Enhancing AMR Parsing with Group Relative Policy Optimization

Botond Barta[1], Endre Hamerlik[1], Milán Nyist[2], Masato Ito[2], Judit Ács[1]

[1]HUN-REN SZTAKI, [2]Eötvös Loránd University

Correspondence: botondbarta@sztaki.hu

Abstract

We investigate the capabilities of the openly available Llama 3.2 1B language model for Abstract Meaning Representation (AMR) parsing through supervised fine-tuning, further enhanced by reinforcement learning via Group Relative Policy Optimization (GRPO). Existing supervised methods for AMR parsing face limitations due to static loss functions and challenges in capturing complex semantic phenomena. To address this, our GRPO-based approach explicitly optimizes fine-grained semantic rewards, including Smatch scores, frame-argument correctness, and structural validity of logical operations. Experimental results show that supervised fine-tuning alone establishes Llama as a capable English AMR parser, and subsequent GRPO fine-tuning further improves its performance. Our final model achieves higher Smatch scores, consistently respects critical low-level semantic constraints, and outperforms existing parsers on high-level semantic evaluation metrics across diverse linguistic phenomena.

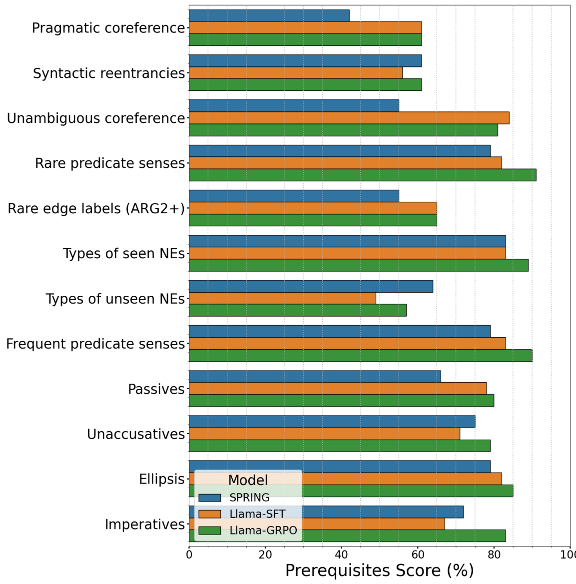

Figure 1: Comparison of AMR parsing models (SPRING, Llama-SFT, and our Llama-GRPO) across various linguistic phenomena measured by the GrAPES prerequisites metric. Higher scores indicate that the parser more consistently generates the necessary semantic structures to capture specific phenomena. Our reinforcement learning-based approach shows consistent improvement over the baselines.

1 Introduction

Abstract Meaning Representation has become essential in various natural language processing tasks, such as machine translation (Song et al., 2019; Damonte et al., 2019; Urešová et al., 2014), question answering (Kapanipathi et al., 2021), dialogue understanding (Bai et al., 2022a), summarization (Liao et al., 2018; Ribeiro et al., 2022; Dohare et al., 2017), and fact-checking (Ribeiro et al., 2022; Kachwala et al., 2024; Ousidhoum et al., 2022). Despite its widespread adoption, AMR parsing remains challenging. Groschwitz et al. (2023) recently demonstrated that parsing accuracy has stagnated, highlighting persistent difficulties in capturing complex semantic phenomena, even with advanced models.

While large language models (LLMs) such as the Llama models (Touvron et al., 2023) have demonstrated impressive performance across various language generation tasks, their capability for structured semantic parsing—particularly AMR parsing—remains unverified. Moreover, it remains unclear whether advanced reinforcement learning (RL) techniques like Group Relative Policy Optimization (GRPO), introduced by Shao et al. (2024), can effectively enhance the performance of LLMs on such structured prediction tasks by directly optimizing for desired graph properties.

In this paper, we first examine the baseline capabilities of the openly available Llama 3.2 1B model by supervised fine-tuning (SFT) on the AMR 3.0 dataset (Banarescu et al., 2013). We refer to this model as **Llama-SFT**. We then further fine-tune

Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025), pages 99–105
August 1, 2025

this model using GRPO, incorporating fine-grained reward signals explicitly designed to encourage adherence to critical low-level AMR properties, such as frame-argument correctness and structural validity of logical operations (AND-OR node correctness), alongside Smatch and graph parsability. We call this enhanced model **Llama-GRPO**. We systematically evaluate our models against publicly available AMR parsing model, SPRING (Bevilacqua et al., 2021), using standard metrics (Smatch) and the detailed GrAPES evaluation suite (Groschwitz et al., 2023).

