An Empirical Study of Large Language Models for Vietnamese Abstract Meaning Representation Parsing

Nhat-Truong Dinh[1,2], Thanh-Trung Ngo[1,2], Quoc-Bao Trinh[1,2], Duc-Vu Nguyen[1,2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{22521575, 22521560, 22520125}@gm.uit.edu.vn **Correspondence:** vund@uit.edu.vn

Abstract

In this paper, we investigated open-source LLMs, including Llama 3, Phi-4, Gemma 3, and Qwen3, for Vietnamese Abstract Meaning Representation (AMR) parsing using instruction fine-tuning. As a result, Qwen3 achieved the highest F-score of 0.58 on the VLSP 2025 Semantic Parsing challenge, ranking among the top solutions on the private test leaderboard. The results showed that our pipeline was stable and robust, avoiding format errors such as mismatched parentheses and invalid AMR structures. An ablation study highlighted the crucial role of preprocessing (variable removal, wiki tag removal, and linearization) and post-processing (variable restoration) in preserving parsing quality. Fine-grained error analysis further revealed challenges in handling complex semantics, with particularly low F-scores for Semantic Role Labeling (0.24) and Reentrancies (0.13).

1 Introduction

Abstract Meaning Representation (Banarescu et al., 2013) is a semantic framework that represents sentence meaning as a graph, where nodes correspond to concepts and edges denote semantic relations. AMR has been successfully applied to a variety of NLP tasks such as question answering, information extraction, machine translation, and summarization (Song et al., 2019; Damonte et al., 2019; Bojar, 2014; Liao et al., 2018). Its key strength lies in abstracting away syntactic variations to focus on the underlying semantics, making it a powerful representation for deep language understanding.

AMR parsing, the task of converting natural language into AMR graphs, has achieved remarkable progress in English, particularly with transformer-based and large language models (LLMs) approaches (Bevilacqua et al., 2021; Bai et al., 2022). However, extending AMR parsing to Vietnamese introduces unique challenges. Unlike English, Vietnamese is an isolating language without inflectional morphology. Tense, aspect, and modality are expressed through particles (e.g., "đã," "đang," "sẽ") or by word order. Vietnamese also exhibits flexible syntax, frequent use of discourse markers, and high lexical ambiguity, which complicate semantic disambiguation. Moreover, the absence of large-scale annotated AMR corpora further limits the development of robust parsers. These characteristics highlight the need for tailored approaches rather than direct transfer from English-trained models.

Recent advances in LLMs such as Qwen (Bai et al., 2023; Yang et al., 2025), Llama 3 (Dubey et al., 2024), Gemma 3 (Team et al., 2025), and Phi-4 (Abdin et al., 2024) have shown strong potential in semantic parsing through supervised fine-tuning and adaptation techniques. Leveraging these models offers a promising path to overcome Vietnamese-specific challenges and build accurate parsers even with limited annotated data.

In this work, we develop a robust Vietnamese AMR parser by fine-tuning state-of-the-art LLMs on a specialized Vietnamese AMR dataset. Our contributions are:

- We introduce a novel pipeline for Vietnamese AMR parsing, integrating Qwen3, Gemma 3, Llama 3, and Phi-4, fine-tuned using supervised fine-tuning (SFT) with extensive preprocessing to ensure high-quality inputs.
- We evaluate our system on the VLSP 2025 Semantic Parsing Challenge[1], where Qwen3 achieved the best results on both the public and private leaderboards. We also provide a linguistic error analysis to highlight remaining challenges and opportunities for AMR in Vietnamese.

