

System Report for CCL25-Eval Task 3: Hallucination Mitigation in Chinese Abstract Meaning Representation Parsing with a Multi-Agent Approach

Rongbo Chen, Xuefeng Bai[†], Kehai Chen, Min Zhang

Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen

24s151006@stu.hit.edu.cn

Abstract

This paper introduces our system for the Fifth Chinese Abstract Meaning Representation (CAMR) Parsing Evaluation task at the 24th China National Conference on Computational Linguistics (CCL 2025). Our framework formulates both CAMR parsing and document-level coreference resolution as sequence-to-sequence generation tasks, employing large language models (LLMs) to produce linearized CAMR sequences and coreference sequences. To mitigate hallucinations in generated graphs, we design a multi-agent system comprising: (1) two detection agents for automated error detection and hallucination identification; (2) a refinement agent that corrects graph structures based on detected inconsistencies. Experimental results show that: (1) recent LLMs, especially Qwen-3, achieve promising performance in CAMR parsing; (2) the proposed multi-agent system can effectively identify and correct hallucinations of CAMR predictions; and (3) sequence-to-sequence methods exhibit significant limitations in document-level coreference resolution due to context length constraints.

Keywords: CAMR, Multi-agent, Hallucination, Large language model

1 Introduction

Abstract Meaning Representation (AMR) abstracts words into concepts and represents semantic relations between concepts as directed edges. By linking concepts through labeled directed relations, AMR forms a single-rooted directed acyclic graph (DAG) to capture the full semantic structure of a sentence (Banarescu et al., 2013). Chinese Abstract Meaning Representation (CAMR) extends this paradigm to the Chinese language, incorporating characteristics of Chinese and adding concept alignment, relation alignment, and more fine-grained semantic labels (Li et al., 2016; Li et al., 2019). Due to its expressive semantic structure, AMR has been widely applied in downstream NLP tasks such as dialogue systems (Bai et al., 2021) and machine translation (Li and Flanigan, 2022; Nguyen et al., 2021), covering a variety of applications including text generation, text classification, and information extraction (Mansouri, 2025).

CAMR and AMR parsing refer to the process of predicting an AMR graph from a sentence (Bai et al., 2022b; Bai et al., 2022a). Current approaches mainly fall into three categories: (1) Transition-based parsing, which applies a series of meticulously engineered transition actions to the dependency tree to incrementally build AMR graphs (Wang et al., 2015); (2) Graph-based parsing, which directly predicts nodes (concepts) and edges (relations) from input text, then combines them into a completed AMR graph (Zhang et al., 2019); and (3) Seq2Seq-based parsing, which linearizes AMR graphs into sequences using predefined rules and generates AMR sequences as target sequences, greatly simplifying the parsing process (Bevilacqua et al., 2021; Huang et al., 2021).

Recently, Large Language Models (LLMs) have showcased promising capacities for solving complex problems, due to their broad knowledge and outstanding performance across various tasks. In the field

