GPT4AMR: Does LLM-based Paraphrasing Improve AMR-to-text Generation Fluency?

Jiyuan Ji
Amherst College
cji28@amherst.edu

Shira Wein
Amherst College
swein@amherst.edu

Abstract

Abstract Meaning Representation (AMR) is a graph-based semantic representation that has been incorporated into numerous downstream tasks, in particular due to substantial efforts developing text-to-AMR parsing and AMR-to-text generation models. However, there still exists a large gap between fluent, natural sentences and texts generated from AMR-to-text generation models. Prompt-based Large Language Models (LLMs), on the other hand, have demonstrated an outstanding ability to produce fluent text in a variety of languages and domains. In this paper, we investigate the extent to which LLMs can improve the AMR-to-text generated output fluency post-hoc via prompt engineering. We conduct automatic and human evaluations of the results, and ultimately have mixed findings: LLM-generated paraphrases generally do not exhibit improvement in automatic evaluation, but outperform baseline texts according to our human evaluation. Thus, we provide a detailed error analysis of our results to investigate the complex nature of generating highly fluent text from semantic representations.

1 Introduction

Abstract Meaning Representation (AMR; Banarescu et al., 2013) is a graph-based semantic representation which captures the meaning of a phrase or sentence, with particular emphasis on semantic roles such as "who does what to whom."

The substantial efforts towards AMR-to-text generation (producing text from an AMR graph, see an example AMR graph and generated sentence in Figure 2) and text-to-AMR parsing (producing the graphs from the text) have enabled the AMR schema to be incorporated into a range of downstream tasks (Wein and Opitz, 2024).

Reference Text:
It's more comfortable to me.
Reference Graph:

```
(c / comfortable-02
   :ARG0 (i2 / it)
   :ARG1 (i / i)
   :degree (m / more))
```

Generated Sentence:
I'm more comfortable with it.

Figure 2: Example text generated by AMRBART from an AMR graph in AMR2.0 dataset. The reference text's AMR graph is in 'PENMAN' notation (Kasper, 1989).

Currently, AMR-to-text generation models can produce fairly fluent and adequate sentences that reflect the meaning of the graph. Still, the quality of the generated text from AMR-to-text generation models can be improved, both according to automatic metrics and human evaluation: state-of-the-art AMR-to-text generation models achieve approximately 50 BLEU points (Cheng et al., 2022; Bai et al., 2022) out of 100, and Manning et al. (2020) find that AMR-to-text generated output occasionally suffers from repetition of words or anonymization of low-frequency tokens.

In recent years, Large Language Models (LLMs) show the incredible ability to generate highly fluent text for a range of natural language processing tasks, such as machine translation and summarization. Therefore, in this work, we examine the ability of several prominent LLMs, including a reasoning model, to improve the fluency of AMR-to-text generation output. Specifically, **we investigate whether passing the output of an AMR-to-text generation model through a prompt-based LLM tasked with paraphrasing the text output can enable heightened fluency** (see Figure 1).

Paraphrases generally refer to varied expressions that convey the same meaning (Bhagat and Hovy, 2013). Here, we aim to preserve semantic meaning while improving fluency. We first generate texts from four state-of-the-art AMR-to-text generation

Proceedings of the 9th Widening NLP Workshop, pages 9–18

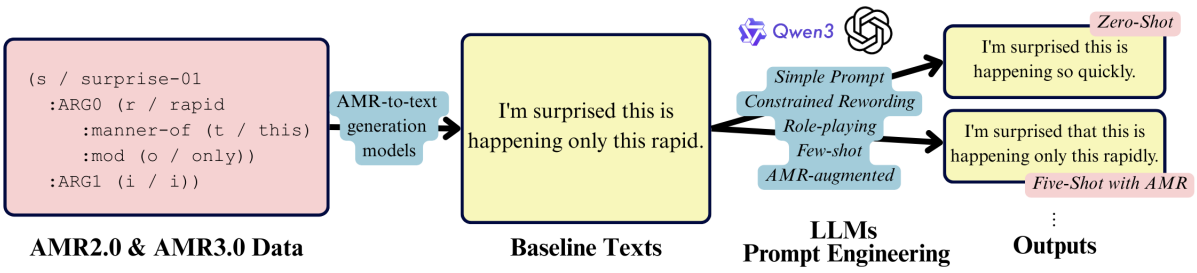

Figure 1: Experiment workflow, passing the original AMR data through AMR-to-text generation models, which results in our baseline texts. We then compare these baseline texts (via automatic metrics and human evaluations) to the texts output by the LLM prompt engineering.

