations remain the same. E.g., for the original L node text *"Claire Fox: that will show how virtuous I am"* we have the following paraphrase-based augmentation: *"Claire Fox: My goodness will be demonstrated to others through this."* and the EDA-based augmentation: *"Claire Fox: appearance that will show how virtuous I am"*.

Tables 3 and 4 in Appendix D show the results for the augmented models in the lower section of each table. In case of DeBERTa-v1$_{+paraphr\_data}$ we fine-tune DeBERTa-v1 model on the paraphrased data and then continue fine-tuning on the original DialAM training set. DeBERTa-v3$_{+EDA\_sequential}$ follows the same strategy but instead of using paraphrased text it applies token-level perturbations (EDA). Note that we changed our base model from DeBERTa-v1 to DeBERTa-v3 in the latest experiments since it gave us the best scores on the validation set. Finally, DeBERTa-v3$_{+EDA\_combined}$ simply extends the dataset by combining both original and augmented documents. The results show that among these three strategies combining the data brings some improvement compared to vanilla DeBERTa-v3 on ARI-Focused (+3.76% F1), GLOBAL-Focused (+1.23% F1), ARI-General (+1.53% F1) and GLOBAL-General (+0.36% F1) tasks but leads to worse scores on ILO-Focused (-1.3% F1) and ILO-General (-0.83% F1).

E.2 Experiments with Class Distribution

As shown in Appendix A (Figures 3, 4a, 4b, 4c), DialAM training data has an imbalanced class distribution with 12 labels representing less than 1% of all the data. Especially YA relations connecting locutions with propositions (see Figure 4c) have very imbalanced distribution. E.g., Asserting appears in more than 90% of S node annotations, while labels such as Restating, Arguing and Agreeing all together make up less than 1%, which poses a challenge for the classifier. Therefore, we test whether using a weighted loss adjusted with regards to label distribution or restricting classification only to more frequent classes (with at least 10 samples per label) can help mitigate this issue. In the experiments with weighted loss we (1) collect statistics from the training set on label distribution and (2) compute each class weight as follows: $w_c = \frac{|D|}{|D_c| \cdot |C|}$ with D the set of all samples (relation-class-pairs), $D_c = \{(r, c) \in D\}$ and C the set of all labels, and then (3) restrict the range of weight values by using 1 as the lower and 20 as the upper bound to avoid over-penalizing classifier on the truly rare classes.

The evaluation results in Tables 3 and 4 demonstrate that training with weighted loss improves the