[†] Corresponding Author

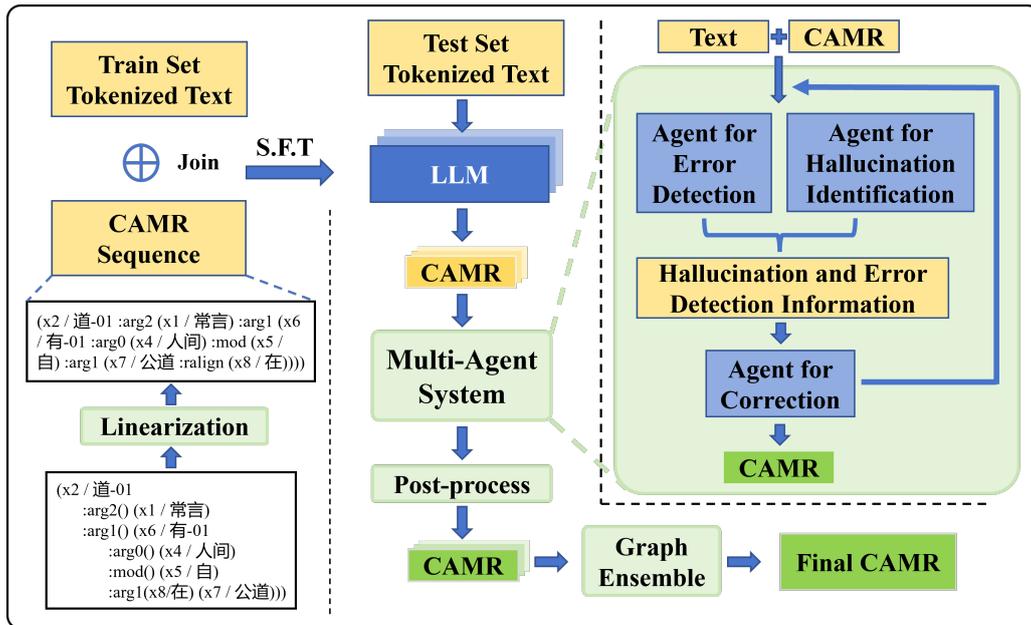


Figure 1: Workflow of sentence-level parsing task

of CAMR parsing, LLMs demonstrate strong capabilities in generating structured semantic graphs. For example, based on the classification of clause types in Chinese sentences, Wu et al. (2024) fine-tuned a large language model to develop a hybrid LoRA-based expert parsing framework for CAMR parsing; by employing a depth-first traversal-based serialization method, Gao et al. (2023) fine-tuned Baichuan-7B and obtained a robust CAMR parsing system, without relying on part-of-speech tags, dependency syntax, or complex rules. However, our preliminary analysis reveals that LLMs frequently hallucinate content inconsistent with source inputs, significantly compromising the faithfulness of generated graphs and reducing overall performance.

To address this, we introduce a multi-agent collaboration framework which aims to automatically identify and correct the hallucinations of the graphs generated by LLMs in a post-processing way. Our framework employs two independent LLM-based agents for error and hallucination detection, and another LLM-based agent to refine the predicted graph based on the feedback from two detection agents. We finetune the open-sourced LLMs to perform CAMR parsing and document-level coreference resolution, and employ GPT-4.1-mini as the backbone of the proposed agents. Experimental results show that: (1) recent large language models, especially Qwen-3, achieve promising performance (81.63 align-smatch) in CAMR parsing; (2) our multi-agent framework could effectively identify and correct hallucinations in predicted CAMR graphs; and (3) sequence-to-sequence methods exhibit limitations in document-level coreference resolution due to context length constraints. Our final system achieves scores of **82.03** and **76.80** on the sentence-level CAMR parsing tasks TestA and TestB, respectively, and for the document-level coreference resolution task, a labeled score of 18.20 and an unlabeled score of 61.15.

2 Methodology

CAMRP2025 task is divided into two subtasks: sentence-level CAMR parsing and document-level coreference resolution. In the sentence-level parsing task (Task A), the train and validation sets provide sentence-level CAMR graph representations, CAMR tuple representations, and dependency trees. The test set provides the sentences to be parsed along with their dependency trees. The goal of Task A is to parse the corresponding CAMR semantic graphs for the target sentences. In the document-level coreference resolution task (Task B), the validation set provides the sentence-level CAMR graphs for each sentence in a document, as well as all entity coreference chains of CAMRs within the document. The test set provides the document text and the corresponding sentence-level CAMR graphs. The goal

```

Before document CAMRs linearization:
sentence#1 x1_上海 x2_浦东 x3_开发 x4_与 x5_法制 x6_建设 x7_同步
CAMR#1 (x7 / 同步-01
      :arg0() (x24 / and
      :op1() (x3 / 开发
      :arg1() (x13 / city-district
      :name() (x2 / name :op1 x2/浦东 )
      :location() (x20 / city
      :name() (x1 / name :op1 x1/上海 )))
      :op2(x4/与) (x6 / 建设-01
      :arg1() (x5 / 法制)
      :arg0() (x25 / x13))))

sentence#N
CAMR#N

After document CAMRs linearization:
sentence#1 x1_上海 x2_浦东 x3_开发 x4_与 x5_法制 x6_建设 x7_同步
CAMR#1 (x7 / 同步-01 :arg0(x24 / and :op1(x3 / 开发 :arg1(x13 / city-district
      :name(x2 / name :op1 x2 / 浦东) :location(x20 / city :name(x1 / name
      :op1 x1 / 上海)))) :op2(x6 / 建设-01 :arg1(x5 / 法制) :arg0(x25 / x13))))

sentence#N
CAMR#N

```

Figure 2: Linearized sentences and CAMRs of the document-level coreference resolution task

of Task B is to predict the entity coreference chains within the document. More specifically, based on the CAMR representation, the task involves extracting coreference relations between concept nodes in the CAMR graphs of individual sentences in the document.

This work adopts a sequence-to-sequence based CAMR parsing approach. For the sentence-level parsing task (as showed in Figure 1), we convert the CAMR graphs into a linear sequence, concatenate it with the tokenized input, and fine-tune LLMs in an autoregressive manner. Then we correct the hallucinations and format errors of the predicted CAMR sequences for the test set using a multi-agent system and post-process, and merge the CAMR graphs from multiple models using graph-ensemble algorithm to produce the final output.

For the document-level coreference resolution task, we linearize and concatenate the tokenized sentences, their corresponding CAMR graphs, and the coreference chains within each document from the validation set to fine-tune the LLMs. After prediction, we post-process the output coreference chains to restore their correct format.

2.1 CAMR Linearization

In CAMRP2023¹, SUDA² adjusted the formats of function word alignment and concept coreference, and removed parentheses used for function word alignment, aligning the CAMR graph format with that of AMR graphs. In addition, the “name” concepts in named entities were removed, and the “:op” relations contained within the “name” concepts were merged to align with the representation of named entity tuples in the final submitted tuple file. This simplification reduced the complexity of CAMR sequences and parsing. This work adopts the effective linearization method to generate single-line CAMR sequences for training.