[1]VLSP 2025 Challenge on Semantic Parsing: https://vlsp.org.vn/vlsp2025/eval/visemparse

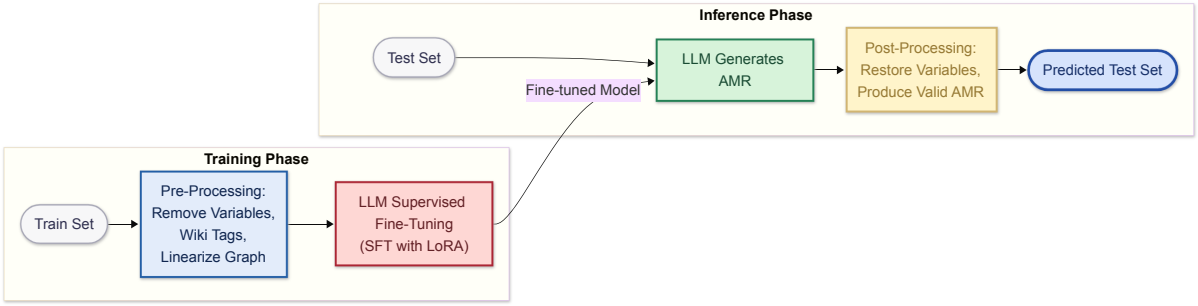


Figure 1: Our pipeline begins with preprocessing the data for fine-tuning the foundation model using supervised fine-tuning (SFT). The outputs generated by the fine-tuned model are then passed through a post-processing phase to ensure correct AMR parsing format.

2 Related Work

Early AMR parsers treated the task as a two-step process. JAMR (Flanigan et al., 2014) established the first widely adopted system, dividing the process into concept identification and relation prediction. CAMR (Wang et al., 2016) improved this idea by leveraging a dependency parser and a transition-based algorithm. With the constant development of transformer-based architecture, these models have become the dominant approach to AMR parsing. SPRING (Bevilacqua et al., 2021) builds on the BART backbone (Lewis et al., 2019) and demonstrates strong performance through large-scale pre-training combined with graph linearization. In the same way as SPRING (Bevilacqua et al., 2021), StructBart (Zhou et al., 2021) is extended by adding transition-based inductive biases to enforce structural constraints. Additionally, AMRBART (Bai et al., 2022) enhances BART with graph-aware pre-training, achieving state-of-the-art results.

In the context of Vietnamese, several studies have explored the adaptation of AMR for semantic parsing. A paper by Linh and Nguyen (2019) introduced modifications to the AMR framework to account for syntactic differences between English and Vietnamese, laying the foundation for applying AMR to annotate Vietnamese sentences.

Similarly, Pham (2020) applied cross-lingual semantic parsing by adapting frameworks like AMR and UCCA from high-resource languages to Vietnamese, addressing the challenges posed by the unique syntax of Vietnamese.

Additionally, AMR has been applied to legal texts to represent complex sentences. Vu et al. (2022) introduced the JCivilCode dataset (Viet et al., 2017) and proposed domain adaptation to improve AMR parsing and generation. Semantic Role Labeling (SRL), a sub-aspect of AMR, has been used in Vietnamese NLP: Duong et al. (2022) combine SRL with BERT for Recognizing Textual Entailment, and Le et al. (2022) integrate SRL into Retro-Reader for Machine Reading Comprehension, improving performance in domain-specific tasks.

These studies highlight the growing application of AMR in non-English languages like Vietnamese, despite the challenges of language-specific syntactic features. As shown in Figure 2, AMR can be represented as a graph with nodes and edges, capturing the semantic relationships between different concepts in a sentence.

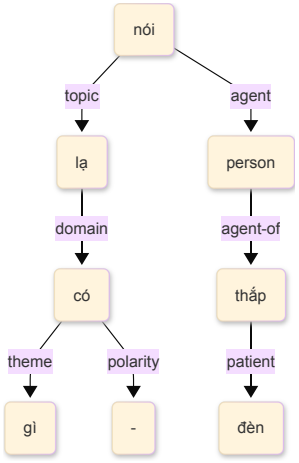


Figure 2: The AMR graph without variables and wiki tags for simplicity for the sentence "– chẳng có gì lạ cả, người thắp đèn nói" (in English: "– nothing strange, the person lighting the lamp said"). This example from the VLSP 2025 AMR dataset shows how nodes represent semantic concepts and edges represent semantic roles such as *agent*, *topic*, *theme*, and *patient*, defining the relationships between concepts.