Our results show that the Llama 3.2 1B model, after supervised fine-tuning (Llama-SFT), achieves AMR parsing performance close to open-source AMR parser models. Critically, when further enhanced through GRPO-based reinforcement learning, our model:

- Achieves higher overall AMR parsing accuracy, as measured by Smatch scores,
- Effectively respects the low-level semantic constraints incorporated into the GRPO reward function,
- Outperforms existing AMR parsers on high-level semantic evaluations, as demonstrated by the comprehensive GrAPES metrics (Figure 1), suggesting improved generalization across diverse linguistic phenomena.

2 Related Work

Early work in AMR parsing often relied on transition-based systems (Wang et al., 2015, 2016) or graph-based approaches (Flanigan et al., 2014), frequently using specialized features and constrained decoding. The advent of neural sequence-to-sequence models marked a significant shift. Many modern parsers treat AMR parsing as a translation task from text to a linearized representation of the AMR graph (Konstas et al., 2017).

Transformer-based architectures (Vaswani et al., 2017) quickly became dominant. Models like SPRING (Bevilacqua et al., 2021), based on BART (Lewis et al., 2020), demonstrated strong performance by leveraging pre-training and specialized techniques like graph linearization. SPRING employs bidirectional pre-training and graph-based regularization during fine-tuning on linearized AMR graphs. Other parsers, such as those based on T5 or BART (Raffel et al., 2020; Jascob, 2024; Lee et al., 2023), have also achieved high results through large-scale pre-training and task-specific adaptations.

Despite these advances, as highlighted by Groschwitz et al. (2023), performance has plateaued, suggesting limitations in current supervised approaches. Challenges in AMR parsing remain, particularly in achieving semantic consistency, cross-lingual adaptability, and structured reasoning.

3 Methods

Reinforcement learning has been increasingly used to fine-tune LLMs for various objectives beyond next-token prediction, such as improving helpfulness, harmlessness, or adherence to specific styles (Ouyang et al., 2022; Bai et al., 2022b). Techniques such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) are commonly used but often require training a separate critic model, which can be computationally expensive. GRPO (Shao et al., 2024) offers a more efficient alternative by using group-based relative ranking, making RL fine-tuning more accessible, especially for complex tasks with non-differentiable or noisy reward signals, as demonstrated in fields like mathematical reasoning (Shao et al., 2024), computer vision (Liang, 2025), and speech processing (Togootogtokh and Klasen, 2025).

3.1 Group Relative Policy Optimization

GRPO (Shao et al., 2024) is a reinforcement learning algorithm designed to fine-tune large language models efficiently by replacing the critic model in PPO with a baseline estimated from a group of sampled outputs. This eliminates the need for a learned value function, reducing computational overhead and memory requirements.

For each query q, GRPO samples a group of responses $\{o_1, \ldots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$, evaluates them using a reward model, and normalizes the rewards within the group. The policy is updated using a clipped importance-weighted objective, similar to PPO, but the advantage estimation relies on the relative performance within the sampled group rather than absolute reward values predicted by a critic. This encourages the policy to shift probability mass towards outputs that perform relatively better within the sampled group according to the reward function.

3.2 Our approach

We started with a vanilla Llama 3.2 1B model, which we fine-tuned using supervised fine-tuning (SFT). To avoid overfitting we used early stopping based on the validation loss. The training stopped after two epochs which resulted in the Llama-SFT model. After generating AMR graphs with Llama-SFT, we manually evaluated them and observed several recurring low-level structural and semantic errors. These errors primarily fell into two categories:

- **Frame-argument error**: Generated frames sometimes included arguments (e.g., ':arg4', ':arg5') that were not defined for that specific predicate sense in the PropBank frame files (Palmer et al., 2005). The arguments of each frame must strictly conform to the roles defined in its sense.

- **AND-OR node error:** Logical connective nodes like 'and' and 'or' require their operand roles (e.g., ':op1', ':op2', ':op3') to be consecutive integers starting from 1. We observed generated graphs violating this (e.g., having only ':op1' and ':op3'). A special case exists where only ':op2' appears, often used in AMR 3.0 for sentences starting with 'and' or 'or'; this specific structure was considered valid.