2.2 Document CAMR Linearization

In the document-level coreference resolution task, the goal is to extract entity coreference chains. Considering that the function word alignment information in sentence-level CAMR graphs within a document is not closely related to the task objective, we remove such alignment information during

¹<https://github.com/GoThereGit/Chinese-AMR/tree/main/CAMRP%202023>

²<https://github.com/EganGu/camr-seq2seq>

<p>Before coreference chain linearization:</p> <pre>p0001_r1_entity :root() s1_:arg1_x13 / city-district :zero() s1_:arg0_x25 / x13 :homo() s3_:arg0_x41 / city-district :zero() s3_:arg0_x51 / x41 :reduce() s3_:arg1_x30 / x41</pre> <p>After coreference chain linearization:</p> <pre>entity_1 :root(s1 :arg1(x13 / city-district)) :zero(s1 :arg0(x25 / x13)) :homo(s3 :arg0(x41 / city-district)) :zero(s3 :arg0(x51 / x41)) :reduce(s3 :arg1(x30 / x41))</pre>

Figure 3: Linearized coreference chain of the document-level coreference resolution task

CAMR graph linearization to minimize context length. Concept coreference information, however, is retained, as it contains a large number of coreference relations in the chains, such as “:zero()” and “:pro()”. An example of the linearized result is shown in Figure 2.

For the linearization of coreference chains, we align the format of coreference relations in the chains with that of the relations in CAMR graphs, to simplify the extraction process for LLMs. In addition, since only coreference relations, sentence IDs, coreferent token IDs, and the tokens themselves are needed, we remove parentheses, annotations, and document order information from the coreference chains. An example of the final linearized result is shown in Figure 3.

2.3 Sentence-level CAMR Parsing Based on Supervised Fine-tuning

This paper employs a supervised learning paradigm to train large language models with either LoRA or full-parameter fine-tuning. We concatenate task prompt, tokenized text, and CAMR sequence for autoregressive prediction. The instruction prompt is constructed as follows:

请输出以下中文文本的中文抽象语义解析序列: “x1_各 x2_方 x3_势力 x4_动用 x5_自己 x6_的 x7_所有 x8_宣传 x9_力量 x10_你 x11_方 x12_唱罢 x13_我 x14_登场”

Let \mathcal{D} denote the train set, I the task instruction prompt, x the tokenized text, and y the CAMR sequence. We train the model by optimizing the following loss function:

$$\mathcal{L} = - \sum_{\{I,x,y\} \in \mathcal{D}} \log P(y | I \oplus x), \quad (1)$$

\oplus denotes the sequence concatenation operation. Including the document-level coreference resolution task, we use a gradient descent-based method for parameter updates.

2.4 Document-level Coreference Resolution Based on Supervised Fine-tuning

Due to the long context in linearized document-level samples, we adopt a supervised learning approach and apply LoRA fine-tuning to large language models to reduce GPU memory usage. We concatenate task prompt, the tokenized text and CAMR sequences of each sentence in the document, as well as the coreference chain sequence for autoregressive prediction. The instruction prompt is constructed as follows:

请根据给定的篇章文本和其中文抽象语义解析, 提取该篇章中的共指链序列: sentence#1 CAMR#1 ... sentence#N CAMR#N

Let \mathcal{D} denote the validation set, I the task instruction prompt, x_i the i -th tokenized sentence in the document along with its CAMR sequence, and y the coreference chain sequence. We train the model by optimizing the following loss function:

$$\mathcal{L} = - \sum_{\{I,x_1,\dots,x_n,y\} \in \mathcal{D}} \log P(y | I \oplus x_1 \oplus \dots \oplus x_n), \quad (2)$$

2.5 Multi-Agent System for Hallucination Identification and Error Detection

Due to the complexity of CAMR structures, the CAMR sequences predicted by large language models may contain format errors or even hallucinations. Format errors mainly include mismatched parentheses, missing or incomplete concept node or relation information, etc. Hallucinations may involve concepts that are irrelevant to the context, concepts whose meanings differ from the corresponding words in the original sentence, or invalid concept variables referenced by coreference relations.

To address these issues and explore the collaborative capabilities of multi-agent systems in handling complex CAMR structures, we design a multi-agent unsupervised system consisting of an agent for error detection, an agent for hallucination identification, and an agent for correction (Figure 1). The agent for error detection is responsible for checking the format error of predicted CAMR sequence and providing possible suggestions for correction. The agent for hallucination identification is responsible for checking the CAMR sequence for potential hallucinations by referring to the original sentence and proposing modifications to alleviate hallucination issues. The agent for correction analyzes the issues and suggestions provided by the other two agents and ultimately outputs a valid CAMR sequence. Then, the sequence is fed back to the other two agents for reflection, repeating the process until no errors or hallucinations remain.