models to serve as baselines. Then, we prompt the LLMs to output paraphrases for these texts through multiple prompting protocols. Finally, we compare the baseline texts and the LLM-generated paraphrases via four automatic metrics and a survey of human judgments. Our contributions include:

- Experimentation using prompt-based LLMs to increase the fluency of four AMR-to-text generation models post-hoc, including a variety of prompts across three LLMs.
- Automatic and human evaluations of our work, using four reference-based automatic metrics of 448 items and human judgments for both fluency and adequacy for 80 randomly selected items.
- A discussion and error analysis addressing our findings, as our prompts lead to mixed results.

2 Approach

In our experiments, we first pass AMR2.0 and AMR3.0 data into AMR-to-text generation models to generate baseline texts (§2.1). Then, we prompt LLMs (§2.2) to produce more fluent paraphrases of these texts through several prompting protocols (§2.3). Finally, we compare the results via automatic metrics and human evaluation (§2.4).

2.1 Data & Models

We use the AMR2.0 and AMR3.0 (Knight et al., 2017, 2020) test splits to generate texts to be passed into the LLMs for paraphrasing. AMR2.0 test data consists of 1,364 English sentences and their gold AMRs, while the AMR3.0 test data consists of 1,891 sentences and their gold AMRs, and collectively are made up of primarily newswire, web discussion forum, and fiction texts.

We use these gold AMRs as input to four state-of-the-art models: BiBL (Cheng et al., 2022), AMRBART (Bai et al., 2022), SPRING (Bevilacqua et al., 2021), and StructAdapt (Ribeiro et al., 2021). The output of these AMR-to-text generation models serve as the baseline, in order to ascertain whether the LLM-generated paraphrases are more fluent text by comparison.

2.2 Large Language Models

We prompt three LLMs: GPT-4o mini (OpenAI et al., 2024a), GPT-4.1 (OpenAI et al., 2024b), and Qwen3-14B (Yang et al., 2025). GPT-4o mini is a cost-efficient model that surpasses many small-sized models in textual processing. We first test all of the prompts with GPT-4o mini, then test the other models with the best-performing prompt. GPT-4.1 has strengths in instruction-following and complex tasks, while Qwen3-14B is an efficient reasoning model (especially for text generation). We enable Qwen3-14B's thinking mode and use the default values for all models.

2.3 Prompting Protocols

To task the LLMs with paraphrasing the AMR-to-text generated output, we develop several prompts. Every protocol is composed of the system prompt and the user prompt. We start by using a **simple prompt** that does not involve any examples, constraints, or role-playing.

Simple Prompt

System: You are an expert in paraphrasing.
User: Paraphrase the following sentence.
Sentence: <test_sentence>
Paraphrase:

As role-playing is shown to improve zero-shot performance (Kong et al., 2024), we then experiment with **two role-play prompts**. Given that the test sentences are from AMR-to-text generation

models, it may be helpful to let the LLM serve as an expert in editing such machine-generated text.

As the datasets largely consist of newswire and web posts, we also craft a prompt having the LLMs role-play an editor specialized in this domain.

Role-playing Machine-Generated Text Paraphrasing Expert (Zero-Shot RP1)

System: You are an expert paraphraser trained to edit machine-generated text.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions.
Sentence: <test_sentence>
Paraphrase:

LLMs may associate the words "paraphrase" or "rephrase" in the prompt with generating more diverse output, which may jeopardize meaning preservation. Thus, we experiment with a **constrained rewording extension of the role-playing prompts**. We instruct the model to avoid replacing words with their synonyms and instead improve sentences primarily via syntactic changes.