To address these issues and improve overall quality, we designed a composite reward function for GRPO incorporating four signals for each generated AMR graph:

- **Parsability:** A binary reward. The generated AMR graph must be parsable by standard AMR parsing tools without errors. Graphs that failed parsing due to structural or syntactic issues were penalized.

- **Frame-argument correctness:** A score between 0 and 1 representing the proportion of frames in the generated graph that adhere to their PropBank argument definitions. Calculated as (Number of valid frames) / (Total number of frames).

- **AND-OR node correctness:** A score between 0 and 1 representing the proportion of 'and'/'or' nodes with correctly structured operands (consecutive from ':op1', or the special ':op2'-only case). Calculated as (Number of valid AND/OR nodes) / (Total number of AND/OR nodes).

- **SMATCH score:** The Smatch F1 score (Cai and Knight, 2013) comparing the generated AMR graph against the gold reference AMR graph. This provides a global measure of semantic similarity.

These four criteria were combined into a single reward function, where each criterion was given equal weight. Additionally, we applied quadratic scaling to the SMATCH score, ensuring that lower scores received a higher penalty.

4 Dataset

For training, we used the AMR 3.0 dataset (LDC2020T02) (Banarescu et al., 2013), which provides a large collection of human-annotated Abstract Meaning Representation (AMR) graphs. We preprocessed the AMR graphs the following way. First, we removed wiki tags from the AMR graphs. Then, we serialized each graph into a single line using a depth-first approach. During serialization, new lines within the original graph notation were replaced with spaces, and consecutive spaces were compressed into a single space. Finally, we added spaces around parentheses to ensure consistent tokenization.

For evaluation, we used the AMR 3.0 test set and The Little Prince (**TLP**) corpus test set, which provides a smaller, out-of-domain evaluation with high-quality annotations. We measured the performance of the models using Smatch scores (Cai and Knight, 2013) computed with the `smatchpp` library (Opitz, 2023).

4.1 Dataset Statistics

Table 1 provides an overview of the dataset sizes used in our experiments.

Dataset	Number of Sentences
AMR 3.0 (Train)	55,635
AMR 3.0 (Test)	1,898
The Little Prince (test)	143

Table 1: Dataset statistics.

The combination of AMR 3.0 and the TLP dataset enables a comprehensive evaluation, balancing broad-domain performance with controlled, high-quality annotations.

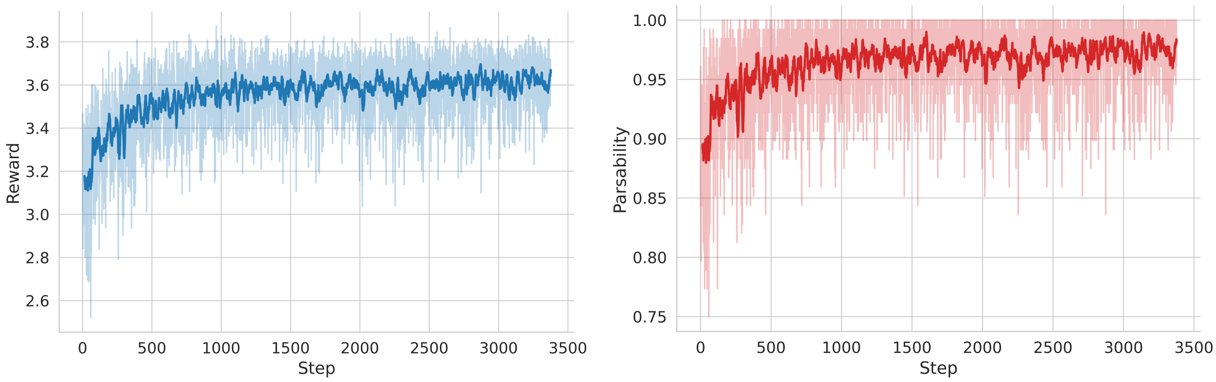

Figure 2: Evolution of average reward and parsability during GRPO fine-tuning on batches from the training set.

5 Results

We compare our Llama-SFT and Llama-GRPO models against SPRING. Table 2 shows the main results on the AMR 3.0 and TLP test sets.