2.5.1 Prompt Design for Agents

The prompts of the agent for error detection and the agent for hallucination identification are provided in [Appendix A](#). These two prompts share a similar structural framework, differing only in task functionality and the provided examples. The prompts mainly include the following components: task description, input format specification, CAMR sequence structure information, error checking list, special notes, output format specification, and both positive(valid) and negative(invalid) examples.

The task description provides a comprehensive overview of the agent’s role and clearly defines the input and output. The input and output format specifications enforce strict format to facilitate accurate information extraction. A particularly important requirement is that if no errors are found in the input CAMR sequence, the agent must explicitly indicate that the sequence is valid. CAMR structure information enables the agent to accurately understand the meaning of each part(concept, relation, etc.) in the CAMR sequence, thereby avoiding invalid error detections caused by misunderstanding the correct format. The error checking list lists potential format issues, guiding the agent to conduct step-by-step analysis through chain-of-thought reasoning. The special notes highlight easily overlooked or misunderstood parts of the CAMR structure, helping to prevent “negative optimization” issues. Positive and negative examples provide few-shot learning samples to help the agent better understand proper CAMR format and analyze input CAMR sequence.

The prompt of the agent for correction is provided in [Appendix B](#), includes the following: task description, input format specification, CAMR structure information, special notes, output format specification, and correction examples. The correction examples provide the agent with reference cases, including the input tokenized sentence, the CAMR sequence to be modified, the “[Detect]” issues provided by the other two agents, the step-by-step correction process, and the final corrected CAMR sequence. The other components are similar to those in the prompt of the agent for error detection. Notably, to enhance collaboration between agents, the agent for correction is explicitly instructed to critically reflect on the issues and suggestions provided by the other agents, avoiding blind acceptance and improving correction accuracy.

2.5.2 I/Os for Agents

The input to the agent for error detection and the agent for hallucination identification includes the target tokenized sentence and the CAMR sequence predicted by the LLMs. If the agents find some problems in sequence, they will output the detected issues and revision suggestions; if not, they will output a statement indicating the sequence is valid.

The input to the agent for correction includes the target tokenized sentence, the predicted CAMR sequence, and the issues and suggestions from the other two agents. It outputs the corrected, valid CAMR sequence.

Model	Total			Instance	Attribute	Relation
	P	R	F ₁	F ₁	F ₁	F ₁
LLaMA-3.1	78.99	79.69	79.34	85.37	93.98	64.24
Glm-4	79.72	80.11	79.91	85.38	94.34	65.32
Qwen-2.5	80.00	79.66	79.83	85.37	94.19	65.41
Qwen-3	81.93	81.34	81.63	86.59	94.82	68.45
Ensemble	82.70	81.37	82.03	87.25	95.10	68.83

Table 1: Align-smatch scores of different models on TestA

2.6 Post-processing

After correction by the multi-agent system, the CAMR sequences may still not fully conform to the required specifications and thus require a post-processing step. The post-processing task is similar to that of the multi-agent system but focuses primarily on adjusting structural issues in invalid formats, such as mismatched brackets, missing concept node information, and other related problems.

2.7 Graph Ensemble

We adopt the graph ensemble algorithm (Hoang et al., 2021) to merge and enhance the predictions from multiple models, resulting in the final CAMR graph. The graph ensemble algorithm is a heuristic approach designed for AMR graph structures. It aggregates concept nodes and relations across multiple AMR graphs through a voting mechanism to produce the optimal AMR graph.

3 Experiment

3.1 Experimental Setup

CAMRP2025 includes two subtasks—sentence-level CAMR parsing and document-level coreference resolution.

Data: For the sentence-level parsing task, the train set contains 16,576 samples, the validation set contains 1,789 samples, and the test sets A and B contain 1,713 and 1,999 samples respectively. For the document-level coreference resolution task, the validation set includes 300 documents, and the test set includes 200 documents.

Models: For the parsing tasks, the large language models selected for supervised fine-tuning include LLaMA-3.1³, GLM-4⁴, Qwen-2.5⁵, and Qwen-3^{6,7}. For the multi-agent system, all agents are based on GPT-4.1-mini.

Parameter settings: For the sentence-level parsing task, LoRA fine-tuning is conducted for 8 epochs with a rank of 32 for the LoRA matrices. Full-parameter fine-tuning is conducted for 5 epochs. The maximum sequence length is set to 2048. Experiments use the AdamW optimizer (Loshchilov and Hutter, 2019) for optimization. Learning rate is selected from 2e-5, 3e-5, 5e-5, 8e-5, 1e-4. A cosine annealing schedule is adopted for learning rate decay, and the warmup ratio is set to 0.1. For the document-level coreference resolution task, due to the longer context and higher memory demands, only LoRA fine-tuning is used. The maximum sequence length is set to 8192, with 5 training epochs and a learning rate of 3e-5.