Recently, LLMs such as Qwen3, Llama 3, and Gemma 3 have been applied to AMR parsing. Ho (2025) show fine-tuned decoder-only LLMs (Phi-4,

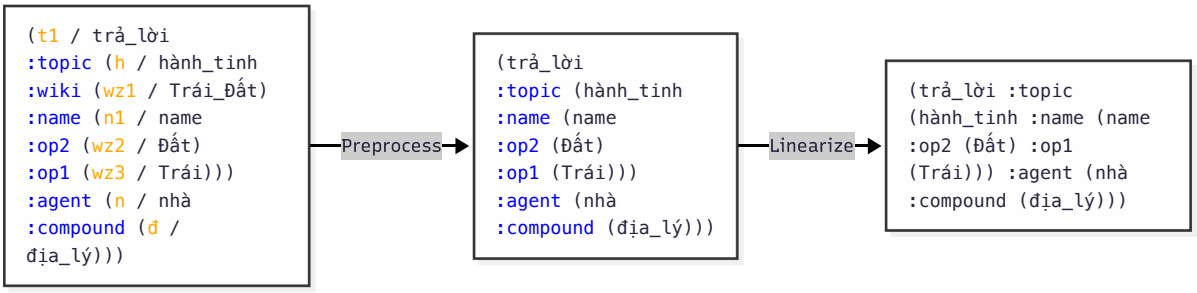


Figure 3: Three aligned views of the AMR preprocessing pipeline: (1) raw graph, (2) graph without variables and wiki tags, and (3) final linearized input for sequence processing.

Gemma 3, Llama 3) match parsers on Smatch (Cai and Knight, 2013). Similarly, Barta et al. (2025) show reinforcement learning methods like GRPO (Shao et al., 2024) further improve semantic and structural consistency. These results highlight the promise of modern LLMs for AMR parsing.

3 Method

Our pipeline consists of three main steps: data preprocessing (Section 3.1), post-processing (Section 3.2), and supervised fine-tuning (Section 3.3), each designed to ensure high-quality input, refined outputs, and optimized model performance. Figure 1 provides an overview of the entire pipeline.

3.1 Data Preprocessing

Our initial hypothesis was that the poor performance of the fine-tuned models was partly due to limitations in the training dataset. To validate this, we applied a systematic preprocessing pipeline designed to simplify the data while preserving its core semantic content (Figure 3).

The pipeline involves three key steps. First, variables, which serve only as placeholders for coreferring nodes (Van Noord and Bos, 2017), are removed to reduce unnecessary complexity. Next, wiki tags are eliminated to avoid noise and inconsistency. Finally, each graph is serialized into a single line through **linearization**, converting hierarchical structures into sequential format. Any new elements generated in this process, such as parentheses, are carefully managed to retain structural integrity.

This component is adapted from van Noord and Bos (2017)[2] and modified to accommodate the Vietnamese AMR parsing task. Further details of the original implementation are provided in (van Noord and Bos, 2017).

[2]https://github.com/RikVN/AMR

3.2 Data Post-processing

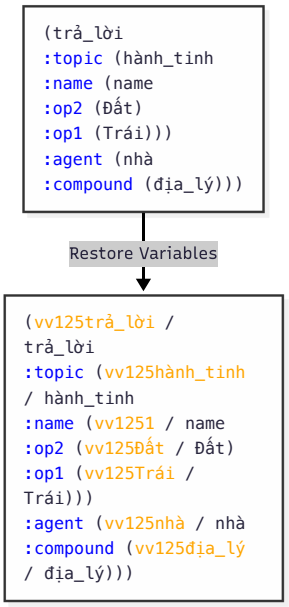


Figure 4: Transition from a variable-free AMR graph to a post-processed version with restored variables and unique concept IDs.

Following preprocessing and model fine-tuning, the outputs remained in AMR format but lacked node variables, resulting in invalid PENMAN graphs (Goodman, 2020). To address this, we applied a deterministic post-processing step that restores variables by assigning unique identifiers to each concept node (Figure 4). The restoration code also attempts to fix invalid AMRs using several heuristics, such as inserting missing parentheses and quotes or removing incomplete nodes. The procedure is based on van Noord and Bos (2017)[2]; however, due to time constraints, only basic variable restoration was implemented. More details can be found in (van Noord and Bos, 2017).