Constrained Rewording Extension of Role-Play Newswire Editor (Zero-shot RP2)

System: You are a professional English copyeditor specializing in both news articles and online discussion posts. Your primary goal is to improve sentence fluency only by restructuring sentences, changing their word order, or splitting and merging clauses as needed. Avoid replacing words with their synonyms.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions.
Sentence: <test_sentence>
Paraphrase:

Next, we experiment with few-shot prompting: **positive examples only** and both **positive and negative examples**. We select the examples from the texts generated by AMRBART on the AMR2.0 dataset. We choose positive examples at test time for five-shot prompting via either *sentence similarity* or *AMR similarity*. For sentence similarity, we obtain the top 5 similar sentences in the dataset to the test sentence based on chrf++ scores. For AMR similarity, we obtain the top 5 similar AMRs in the dataset to the test sentence's AMR based on the Smatch scores (Cai and Knight, 2013), then map these AMR graphs back to their corresponding sentences. The chosen sentences serve as positive example sentences, with their reference texts used as positive example paraphrases. We manually select the negative examples from the generated AMR2.0 texts from AMRBART that clearly do not preserve the reference text's meaning. We then manually write explanations on how it is a negative example (see Appendix A for an example).

Positive Examples with Role-Play & Constrained Rewording (Five-Shot Sent/AMR+RP1*)

System: You are an expert paraphraser trained to edit machine-generated text. Your primary goal is to improve sentence fluency only by restructuring sentences, changing their word order, or splitting and merging clauses as needed. Avoid replacing words with their synonyms.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions.
Sentence: <positive_example_sentence_1>
Paraphrase: <positive_example_paraphrase_1>

<more positive examples>

Sentence: <test_sentence>
Paraphrase:

Finally, we create **AMR-augmented prompts**. In addition to the example and test sentences in five-shot prompting, we include their respective AMR graphs in the user prompt.[1] The graphs are linearized and in text-based PENMAN notation.

2.4 Evaluation

We use BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2020), METEOR (Banerjee and Lavie, 2005), and chrf++ (Popović, 2017) to evaluate the baseline texts from the AMR-to-text generation models, and then the output paraphrases after prompting the LLMs.

We additionally conduct a human evaluation with four college students who are native English speakers. The survey has 20 questions and 80 judgments in total. For each question, we provide the reference sentence chosen randomly from the AMR2.0 dataset, and its four paraphrase candidates: (1) AMRBART-generated text (baseline), (2) zero-shot paraphrase from GPT-4o mini on baseline text, (3) paraphrase from GPT-4.1 on baseline text, and (4) paraphrase from Qwen3 on baseline text. The annotators are asked to evaluate fluency and ad-

[1] Since the LLMs may have seen the gold AMRs during pre-training, we use StructBart (Lee et al., 2022) to produce AMR graphs.

BERTScore												
		GPT-4o mini									GPT-4.1	Qwen3
		No AMR							AMR-augmented		AMR-augmented	
		Zero-Shot				Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	Simple	RP1	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**87.985**	76.385	78.829	80.393	81.223	85.345	85.080	85.911	86.023	86.417	86.911	86.478
SPRING	**86.050**	75.887	77.652	79.394	80.176	83.914	83.515	84.407	84.453	84.767	85.333	84.753
BiBL	**87.896**	76.493	78.467	80.571	81.206	85.409	85.110	85.884	85.968	86.292	86.826	86.397
StructAdapt	85.370	76.446	78.573	81.048	82.198	85.323	85.133	85.629	85.947	86.266	**86.620**	86.466

Table 1: BERTScore results on the AMR2.0 dataset. Baseline: AMR-to-text generation model results, Simple: simple prompt, RP1: role-play expert in editing machine-generated text, RP1*: RP1 with constrained rewording, RP2: role-play newswire editor with constrained rewording, Sent: positive examples chosen by sentence similarity, AMR: positive examples chosen by AMR graph similarity, Neg: both positive and negative examples.

Models	Fluency	Adequacy	Sum
Baseline	3.475	3.163	6.638
Zero-shot	**3.763**	3.175	**6.938**
GPT-4.1	3.382	**3.213**	6.594
Qwen3	3.447	3.038	6.485

Table 2: Human evaluation results on the four paraphrase candidates of the chosen sentences (Section 2.4).

equacy on a scale from 1 to 4 (instructions provided to the annotators are available in Appendix C). Fluency is judged first, without access to the reference, and then adequacy is judged with respect to the reference. All punctuation is normalized to ensure that the annotators do not unduly penalize text when they suspect it is machine-generated. [2]

3 Results

Table 1 presents the BERTScore results for GPT-4o mini on all the prompts applied to texts generated by each of the four AMR-to-text generation models.[3] We find that most LLM-generated texts score lower than the baseline, except for some minimal improvement in texts generated by StructAdapt.