Evaluation metrics: Align-Smatch (Xiao et al., 2022) is used as the evaluation metric to assess the tuples contained in the CAMR graphs.

3.2 Experimental Results

Sentence-level CAMR parsing performance: Table 1 lists the performance of each model on the sentence-level parsing task after supervised fine-tuning, as well as the final results obtained through

³<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

⁴<https://huggingface.co/THUDM/glm-4-9b-chat-hf>

⁵<https://huggingface.co/Qwen/Qwen2.5-14B-Instruct>

⁶<https://huggingface.co/Qwen/Qwen3-14B>

⁷<https://huggingface.co/Qwen/Qwen3-32B>

Model	TestA			TestB		
	P	R	F ₁	P	R	F ₁
SUDA-HUAWEI ₂₂	82.16	78.20	80.13	75.52	71.79	73.61
ECNU ₂₂	73.83	66.05	69.72	66.01	57.71	61.58
BUPT ₂₂	50.41	42.55	46.15	49.95	42.72	46.05
GDUFE ₂₃	75.53	75.60	75.56	69.71	67.33	68.50
SJTU ₂₃	47.41	46.45	46.92	46.44	45.68	46.06
SUDA ₂₃	80.82	81.79	81.30	74.39	75.03	74.71
WestlakeNLP ₂₃	74.40	70.24	72.26	70.42	68.63	69.52
GDUFE ₂₄	80.87	81.53	81.19	74.80	75.75	75.27
BLCU ₂₄	78.93	78.89	78.91	74.04	74.27	74.16
Ours	82.70	81.37	82.03	76.87	76.73	76.80

Table 2: Comparison of different CAMR parsing systems.

graph ensemble. Generally, Qwen-3 achieves a precision of 81.93, recall of 81.34, and F₁ of 81.63. Glm-4 achieved a precision of 79.72, recall of 80.11, and F₁ of 79.91. After categorizing the tuples into three types—Instance, Attribute, and Relation—we evaluate each type separately. Among all single-model results, Qwen-3 achieves the highest F₁ scores across all three categories, with scores of 87.25 for Instance, 95.10 for Attribute, and 68.83 for Relation. Furthermore, after applying graph ensemble algorithm, the resulting CAMR graphs show improvements in all metrics: precision (82.70), recall (81.37), and F₁ (82.03). Compared to the models above, the F₁ scores for separate evaluations on Instance, Attribute, and Relation tuples have also improved across the board: Instance (87.25), Attribute (95.10), Relation (68.83). This indicates that the graph ensemble algorithm effectively mitigates noise and random errors in individual model predictions through voting, thereby enhancing parsing precision.

Comparison with previous CAMR parsing systems: Table 2 presents our final scores compared with the results from previous evaluation tasks. Our method achieves scores of **82.03** on Test A and **76.80** on Test B. Notably, the score on Test A surpasses the record 81.30 set by SUDA in CAMRP2023, and the score on Test B exceeds the record 75.27 set by GDUFE in CAMRP2024. The reason can be two folds: 1) the baseline LLMs used in this paper gives relative better performance than previous ones; 2) the proposed multi-agent framework achieves further improvements over LLMs.

Performance on document-level coreference resolution: Table 3 presents our final scores on the document-level coreference resolution task. Since this is the first evaluation of the document-level task, there are no existing methods available for comparison. Our results achieve a score of 18.20 on the labeled metric and 61.15 on the unlabeled metric. We provide further analysis in Section 3.3.

Model	Unlabel			Label		
	P	R	F ₁	P	R	F ₁
Ours	75.28	57.85	61.15	16.86	21.67	18.20

Table 3: Scores of document-level coreference resolution task

3.3 Experimental Analysis

Impact of the multi-agent framework: Table 4 presents the performance of multi-agent framework. Through the correction of multi-agent system, the align-smatch score of 153 samples improves from 69.00 to 69.74. This indicates that the proposed framework can effectively improve the performance of LLM-based CAMR parsers.

In addition, due to the presence of post-processing, corrections to format errors do not have a significant impact on the final results. Corrections for hallucination issues made by the agents can change some hallucinated concept nodes or remove nonexistent coreference relations, causing changes in the final output. As shown in Table 4, after correction, the recall of the samples increases from 68.49 to 70.98, and the F-score improves from 69.00 to 69.74, while precision decreases from 69.52 to 68.53. The results

	Total		Instance	Attribute	Relation	
	P	R	F ₁	F ₁	F ₁	
Before correction	69.52	68.49	69.00	76.31	88.64	50.27
After correction	68.53	70.98	69.74	77.91	87.97	50.99

Table 4: Performance of the multi-agent system

indicate that through the multi-agent system, some hallucinated concept nodes are correctly modified, however, there are also some correct nodes or relations mistakenly altered. Overall, the correction made by multi-agent system optimizes hallucination issues and improves the F₁. Some examples of corrections are provided in Figure 4 of Appendix C. For the first example, the agent for hallucination identification finds the hallucinated concept “x26 / 香”, and the agent for correction corrects it to proper concept “x26 / 馊”. For the second example, the agent for hallucination identification finds the hallucinated concept “x4 / 头脑”, and the agent for correction corrects it to proper concept “x4 / 埋头”. Please refer to Appendix C for more examples.