3.3 Supervised Fine-Tuning

Adapting pretrained LLMs to a specific dataset via supervised fine-tuning (SFT) (Dong et al., 2023) is a straightforward, cost-effective approach and has been widely used for LLM-related tasks in recent years. SFT aims to align the outputs of LLMs with the desired responses by replicating the style and patterns of the samples in the training dataset.

In this work, we employ Supervised Fine-Tuning for our foundation model, Qwen3-14B. While this model is capable of handling a wide range of multilingual language tasks, it lacks knowledge of AMR parsing and the required output format. SFT allows the model to adapt effectively to the AMR parsing task, enabling it to generate precise responses that adhere to the rules of AMR.

We also fine-tuned additional models, including Phi-4, Gemma 3, and Llama 3, using the same approach. Each model was trained on the Vietnamese AMR dataset, with specific attention to the nuances of the language. These models demonstrated varying levels of performance, with Qwen3-14B achieving the highest scores.

We selected a 4-bit quantized version of Qwen3-14B. Although quantization can lead to a slight reduction in performance, this trade-off is considered acceptable due to the significant decrease in computational cost.

To improve computational efficiency, we applied LoRA (Hu et al., 2021) fine-tuning, as it is both time-saving and resource-efficient compared to fully fine-tuning a model with 14 billion parameters. We configured the LoRA hyperparameters to R = 128 and LoRA alpha = 256.

4 Experiment

4.1 Data Statistics

In this study, we utilize the AMR parsing dataset from the VLSP 2025 Challenge on Semantic Parsing. Table 1 shows detail of datasets sizes.

Data Split	Number of Samples
Training Set	1,842
Public Test Set	150
Private Test Set	1,200

Table 1: Statistics of the VLSP 2025 AMR dataset.

The training dataset consists of 1,842 pairs, where each pair comprises a sentence and its corresponding AMR in PENMAN format. For further experimentation, we partitioned the VLSP AMR training set into shuffled training and validation splits with a 8:2 ratio, corresponding to 1,473 and 369 samples, respectively.

4.2 Experimental Setup

Based on experiments with the train–dev split (Section 4.4), we applied the data preprocessing pipeline (Section 3.1) to all samples in the VLSP 2025 AMR train dataset. We then performed LoRA-based fine-tuning using a 4-bit quantized Qwen3-14B as the base model. Hyperparameters were set to a learning rate of 2e-4 with the AdamW optimizer (Yao et al., 2021) and a weight decay of 0.01. Fine-tuning ran for 10 epochs with a batch size of 2 and gradient accumulation of 8 on a single RTX 4090 (24 GB VRAM) for 1–2 hours. For inference on the VLSP 2025 AMR private test set, we used 1 × A100 SMX4 (40 GB VRAM) for roughly 1 hour. Both fine-tuning and inference leveraged the Unsloth framework for computational efficiency.

4.3 Evaluation Metric

In this work, we use the Smatch[3] (Cai and Knight, 2013) score to evaluate the performance of the semantic parsing system. Smatch compares the predicted and reference AMRs by measuring the overlap of their matching triples. A triple is defined as a three-element tuple consisting of a source concept, a semantic relation, and a target concept, effectively representing a single edge in the AMR graph that encodes a specific semantic relationship. Precision is computed as the proportion of matching triples relative to the predicted AMR, while Recall is computed as the proportion of matching triples relative to the reference AMR. The Smatch score is reported as the F_1-score, calculated as

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}.$$

This metric provides a balanced evaluation of parser performance by simultaneously accounting for both correctness and completeness of the predicted semantic structures, making it a standard measure for AMR parsing tasks.

4.4 Main Result

To assess the impact of data processing on the performance of the fine-tuned model, we conducted

[3] https://github.com/snowblink14/smatch.git

experiments on the development dataset, with results summarized in Table 2. Without any preprocessing, the model performed poorly, achieving an F1-score of 0.18, and most generated graphs could not be parsed into valid AMR structures, highlighting the need for data cleaning. Applying a preprocessing step that removes variable names and applies linearization substantially improved performance, raising the F1-score to 0.65; in this setting, postprocessing was applied to restore the removed variables, producing consistently parsable outputs.