The results show that Llama-SFT achieves a competitive Smatch score in two epochs, confirming the adaptability of LLMs to an unseen task. We observe that one epoch of subsequent GRPO fine-tuning yields further improvements. Llama-GRPO achieves a Smatch score of 81.92 on AMR 3.0, a gain of over 2.3 points over Llama-SFT, and it outperforms SPRING. In addition, GRPO improves compliance with the targeted low-level constraints. Frame-argument correctness improves from 96.5% to over 99% on AMR 3.0 and reaches 99.75% on TLP. Similarly AND-OR node correctness jumps from 96.5% to over 99.6% on AMR 3.0 and achieves perfect compliance on TLP. This demonstrates the effectiveness of incorporating these specific structural and semantic properties directly into the reward function via GRPO.

Figure 2 illustrates the progression of the overall reward during GRPO training. The reward score exhibits a smooth and consistent upward trend throughout the GRPO fine-tuning process. This indicates that the model has effectively learned to generate AMR structures that better satisfy these constraints, validating the utility of GRPO with these specific reward signals. The learning appears stable, without drastic fluctuations, suggesting that GRPO provides a reliable optimization process for these objectives.

5.1 GrAPES evaluation

According to Groschwitz et al. (2023), **Edge Recall** measures the parser's accuracy in identifying crucial semantic edges for specific phenomena. **Prerequisites** evaluate whether the parser generates the required graph structure to attempt to recognize these phenomena. Tables 3 and 4 summarize model performance accordingly.

From Tables 3 and 4 we see that SPRING demonstrates higher accuracy in Edge Recall, indicating slightly better capability in accurately identifying semantic edges once generated. This difference in performance for SPRING can potentially be explained by the AMR-specific adaptation of its tokenizer and vocabulary[1].

Llama-SFT on the other hand, consistently excels at Prerequisites, indicating that it more reliably constructs graph structures necessary for capturing complex phenomena, even if its edge-level precision is slightly lower.

Limitations

Our study has several limitations:

- **Model Scale:** We focused exclusively on the Llama 3.2 1B model due to resource limitations. Larger models or other LLM architectures might yield different baseline performance and respond differently to GRPO tuning.

- **Language Coverage:** Our experiments were conducted solely on English AMR. The applicability and effectiveness of this approach for other languages remain unexplored.

- **Reward Design:** While our fine-grained rewards proved effective, the specific combination and weighting could be further optimized.

[1] The set of possible edge labels is added to the vocabulary.

Model	Smatch++ F1		FRAME-ARG correctness		AND-OR correctness	
	AMR 3.0	TLP	AMR 3.0	TLP	AMR 3.0	TLP
Llama-SFT	79.58	78.06	0.96491	0.97550	0.96514	0.97887
Llama-GRPO	**81.92**	78.30	0.99178	**0.99758**	**0.99624**	**1.00000**
SPRING	80.15	**81.12**	**0.99396**	0.99703	0.95978	0.98501

Table 2: Comparison of different AMR parsers on AMR 3.0 and TLP datasets based on Smatch++ F1, ARG correctness, and AND-OR correctness.

Category	SPRING	SFT	GRPO
Pragmatic Coreference	42	**61**	**61**
Syntactic Reentrancies	61	56	**61**
Unambiguous Coreference	55	**84**	81
Rare Predicate Senses	79	82	**91**
Rare Edge Labels	55	**65**	**65**
Types of Seen NEs	83	83	**89**
Types of Unseen NEs	**64**	49	57
Frequent Predicate Senses	79	83	**90**
Passives	66	78	**80**
Unaccusatives	75	71	**79**
Ellipsis	79	82	**85**
Imperatives	72	67	**83**

Table 3: Prerequisites scores from GrAPES evaluation. Best results highlighted in bold. Llama-SFT and Llama-GRPO are abbreviated as SFT and GRPO respectively.

Category	SPRING	SFT	GRPO
Pragmatic Coreference	**31**	25	25
Syntactic Reentrancies	**46**	27	32
Unambiguous Coreference	52	**58**	55
Rare Edge Labels	20	**20**	18
Rare Node Labels	61	58	**65**
Unseen Node Labels	**54**	35	44
Rare Predicate Senses	30	**34**	**34**
Seen Names	84	83	**89**
Unseen Names	**70**	56	64
Seen Dates	74	88	**91**
Unseen Dates	71	82	**84**
Other Seen Ents	**88**	79	87
Other Unseen Ents	59	61	**64**
Types of Seen NEs	82	81	**87**
Types of Unseen NEs	**47**	31	39
Frequent Predicate Senses	70	72	**79**
Passives	59	**64**	**64**
Unaccusatives	**67**	58	65
Ellipsis	42	39	**48**
Multinode Word Meanings	68	**80**	78
Imperatives	50	42	**59**

Table 4: Recall and Edge Recall scores from GrAPES evaluation. Best results highlighted in bold. Llama-SFT and Llama-GRPO are abbreviated as SFT and GRPO respectively.