In summary, the multi-agent system can effectively mitigate hallucination phenomena present in CAMR sequences. However, since it may sometimes misjudge correct cases, there is still room for improving multi-agent system.

The rationale for low results on document-level coreference resolution: In the document-level coreference resolution task, due to limitations in context length, directly extracting entity coreference chains from long documents is too brute-force and yields suboptimal results. Both the validation set used for training and the test set for prediction contain some samples with extremely long contexts that exceed the maximum sequence length defined in the experimental setup, resulting in truncated sequences during actual training and inference.

4 Conclusion

In CAMRP2025, we assessed the capabilities of large language models released over the past year on the CAMR parsing task. Additionally, we explored the potential of multi-agent systems in detecting and correcting format issues and hallucinations in CAMR prediction sequences, achieving promising results.

In the future, for sentence-level CAMR parsing, we will address the limitations of sequence-to-sequence methods, where linearization increases distances between connected concept nodes, by optimizing graph prediction to reduce excessive edge distances. For document-level coreference resolution, we plan to process two sentences at a time to avoid truncation, then merge results across the full document, balancing context length and scalability.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China (62406091, 62276077, U23B2055), Shenzhen Science and Technology Program (KQTD2024072910215406, ZDSYS20230626091203008), and Shenzhen College Stability Support Plan under Grant GXWD20231130104007001.

References

- Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.
- Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022a. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Xuefeng Bai, Sen Yang, Leyang Cui, Linfeng Song, and Yue Zhang. 2022b. Cross-domain generalization for AMR parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

- 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*.
- Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Wenyang Gao, Xuefeng Bai, and Yue Zhang. 2023. System report for CCL23-eval task 2: WestlakeNLP, investigating generative large language models for Chinese AMR parsing. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*.
- Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam M. Nguyen, Dzung T. Phan, Vanessa López, and Ramón Fernández Astudillo. 2021. Ensembling graph predictions for AMR parsing. In *Annual Conference on Neural Information Processing Systems*.
- Ziyi Huang, Junhui Li, and Zhengxian Gong. 2021. Chinese AMR parsing based on sequence-to-sequence modeling. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*.
- Changmao Li and Jeffrey Flanigan. 2022. Improving neural machine translation with the Abstract Meaning Representation by combining graph and sequence transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing*.
- Bin Li, Yuan Wen, Weiguang Qu, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016*.
- Bin Li, Yuan Wen, Li Song, Weiguang Qu, and Nianwen Xue. 2019. Building a Chinese AMR bank with concept and relation alignments. *Linguistic Issues in Language Technology*.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Behrooz Mansouri. 2025. Survey of abstract meaning representation: Then, now, future. *ArXiv preprint*, abs/2505.03229.
- Long H. B. Nguyen, Viet H. Pham, and Dien Dinh. 2021. Improving neural machine translation with amr semantic graphs. *Mathematical Problems in Engineering*.
- 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*.
- Zihao Wu, Hua Yin, Ziqian Gao, Jijia Zhang, Yuelei Ji, and Kuntian Tang. 2024. System report for ccl24-eval task 4: Chinese amr parsing framework with mixture of lora experts. In *Proceedings of the 23rd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*.
- Liming Xiao, Bin Li, Zhixing Xu, Kairui Huo, Minxuan Feng, Junsheng Zhou, and Weiguang Qu. 2022. Align-smatch: A novel evaluation method for chinese abstract meaning representation parsing based on alignment of concept and relation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.
- Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Appendix A. The Prompts of the Agent for Error Detection and the Agent for Hallucination Identification

You are an error detection / a hallucination identification assistant for Chinese Abstract Meaning Representation (CAMR) sequences. Based on the input [sentence] and [CAMR], please check the validity of the CAMR sequence, and determine whether it contains any of the following issues. Point out each issue item by item. If there are no problems, please clearly state that the sequence is valid.

Input Format:

* [sentence]: A tokenized Chinese sentence, in the form “x1_词语1 x2_词语2 ...”

* [CAMR]: The CAMR sequence corresponding to the sentence (in a bracketed semantic structure similar to AMR)

CAMR Sequence Structure Information:

* CAMR sequences are evolved from English AMR and are used to represent abstract semantic relations extracted from Chinese sentences, though they are not exactly the same.

* In CAMR sequences, parentheses “(” and “)” are used to define concepts and their hierarchical relationships.

* Phrases beginning with “:” such as “:arg0” and “:arg1” are relation labels, indicating the type of relationship between concepts.

* The “/” symbol is used to define concept nodes, e.g., “x4 / 怀疑-01”. The left part “x4” is a unique variable identifying the concept, and the right part “怀疑-01” is the actual concept of the node.