The poor performance of the simple prompt via automatic metrics follows the results of prior research (Zhou et al., 2024). Without any given constraints, GPT-4o mini tends to output diverse results through synonym substitution, which may not preserve the original meaning, for example:

Generated text from SPRING:
Pledge to fight to defend the Diaoyu Islands and its related islands by death.
GPT-4o mini paraphrase:
Commit to defending the Diaoyu Islands and their associated territories with unwavering determination.

[2]Our experimentation cost approximately 70 USD.
[3]See Appendix D for more automatic evaluation results.

BERTScore appears to be the most resistant metric to synonym substitution. With the simple prompt, BLEU drops by approximately 60% and METEOR and chrf++ by approximately 40%, while BERTScore decreases by only 10%. This may be attributed to its reliance on word embedding similarity rather than exact word mapping.

Role-playing shows a substantial improvement, increasing zero-shot performance by approximately 30-40% for METEOR and chrf++ and 65-90% for BLEU compared to the simple prompt. The best zero-shot results come from prompting the model as a newswire copyeditor, confirming our conjecture that role-specific prompting triggers LLMs to draw upon their domain familiarity.

AMR-augmented prompting results in a mixed performance. BERTScore decreases slightly with zero-shot, while the rest show minor improvement. However, the improved performance may have resulted from LLMs extracting the reference text's exact words retained in AMR graphs, whereas the generation model might have substituted them with synonyms.

Test Sentence: *The youngest brother remains a tender youth.*
Qwen3 Paraphrase: *The youngest brother is still a tender youth.*

```
(y2 / youth
    :ARG1-of (t / tender-02)
    :domain (p / person
        :ARG0-of (h / have-rel-role-91
            :ARG2 (b / brother))
        :mod (y / young
            :degree (m / most)))
    :mod (s / still))
```

Thus, by referencing the AMR, LLMs generally produce sentences that are "better" in the sense that they more closely match the reference text. This is supported by the fact that BERTScore does not increase as much as BLEU when using AMR-

augmented prompting. Although paraphrases generated by GPT-4.1 and Qwen3 outperform those of GPT-4o mini's, they do not exceed the baseline.

Table 2 presents the human evaluation results. Surprisingly, the best-performing zero-shot prompt (i.e., role-playing newswire copyeditor with constrained rewording) attains the best fluency and overall scores, outperforming the baseline in this case. By conducting a paired t-test comparing the zero-shot and baseline scores, we find that the mean difference is statistically significant (the one-tailed p-value is 0.00955), which suggests that zero-shot prompting actually yields mixed results.

The preference for zero-shot prompting output in the human evaluation may be attributed to the use of more common phrases and prepositions, such as the baseline saying "athletes [...] competing under strong sunlight" versus "*in* strong sunlight."

4 Related Work

The rise of LLMs and the subsequent development of prompt engineering (Liu et al., 2021) have led to recent work prompting LLMs to generate text in a variety of domains, such as paraphrasing math problem to improve solve rates (Zhou et al., 2024) and to produce specific types of paraphrases following linguistic instructions (Vahtola et al., 2025). However, it has been noted that LLMs tend to provide overly complicated lexical expressions (Wu and Arase, 2025) and struggle to understand sentence structure (Vahtola et al., 2025) when paraphrasing, which presents a challenge for our approach.

Although in-context learning (ICL) prompting is common, work integrating AMR graphs has been sparse. One such study (Raut et al., 2025) discovers that AMR-augmented prompting may improve LLMs' performance in tasks involving long context, such as summarization, which suggests that AMRs may help with certain text generation tasks.

In regard to AMR-to-text generation, the output is mostly evaluated with automatic metrics, such as BLEU (Papineni et al., 2002), that compare the generated text with the human-annotated reference. However, it is unclear whether these metrics are suitable for assessing paraphrases, as they punish results with less n-gram overlap despite successful semantic preservation (Jin and Gildea, 2022). BERTScore (Zhang et al., 2020), on the other hand, relies on comparing contextual embeddings to more accurately reflect semantic similarity. In addition to automatic metrics, using human evaluation has been emphasized for a fuller analysis of AMR-to-text output (Manning et al., 2020).