Method	Precision	Recall	F_1
No preprocessing	0.18	0.17	0.18
Var. removal & linear.	0.67	0.64	0.65
Full preprocessing	0.72	0.69	**0.71**

Table 2: Evaluation of the fine-tuned Qwen3 model on the development set under different preprocessing settings. "Var. removal & linear." refers to removing variable names and linearizing the AMR. Invalid predicted AMRs were replaced with a dummy to satisfy the Smatch library for analysis.

When full preprocessing was applied, including variable removal, wiki tag removal, and linearization of the fine-tuning data, together with post-processing of model outputs, performance further increased to an F_1-score of 0.71, with nearly all outputs in the correct AMR format, as shown in Table 2. These results demonstrate that the processing pipeline effectively enhances both the accuracy and consistency of the fine-tuned model.

Method	Dev	Public	Private
No preprocessing	6–10	8–10	–
Var. removal & linear.	2	0	–
Full preprocessing	1	**0**	**0**

Table 3: Number of incorrectly generated samples by the fine-tuned Qwen3 model on the VLSP 2025 AMR development, public, and private datasets under different preprocessing settings.

We also used the AMR library to manually inspect the number of samples that could not be parsed from the outputs of the fine-tuned model, typically due to issues such as duplicate nodes, extraneous commas, or unbalanced parentheses, either missing or redundant. As shown in Table 3, without any processing, roughly 6 to 10 instances in the development set and 8 to 10 in the public set failed to parse. When full processing was applied, including variable removal, wiki tag removal, and linearization of the fine-tuning data, together with post-processing of the model outputs, only one unparsable case remained in the development set, while both the public and private sets were parsed without errors. Note that in both the variable-removal and full-processing experiments, variable restoration was applied to the model outputs before performing the AMR parsing check. These results on the public and private VLSP datasets indicate that the post-processing step does not introduce any additional errors.

Model	Public Test	Private Test
Llama-3.1-8B	0.45	0.50
Phi-4	0.51	0.55
Gemma-3-12B	0.52	0.56
Qwen3-14B	**0.54**	**0.58**

Table 4: Evaluation of different models using F_1-score on the VLSP 2025 AMR public and private test sets after full preprocessing.

Table 4 presents the evaluation of different models using F_1-score on the VLSP 2025 AMR public and private test sets after full data processing. With full data processing and LoRA-based fine-tuning, our Qwen3-14B achieved a Smatch F_1-score of 0.54 on the public dataset and 0.58 on the private dataset. These results indicate that our pipeline consistently generates correct semantic representations in PENMAN format while effectively avoiding invalid outputs, as illustrated in Table 3, where the model exhibits zero parsing failures on both datasets. We further fine-tuned and evaluated Gemma-3-12B, Phi-4, and Llama-3.1-8B using the same full preprocessing pipeline and observed consistent performance across both test sets. Qwen3-14B was ultimately selected as the final model due to achieving the highest Smatch F_1-score on both public and private datasets.

The experiments in Tables 2, 3, and 4 show that data processing is crucial for consistent AMR parsing. Without it, the fine-tuned model produced many unparsable outputs (Table 3) and lower F_1-scores, whereas variable removal and full processing significantly reduced errors and improved performance. Results on the VLSP public and private test sets (Table 4) further confirm that preprocessing the fine-tuning data and post-processing model outputs are essential for model robustness and reliability.

4.5 Fine-grained Error Analysis

According to (Damonte et al., 2017), while AMR parsing involves a large number of subtasks, the Smatch score consists of a single number that does not assess the quality of each subtasks separately. Furthermore, the Smatch score weighs different types of errors in a way which is not necessarily useful for solving a specific NLP problem. Therefor, we applied the evaluation tool from (Damonte et al., 2017) to gain a better insight of our best model's performance on public test set.