Checklist for Validation:

Agent for error detection

1. Parenthesis Matching Errors:

* Check for missing or extra parentheses, unclosed structures, or incorrect closing order.

2. Structural and Formatting Errors:

* Missing concepts in nodes;

* Repeated or illegally nested relation labels, or labels in the wrong position;

* Malformed node structure, with invalid segments;

* Illegal spaces or incorrect concatenation in named entities;

* Clearly incomplete structures.

Agent for hallucination identification

1. Undefined Coreference Variables:

* All variables used for coreference must be defined elsewhere in the sequence;

* If a variable is undefined, the coreference relation should be deleted or the reference corrected.

2. Hallucinated Concepts:

* If a concept in Chinese of CAMR does not originate from [sentence], it is considered hallucinated;

* This includes named entities, abstract nodes, or structures not formed from any words in the sentence;

Special Notes:

* Please focus on the structural validity of the CAMR and its consistency with the input sentence;

* When checking for hallucinated concepts, English terms are often valid abstractions from the sentence—please pay closer attention to hallucinations involving Chinese concepts.

Output Format Requirements:

* You may include your reasoning process. After completing the checks, summarize any detected errors using the format below.

* If there are errors, list each issue clearly in the following format: [Detect]: Issue 1: ...; Issue 2: ...

* If no issues are found, output: [Detect]: The sequence is valid.

Valid Example:

Example1-Example5

Invalid Example:

Example1-Example5

Please refer to the above examples and instructions. Carefully analyze and check the input CAMR sequence for errors step by step, summarize your reasoning process, and output the results as [Detect]:

[sentence]: *Input sentence*

[CAMR]: *Input CAMR sequence*

Appendix B. The Prompt of the Agent for Correction

You are a correction assistant for Chinese Abstract Meaning Representation (CAMR) sequences. Based on the input [sentence], [CAMR], and [Detect], please correct errors or hallucinations in the CAMR sequence to ensure the sequence is valid.

Input Format:

* [sentence]: A tokenized Chinese sentence, in the form “x1_词语1 x2_词语2 ...”

* [CAMR]: The CAMR sequence corresponding to the sentence (in a bracketed semantic structure similar to AMR)

* [Detect]: Error information to be corrected (e.g., Issue 1: ... Issue 2: ...)

CAMR Sequence Structure Information:

* CAMR sequences are evolved from English AMR and are used to represent abstract semantic relations extracted from Chinese sentences, though they are not exactly the same.

* In CAMR sequences, parentheses “(” and “)” are used to define concepts and their hierarchical relationships.

* Phrases beginning with “:” such as “:arg0” and “:arg1” are relation labels, indicating the type of relationship between concepts.

* The “/” symbol is used to define concept nodes, e.g., “x4 / 怀疑-01”. The left part “x4” is a unique variable identifying the concept, and the right part “怀疑-01” is the actual concept of the node.

Special Notes:

* Issues in [Detect] may be false positives. Please carefully consider whether modification is necessary. Do not blindly follow the [Detect] suggestions.

* If an issue involves hallucinated English concepts such as “amr-unknown”, “and”, “mean”, “person”, “include-91”, etc., they are abstracted from the sentence., no correction is needed—this is considered normal in CAMR sequences.

Output Format Requirements:

* You may include your reasoning process. After correcting all errors, output the final valid CAMR sequence as follows: [correct CAMR]: Valid CAMR sequence.

Example:

Example1-Example5

Please refer to the above examples and instructions to carefully analyze and correct errors in the input CAMR sequence. After thorough reasoning step by step, output the final valid [correct CAMR]:

[sentence]: *Input sentence*

[CAMR]: *Input CAMR sequence*

[Detect]: *Error information from the other two agent*

Appendix C. Examples of the Multi-agent System

[sentence]: ...x20_这 x21_是 x22_哪 x23_位 x24_高官 x25_的 x26_馊 x27_主意 x28_啊...

[CAMR]: ...:op2 (x27 / 主意 :mod (x26 / 香) :poss (x24 / 高官 :ralign (x25 / 的) ...

[Detect]: Issue 1: The concept “(x26 / 香)” does not match the token “x26_馊” in the sentence and is hallucinated; it should be corrected to reflect the original token concept.

[Correct]: ...:op2 (x27 / 主意 :mod (x26 / 馊) :poss (x24 / 高官 :ralign (x25 / 的) ...

[sentence]: x1_她 x2_一如既往 x3_, x4_埋头 x5_工作 x6_到 x7_67 x8_岁 x9_离开 x10_人世...

[CAMR]: ... (x18 / and :op1 (x2 / 一如既往-01 :arg0 (x1 / 她)) :op2 (x5 / 工作-01 :manner (x4 / 头脑...)

[Detect]: Issue 1: The concept “头脑” in “:manner (x4 / 头脑...)” is hallucinated, as it does not originate from the sentence (“埋头” is in the sentence, but “头脑” is not); it should be revised to correctly represent “埋头”.