5 Conclusion & Future Work

In this work, we explore the extent to which prompting LLMs to paraphrase can improve AMR-to-text generated output fluency, experimenting with variations of prompts such as constrained rewording, role-playing, and AMR augmentation. Our findings are mixed. Through automatic evaluation, we find that none of the prompts lead to better LLM-generated paraphrases compared to the baseline. Specifically, we reveal LLMs' tendency to relate paraphrasing to synonym substitution, which may result in meaning drift. We discover LLMs' sensitivity to prompt wording, especially when given rewording constraints. Few-shot and AMR-augmented prompting improve LLMs' performance in most cases, but this may have arisen from LLMs extracting the surface form instead of truly utilizing the semantic content of the AMR graphs. Human evaluation, on the other hand, shows that the best zero-shot prompt leads to a statistically significant increase in fluency. The higher ratings may be due to the fact that the zero-shot prompting has not been exposed to the rigid AMR-generated outputs and still has sufficient freedom to use more natural phrases and grammar. Additionally, applying role-play exhibits potential in aiding output fluency, given LLMs' massive training and thus the need to specify a trigger of specific domain knowledge. Our study highlights the complex nature of generating fluent text from a semantic representation that abstracts away from the surface form, as we find that leveraging a wide range of LLM prompts post-hoc to paraphrase the AMR-to-text generation system output generally does not improve performance.

Limitations

Our work is conducted using the AMR2.0 and AMR3.0 datasets (Knight et al., 2017, 2020), which consist primarily of broadcast scripts, newswire, and web discussion posts. Thus, it is unclear whether our results can be generalized to other domains of knowledge. Since domain-specific role-playing performs relatively better than other prompts in our study, future work might experiment with other role-play prompts with different datasets, such as *The Little Prince* (Banarescu et al., 2013). Future work may also investigate how other

models or syntactically controlled generation could be leveraged to improve AMR-to-text generation.

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. 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)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

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*.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

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.

Ziming Cheng, Zuchao Li, and Hai Zhao. 2022. BiBL: AMR parsing and generation with bidirectional Bayesian learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5461–5475, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Lisa Jin and Daniel Gildea. 2022. Rewarding semantic similarity under optimized alignments for AMR-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 710–715, Dublin, Ireland. Association for Computational Linguistics.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Kevin Knight, Bianca Badarau, Laura Banarescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2017. Abstract Meaning Representation (AMR) Annotation Release 2.0. Technical Report LDC2017T10, Linguistic Data Consortium, Philadelphia, PA.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and 1 others. 2020. Abstract Meaning Representation (AMR) Annotation Release 3.0. Technical Report LDC2020T02, Linguistic Data Consortium, Philadelphia, PA.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113, Mexico City, Mexico. Association for Computational Linguistics.

Young-Suk Lee, Ramón Astudillo, Hoang Thanh Lam, Tahira Naseem, Radu Florian, and Salim Roukos. 2022. Maximum Bayes Smatch ensemble distillation for AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5379–5392, Seattle, United States. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Preprint*, arXiv:2107.13586.

Emma Manning, Shira Wein, and Nathan Schneider. 2020. A human evaluation of AMR-to-English generation systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4773–4786, Barcelona, Spain (Online). International Committee on Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and

262 others. 2024b. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ankush Raut, Xiaofeng Zhu, and Maria Leonor Pacheco. 2025. Can llms interpret and leverage structured linguistic representations? a case study with amrs. *Preprint*, arXiv:2504.04745.

Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021. Structural adapters in pretrained language models for AMR-to-Text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4269–4282, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Teemu Vahtola, Songbo Hu, Mathias Creutz, Ivan Vulić, Anna Korhonen, and Jörg Tiedemann. 2025. Analyzing the effect of linguistic instructions on paraphrase generation. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 755–766, Tallinn, Estonia. University of Tartu Library.

Shira Wein and Juri Opitz. 2024. A survey of AMR applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6856–6875, Miami, Florida, USA. Association for Computational Linguistics.