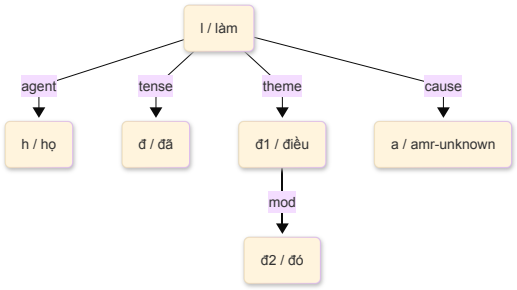
Method	Precision	Recall	F_1
Smatch	0.55	0.51	**0.54**
Unlabeled	0.66	0.60	**0.63**
No WSD	0.55	0.51	**0.54**
Concepts	**0.66**	0.62	**0.64**
Named Ent.	0.65	0.48	0.55
Negations	0.53	0.53	0.53
Reentrancies	0.23	**0.09**	0.13
SRL	0.25	0.23	0.24

Table 5: Fine-grained evaluation of the public test set with the evaluation suite from (Damonte et al., 2017).

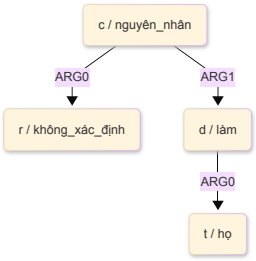
Based on a fine-grained error analysis, as detailed in Table 5, the parser demonstrates significant strengths in foundational areas. It excels at identifying concepts, achieving the highest F_1-score of 0.64 in this subtask, which is crucial for the overall parsing process. The model is also proficient at determining the correct graph structure, indicated by a high Unlabeled F_1-score of 0.63. Furthermore, the analysis reveals that word sense disambiguation is not a major source of error, as the "No WSD" F_1-score is identical to the overall Smatch score, suggesting the parser handles this aspect effectively.

However, the parser shows considerable weaknesses in more complex semantic and structural tasks, as shown in Table 5. The most significant issues are with SRL and Reentrancies, which scored very low F_1-scores of 0.24 and 0.13, respectively. This indicates that while the model can connect predicates and arguments, it often fails to assign the correct semantic roles and struggles to handle graph structures requiring a node to have multiple parents. Additionally, the parser has moderate performance on named entities with a low recall, and it correctly identifies only about half of the negations.

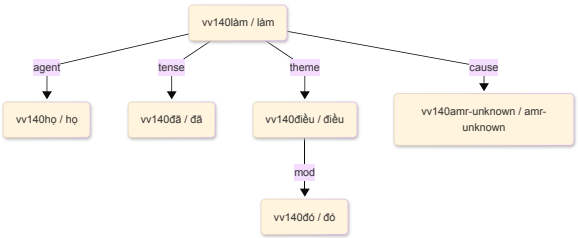
4.6 Case Study



(a) Gold graph representing the correct structure.



(b) Graph predicted by the model without any preprocessing.



(c) Graph predicted by the model with full preprocessing.

Figure 5: Gold annotation (a) and two model predictions (b) and (c) for the input sentence "vì sao họ đã làm điều đó?", with and without preprocessing, from the public test set.

The input sentence "vì sao họ đã làm điều đó?" translates to "why did they do that?". The gold annotation (Figure 5a) correctly identifies "làm" (do) as the central action, with "họ" (they) as the agent, "đã" indicating past tense, "điều đó" (that) as the theme, and an amr-unknown node representing the questioned cause. In contrast, the prediction without preprocessing (Figure 5b) fundamentally misinterprets the sentence's structure. It incorrectly centers the graph on "nguyên_nhân" (cause) rather than the main verb "làm". Consequently, it fails to represent the tense ("đã") and the theme ("điều đó") and misstructures the relationship between the agent and the action. With full preprocessing,

the model's prediction (Figure 5c) is structurally identical to the gold graph. It accurately captures the central verb and all its corresponding semantic components, including the agent, tense, theme, and cause. This comparison highlights the critical importance of preprocessing for achieving accurate Vietnamese AMR parsing.

5 Discussion

Our experiments highlight several important details. LoRA-based fine-tuning combined with preprocessing proved effective in stabilizing training and enabling large models such as Qwen3-14B to achieve the best performance. Without such processing, preliminary experiments (Table 2) produced invalid PENMAN outputs or unparseable graphs. This confirms that structured semantic parsing for Vietnamese requires careful data processing in addition to model scaling.

Moreover, both Gemma-3-12B and Phi-4 achieved stable but moderate results. Their performance gap with Qwen3-14B highlights that some architectures may generalize less effectively to Vietnamese AMR parsing, even when provided with the same processing pipeline and fine-tuning strategy.