[8] huggingface.co/humarin/chatgpt_paraphraser_on_T5_base

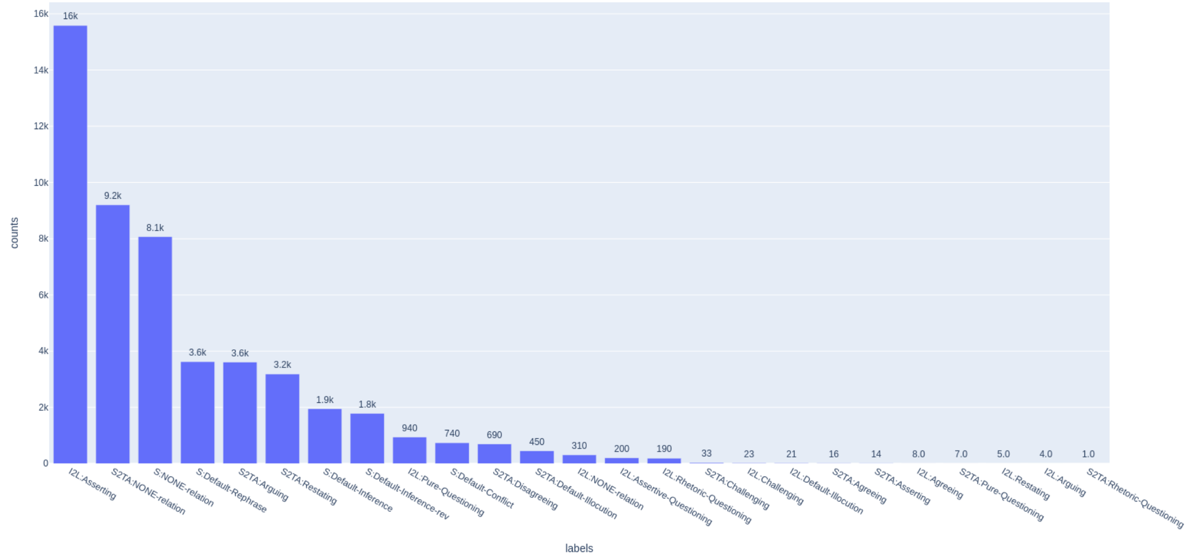
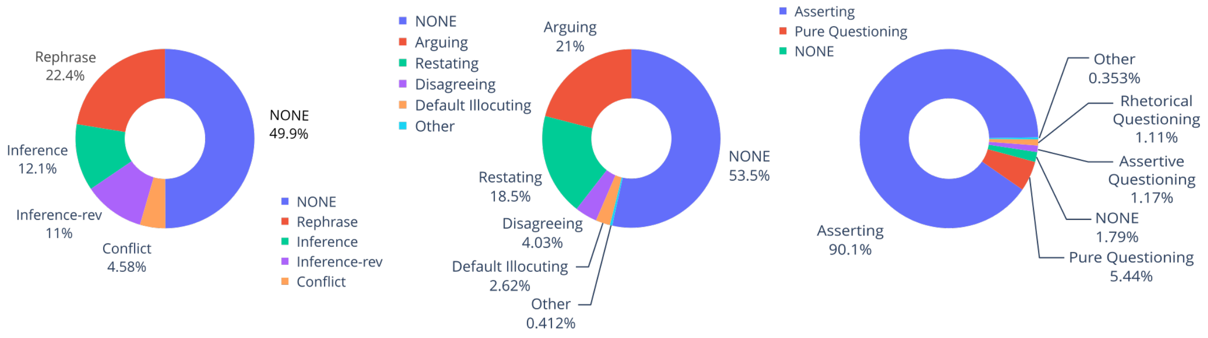


Figure 3: Overall label distribution in the DialAM training set.



(a) S relations between I nodes. (b) YA relations between TA and S nodes. (c) YA relations between L and I nodes.

Figure 4: Label distribution for different types of relations: S nodes for argumentative relations and YA nodes for illocutionary ones.

	Model	ARI			ILO			GLOBAL		
		Prec	*Rec*	*F*1	*Prec*	*Rec*	*F*1	*Prec*	*Rec*	*F*1
Focused	preprocessing only	82.85	76.60	78.61	84.17	82.27	83.05	83.51	79.44	80.83
	full pipeline	49.02	31.74	36.33	71.75	69.82	70.35	60.39	50.78	53.34
	full pipeline, normalized	59.17	41.44	46.22	85.24	84.87	84.71	72.31	63.93	65.99
General	preprocessing only	97.12	91.18	93.04	96.71	94.86	95.61	96.92	93.02	94.33
	full pipeline	66.43	58.91	60.06	86.23	84.71	85.11	76.33	71.81	72.59
	full pipeline, normalized	68.40	64.61	64.55	89.16	89.30	89.02	78.76	77.20	76.95

Table 2: Impact of nodeset normalization on the performance, evaluated on the validation data. The values for `full pipeline` are the scores of our model (dfki-mlst). `preprocessing only` values are computed by first normalizing the data as described in section 2.1, then reverting `Inference-rev` relations back as well as removing `NONE` relation nodes to make the normalized nodesets comparable with the original data and, finally, calculating the metrics with the official evaluation script. Values for `full pipeline, normalized` are the ones of `full pipeline` divided by `preprocessing only`.