[Correct]: ... (x18 / and :op1 (x2 / 一如既往-01 :arg0 (x1 / 她)) :op2 (x5 / 工作-01 :manner (x4 / 埋头-01 ...

[sentence]: ...x9_面对 x10_杀人 x11_不 x12_眨眼 x13_的 x14_蒙古 x15_、 x16_匈奴 x17_敌军 x18_尚且...

[CAMR]: ... (x17 / 敌军 :mod (x56 / country :name (x14 / 蒙古)) :mod (x58 / ethnic-group :name (x16 / 匹奴)) ...

[Detect]: Issue 1: The CAMR concept “(x58 / ethnic-group :name (x16 / 匹奴))” contains the hallucinated Chinese concept “匹奴”, which does not exist in the input sentence. It should be corrected to “匈奴” to match the token x16.

[Correct]: ... (x17 / 敌军 :mod (x56 / country :name (x14 / 蒙古)) :mod (x58 / ethnic-group :name (x16 / 匈奴)) ...

[sentence]: x1_主席 x2_用 x3_食指 x4_按 x5_着 x6_紧 x7_锁 x8_的 x9_眉头 ...

[CAMR]: ...:op1 (x4 / 按-01 :aspect (x5 / 着) :arg0 (x1 / 主席) :arg1 (x9 / 虚目 :arg1-of (x7 / 锁-01 :ralign (x8 / 的) ...

[Detect]: Issue 1: The concept node “(x9 / 虚目)” is hallucinated because “虚目” is not present in the sentence; it should be corrected to “(x9 / 眉头)”.

[Correct]: ...:op1 (x4 / 按-01 :aspect (x5 / 着) :arg0 (x1 / 主席) :arg1 (x9 / 眉头 :arg1-of (x7 / 锁-01 :ralign (x8 / 的) ...

[sentence]: ...x21_玉帝 x22_却 x23_不 x24_是 x25_道教 x26_创始人 x27_.

[CAMR]: ...:arg2 (x26 / 创世人 :ralign (x22 / 却) :mod (x25 / 道教) :domain (x21 / 玉帝...

[Detect]: Issue 1: The Chinese concept “创世人” (x26) does not match the corresponding sentence token “创始人” (x26), indicating a hallucinated concept; it should be corrected to align exactly with the sentence token.

[Correct]: ...:arg2 (x26 / 创始人 :ralign (x22 / 却) :mod (x25 / 道教) :domain (x21 / 玉帝...

[sentence]: ...x12_里面 x13_含有 x14_大量 x15_的 x16_苯 x17_, x18_芳香烃 x19_类...

[CAMR]: ... (x20 / and :op1 (x16 / 苯 :quant (x14 / 大量 :ralign (x15 / 的))) :op2 (x19 / 类 :mod (x18 / 香料烃)) ...

[Detect]: Issue 1: The concept “(x18 / 香料烃)” in the CAMR is hallucinated or a typo, as the sentence contains “芳香烃” (aromatic hydrocarbon) at x18, but the CAMR uses “香料烃” instead. This should be corrected to match the sentence.

[Correct]: ... (x20 / and :op1 (x16 / 苯 :quant (x14 / 大量 :ralign (x15 / 的))) :op2 (x19 / 类 :mod (x18 / 芳香烃)) ...

[sentence]: ...x15_潘辉强 x16_勇 x17_斗 x18_持 x19_刀 x20_歹徒 x21_身负 x22_重伤...

[CAMR]: ...:cause (x17 / 斗-01 :manner (x16 / 勇) :arg1 (x20 / 歹徒 :arg0-of (x18 / 持-01 :arg1 (x19 / 刀))) ...

[Detect]: Issue 1: The concept “(x20 / 歹徒)” is a hallucinated or miswritten concept, as the sentence contains “歹徒” for x20. It should be corrected to “(x20 / 歹徒)”.

[Correct]: ...:cause (x17 / 斗-01 :manner (x16 / 勇) :arg1 (x20 / 歹徒 :arg0-of (x18 / 持-01 :arg1 (x19 / 刀))) ...

[sentence]: x1_一 x2_个 x3_巴掌 x4_拍 x5_不 x6_响 x7_, x8_很多 x9_事情 x10_没 x11_那么 x12_简单 x13_啊

[CAMR]: ...:arg1 (x4 / 拍-01 :arg1 (x3 / 掌巴 :quant (x1 / 1) :cunit (x2 / 个)) ...

[Detect]: Issue 1: The concept “(x3 / 掌巴)” is a hallucinated concept since “掌巴” does not exist in the sentence; the correct concept should be “(x3 / 巴掌)”.

[Correct]: ...:arg1 (x4 / 拍-01 :arg1 (x3 / 巴掌 :quant (x1 / 1) :cunit (x2 / 个)) ...

Figure 4: Examples of the multi-agent system