A domain-specific insight is that Vietnamese AMR parsing remains challenging. Even the best-performing system reached 0.58 F_1 on the private test set, indicating that there is still room for further development, particularly in addressing data sparsity, linguistic complexity, and annotation consistency. In addition, it would be beneficial to explore augmentation techniques, such as introducing greater variability in long queries together with their corresponding complex AMR structures in PENMAN format.

6 Conclusion

This study demonstrates that data processing and LoRA-based fine-tuning are crucial for achieving reliable AMR parsing in Vietnamese. Without these steps, we observed that models frequently produced invalid PENMAN outputs and unstable performance. By applying our full data processing pipeline, we were able to significantly improve both parsing accuracy and structural validity.

Among the evaluated models, Qwen3-14B consistently achieved the best results, with Smatch F_1-scores of 0.54 on the public test set and 0.58 on the private test set, ranking as one of the top solutions on both public and private leaderboards. Its superior performance highlights the effectiveness of combining large-scale model capacity with a carefully designed adaptation strategy. Future research should further investigate and experiment with alternative solutions to advance performance within this domain.

Limitations

The study's primary limitation is the modest performance of its top model, which achieved a 0.58 F_1-score, indicating that significant challenges in Vietnamese AMR parsing remain. A fine-grained error analysis reveals the model struggles with complex semantic structures, performing poorly on Semantic Role Labeling and Reentrancies with F_1-scores of just 0.24 and 0.13, respectively. Furthermore, because the paper's scope was defined by the VLSP 2025 Challenge on Semantic Parsing, it does not provide a comparative analysis against established AMR parsers like SPRING or StructBART, making it difficult to contextualize its performance within the wider field. Finally, the reliance on a large language model makes the system computationally expensive and resource-intensive, which poses a practical barrier to its scalable application for parsing big data and contributes to a significant carbon footprint.

Acknowledgement

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund. We also thank the anonymous reviewers for their time and helpful suggestions that improved the quality of this paper.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for amr parsing and generation. *arXiv preprint arXiv:2203.07836*.

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.

Botond Barta, Endre Hamerlik, Milán Nyist, Masato Ito, and Judit Ács. 2025. Enhancing amr parsing with group relative policy optimization. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 99–105.

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 AAAI*.

Zdenka Urešová Jan Hajic Ondrej Bojar. 2014. Comparing czech and english amrs. In *Workshop on Lexical and Grammatical Resources for Language Processing*.

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.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. *arXiv preprint arXiv:1903.04521*.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Quoc-Loc Duong, Duc-Vu Nguyen, and Ngan Luu-Thuy Nguyen. 2022. Leveraging semantic representations combined with contextual word representations for recognizing textual entailment in vietnamese. In *2022 9th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 47–52.

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.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for amr graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319.

Shu Han Ho. 2025. Evaluation of llms in amr parsing. *arXiv preprint arXiv:2508.05028*.

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. *Preprint*, arXiv:2106.09685.

Hang Thi-Thu Le, Viet-Duc Ho, Duc-Vu Nguyen, and Ngan Luu-Thuy Nguyen. 2022. Integrating semantic information into sketchy reading module of retro-reader for vietnamese machine reading comprehension. In *2022 9th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 53–58.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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.

Ha Linh and Huyen Nguyen. 2019. A case study on meaning representation for vietnamese. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 148–153.

Tessa Pham. 2020. Semantic parsing for vietnamese: A cross-lingual approach.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 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.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Rik Van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *arXiv preprint arXiv:1705.09980*.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.

Lai Dac Viet, Vu Trong Sinh, Nguyen Le Minh, and Ken Satoh. 2017. Convamr: Abstract meaning representation parsing for legal document. *arXiv preprint arXiv:1711.06141*.

Sinh Trong Vu, Minh Le Nguyen, and Ken Satoh. 2022. Abstract meaning representation for legal documents: an empirical research on a human-annotated dataset. *Artificial Intelligence and Law*, 30(2):221–243.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. 2021. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10665–10673.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based amr parsing. *arXiv preprint arXiv:2110.15534*.