Model	ARI-Focused			ILO-Focused			GLOBAL-Focused		
	$Prec$	Rec	$F1$	$Prec$	Rec	$F1$	$Prec$	Rec	$F1$
baseline$_{RoBERTa}$	37.10	18.42	22.80	73.10	72.55	**72.09**	55.10	45.49	47.45
best-competitor	46.26	32.43	**35.89**	71.18	69.23	69.95	50.68	43.41	45.23
dfki-mlst$_{DeBERTa\text{-}v3}$	43.87	24.82	30.40	69.12	66.25	66.10	56.50	45.53	**48.25**
DeBERTa-v1	50.98	27.98	**33.82**	66.04	64.32	64.63	58.51	46.15	**49.22**
RoBERTa	48.11	20.45	26.62	63.64	60.66	61.24	55.88	40.55	43.93
RemBERT	41.02	18.35	24.20	62.33	59.49	60.20	51.67	38.92	42.20
ELECTRA	37.46	14.65	20.25	68.76	67.54	67.37	53.11	41.10	43.81
BART	34.09	18.14	22.41	73.50	72.12	**72.28**	53.80	45.13	47.34
XLNet	36.75	19.90	24.04	60.63	58.89	58.22	48.69	39.39	41.13
Mistral-30	33.40	16.50	19.66	67.91	69.08	68.02	50.66	42.79	43.84
Llama-30	21.75	13.25	14.28	60.68	60.10	59.61	41.22	36.67	36.95
Mistral-LoRA	33.62	18.90	23.08	68.36	64.89	65.55	50.99	41.89	44.31
Llama-LoRA	39.07	16.56	22.08	64.68	62.40	63.16	51.88	39.48	42.62
DeBERTa-v1$_{+l\_and\_i\_node\_text}$	44.32	23.39	29.24	75.17	73.51	**74.10**	59.75	48.45	**51.67**
DeBERTa-v1$_{+freq\_classes}$	48.04	24.44	30.37	68.09	64.24	64.81	58.06	44.34	47.59
DeBERTa-v1$_{+weighted\_loss}$	47.35	28.98	**34.22**	59.48	59.75	58.85	53.41	44.37	46.53
DeBERTa-v1$_{+paraphr\_data}$	43.99	22.03	27.69	68.35	66.01	66.31	56.17	44.02	47.00
DeBERTa-v3$_{+EDA\_sequential}$	48.08	25.26	30.66	65.72	62.21	62.71	56.90	43.73	46.69
DeBERTa-v3$_{+EDA\_combined}$	47.73	29.11	34.16	65.92	64.53	64.80	56.83	46.82	49.48

Table 3: Focused scores represent the performance on the existing relations in the gold standard maps (excluding non related propositions). The scores were computed with the official evaluation script using the gold test data.

Model	ARI-General			ILO-General			GLOBAL-General		
	$Prec$	Rec	$F1$	$Prec$	Rec	$F1$	$Prec$	Rec	$F1$
baseline$_{RoBERTa}$	28.59	34.69	26.46	39.11	62.07	45.75	33.85	48.38	36.10
best-competitor	49.21	46.32	46.22	81.99	80.79	**81.17**	65.60	63.55	63.70
dfki-mlst$_{DeBERTa\text{-}v3}$	61.96	53.30	**55.33**	81.08	79.25	78.78	71.52	66.28	**67.05**
DeBERTa-v1	64.05	57.14	**57.93**	79.04	78.19	78.12	71.55	67.66	**68.03**
RoBERTa	64.86	49.55	52.73	76.83	75.05	75.17	70.84	62.30	63.95
RemBERT	54.79	46.00	47.56	75.92	74.28	74.49	65.36	60.14	61.02
ELECTRA	46.18	39.37	41.41	81.23	81.07	80.53	63.70	60.22	60.97
BART	49.78	44.3	45.49	84.22	83.97	**83.68**	67.00	64.13	64.59
XLNet	55.51	48.28	48.80	73.88	72.93	72.05	64.69	60.61	60.42
Mistral-30	47.20	43.14	42.33	82.15	83.91	82.62	64.68	63.52	62.48
Llama-30	39.59	38.38	37.10	74.93	75.01	74.22	57.26	56.70	55.66
Mistral-LoRA	54.41	47.06	49.03	82.07	79.78	79.99	68.24	63.42	64.51
Llama-LoRA	51.72	42.95	44.89	77.03	75.55	75.99	64.38	59.25	60.44
DeBERTa-v1$_{+l\_and\_i\_node\_text}$	57.52	50.39	52.33	86.57	85.17	**85.65**	72.05	67.78	**68.99**
DeBERTa-v1$_{+freq\_classes}$	65.05	52.71	55.66	80.38	77.62	77.80	72.72	65.17	66.73
DeBERTa-v1$_{+weighted\_loss}$	63.81	55.97	**58.20**	73.81	74.73	73.65	68.81	65.35	65.93
DeBERTa-v1$_{+paraphr\_data}$	58.42	49.66	51.36	80.44	79.30	79.16	69.43	64.48	65.26
DeBERTa-v3$_{+EDA\_sequential}$	62.53	52.97	54.88	78.31	75.79	75.92	70.42	64.38	65.40
DeBERTa-v3$_{+EDA\_combined}$	60.21	56.75	56.86	78.50	78.12	77.95	69.36	67.44	67.41