Exploring other potential reward signals related to AMR quality could yield further improvements.

- **Comparison Models:** We compared against the publicly available SPRING model. Comparisons against state-of-the-art closed models or models using proprietary data were not possible.

- **Dataset Contamination:** We did not investigate whether the dataset we used for evaluation was included in the pre-training data of the Llama 3.2 1B model, which could lead to information leakage that artificially inflates performance.

6 Conclusion

In this work, we investigated the application of the Llama 3.2 1B language model to AMR parsing, enhanced by Group Relative Policy Optimization (GRPO). We demonstrated that supervised fine-tuning establishes Llama as a competent baseline AMR parser. Subsequently, by incorporating fine-grained reward signals targeting Smatch, graph parsability, frame-argument correctness, and AND-OR node validity into a GRPO fine-tuning stage, we achieved significant improvements.

Our Llama-GRPO model not only outperformed its supervised counterpart (Llama-SFT) in Smatch scores, but also showed significantly better performance for crucial low-level semantic and structural constraints. Furthermore, evaluation using the GrAPES suite revealed that Llama-GRPO generated more complete graph structures (higher Prerequisites scores) necessary to capture diverse and complex linguistic phenomena, outperforming both Llama-SFT and SPRING while achieving competitive recall performance.

These results highlight the potential of combining moderately sized, openly available LLMs with efficient reinforcement learning techniques like

GRPO, guided by carefully designed reward functions, to tackle complex structured prediction tasks like AMR parsing. This approach allows for direct optimization of desired output properties beyond what is easily achievable with standard supervised learning alone.

Acknowledgments

This study was supported by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory Program, Hungary.

References

Xuefeng Bai, Linfeng Song, and Yue Zhang. 2022a. Semantic-based pre-training for dialogue understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 592–607.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022b. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12564–12573.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 16–23, Minneapolis, Minnesota. Association for Computational Linguistics.

Shibhansh Dohare, Harish Karnick, and Vivek Gupta. 2017. Text summarization using abstract meaning representation. *arXiv preprint arXiv:1706.01678*.

Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

Jonas Groschwitz, Shay Cohen, Lucia Donatelli, and Meaghan Fowlie. 2023. AMR parsing is far from solved: GrAPES, the granular AMR parsing evaluation suite. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10728–10752, Singapore. Association for Computational Linguistics.

Brad Jascob. 2024. amrlib: A python library for amr parsing, generation, and visualization. https://github.com/bjascob/amrlib.

Zoher Kachwala, Jisun An, Haewoon Kwak, and Filippo Menczer. 2024. REMATCH: Robust and efficient matching of local knowledge graphs to improve structural and semantic similarity. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1018–1028, Mexico City, Mexico. Association for Computational Linguistics.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Young-Suk Lee, Ramón Fernandez Astudillo, Radu Florian, Tahira Naseem, and Salim Roukos. 2023. Amr parsing with instruction fine-tuned pre-trained language models. *arXiv preprint arXiv:2304.12272*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xu Liang. 2025. Group relative policy optimization for image captioning.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

Nedjma Ousidhoum, Zhangdie Yuan, and Andreas Vlachos. 2022. Varifocal question generation for fact-checking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2532–2544, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Enkhtogtokh Togootogtokh and Christian Klasen. 2025. Voicegrpo: Modern moe transformers with group relative policy optimization grpo for ai voice health care applications on voice pathology detection. *arXiv preprint arXiv:2503.03797*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zdeňka Urešová, Jan Hajič, and Ondřej Bojar. 2014. Comparing Czech and English AMRs. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, pages 55–64, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chuan Wang, Sameer Pradhan, Xiaoman Pan, Heng Ji, and Nianwen Xue. 2016. CAMR at SemEval-2016 task 8: An extended transition-based AMR parser. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1173–1178, San Diego, California. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado. Association for Computational Linguistics.