Table 4: General scores consider complete argument maps including non related nodes. The scores were computed with the official evaluation script using the gold test data.

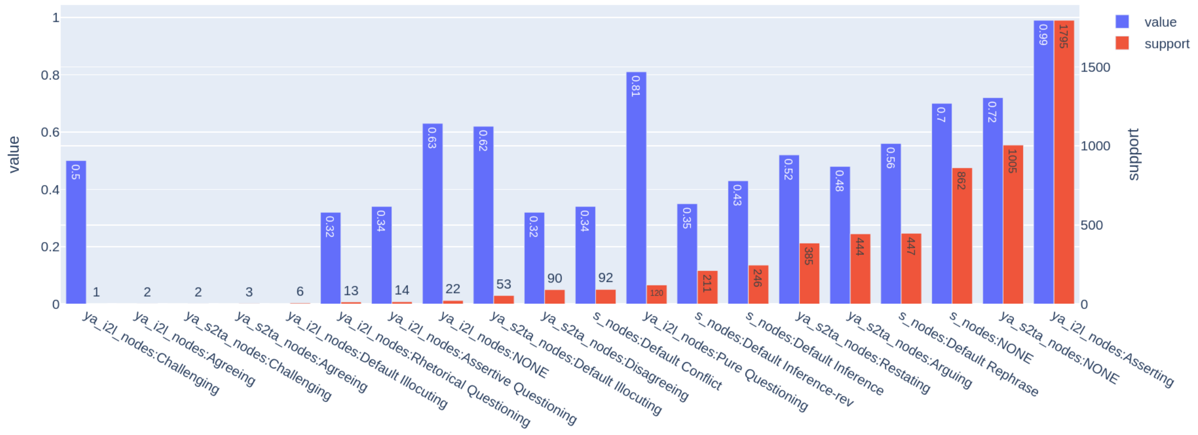


Figure 5: Performance of dfki-mlst with DeBERTa-v3 on the fixed validation set (140 documents). Blue bars indicate F1 scores while red bars correspond to the support set (how many items per class are available).

scores on the ARI task (for both Focused and General), however, this approach leads to a drop in performance for illocutionary relation identification. Furthermore, using only samples from more frequent classes results in overall worse performance which shows that having even few examples of rare labels is beneficial for the classifier.

E.3 Experiments with Combined Node Text

Since I and L nodes represent arguments and dialogue turns, they have slightly different texts. I node text often includes more details "summarizing" the content as an argument, while L node text represents an unedited dialogue turn. dfki-mlst uses only the L node texts to encode the relations because this type of text is more similar to the data used for pre-training of the base model (DeBERTa) but we also test the setting that combines both texts of the aligned L and I nodes separated by the *"Argument:"* token that indicates the transition. As shown in Tables 3 and 4, this approach results in the best performance on illocution identification and achieves overall best scores in GLOBAL, outperforming the version that uses only L node texts by 3.4% F1 in Focused and 1.9% F1 in General for DeBERTa-v3 and showing a similar trend for DeBERTa-v1. However, it under-performs on the argument relation identification task compared to the dfki-mlst submission.