Xuanxin Wu and Yuki Arase. 2025. An in-depth evaluation of large language models in sentence simplification with error-based human assessment. *Preprint*, arXiv:2403.04963.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yue Zhou, Yada Zhu, Diego Antognini, Yoon Kim, and Yang Zhang. 2024. Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2793–2804, Mexico City, Mexico. Association for Computational Linguistics.

A Demonstration of Negative Examples

Sentence: I'm just passing by.
Paraphrase: I just passed.
Explanation: The original sentence is present continuous, meaning the speaker is currently near the place, but the paraphrase is past tense, meaning the speaker is no longer near the place. Therefore, meaning is not preserved.

Figure 3: A demonstration of our negative example. Sentence: reference text, Paraphrase: text generated by AMRBART on a sentence from AMR2.0, Explanation: manually drafted to explain why the output fails to be a fluent paraphrase.

B Additional Prompt Templates

Constrained Rewording Extension of Role-playing Machine-Generated Text Paraphrasing Expert (Zero-Shot RP1*)

System: You are an expert paraphraser trained to edit machine-generated text. Your primary goal is to improve sentence fluency only by restructuring sentences, changing their word order, or splitting and merging clauses as needed. Avoid replacing words with their synonyms.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions.
Sentence: <test_sentence>
Paraphrase:

Positive & Negative Examples with Role-Play & Constrained Rewording (Five-Shot Neg+RP1*)

System: You are an expert paraphraser trained to edit machine-generated text. Your primary goal is to improve sentence fluency only by restructuring sentences, changing their word order, or splitting and merging clauses as needed. Avoid replacing words with their synonyms.

In the task, you will be shown positive and negative examples. Positive examples show correct paraphrasing that preserves meaning while improving fluency. Negative examples show incorrect paraphrases that change the meaning, use synonyms, or add/remove information. Produce output that matches the style and constraints of the positive examples and avoids the mistakes shown in the negative examples.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions.
Sentence: <positive_example_sentence_1>
Paraphrase: <positive_example_paraphrase_1>

<more positive examples>

Sentence: <negative_example_sentence_1>
Paraphrase: <negative_example_paraphrase_1>
Explanation: <negative_example_explanation_1>

<more negative examples>

Sentence: <test_sentence>
Paraphrase:

AMR-augmented Positive & Negative Examples with Constrained Rewording Extension of Role-playing (AMR-augmented Five-Shot Neg+RP1*)

System: You are an expert paraphraser trained to edit machine-generated text. Your primary goal is to improve sentence fluency only by restructuring sentences, changing their word order, or splitting and merging clauses as needed. Avoid replacing words with their synonyms.

In the task, you will be shown positive and negative examples. Positive examples show correct paraphrasing that preserves meaning while improving fluency. Negative examples show incorrect paraphrases that change the meaning, use synonyms, or add/remove information. Produce output that matches the style and constraints of the positive examples and avoids the mistakes shown in the negative examples.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions. You may use the provided linearized Abstract Meaning Representation (AMR) structure of the sentence to your aid.

Sentence: <positive_example_sentence_1>
AMR: <positive_example_amr_1>
Paraphrase: <positive_example_paraphrase_1>

<more positive examples>

Sentence: <negative_example_sentence_1>
AMR: <negative_example_amr_1>
Paraphrase: <negative_example_paraphrase_1>
Explanation: <negative_example_explanation_1>

<more negative examples>

Sentence: <test_sentence>
AMR: <test_sentence_amr>
Paraphrase:

AMR-augmented Positive & Negative Examples with Constrained Rewording Extension of Role-playing (AMR-augmented Five-Shot Neg+RP2)

System: You are a professional English copyeditor specializing in both news articles and online discussion posts. Your primary goal is to improve sentence fluency only by restructuring sentences, changing their word order, or splitting and merging clauses as needed. Avoid replacing words with their synonyms.

In the task, you will be shown positive and negative examples. Positive examples show correct paraphrasing that preserves meaning while improving fluency. Negative examples show incorrect paraphrases that change the meaning, use synonyms, or add/remove information. Produce outputs that match the style and constraints of the positive examples and avoid the mistakes shown in the negative examples.

User: Rephrase the following sentence to make it more fluent. Ensure the paraphrase conveys the same meaning, with no omissions or additions. You may use the provided linearized Abstract Meaning Representation (AMR) structure of the sentence to your aid.

Sentence: <positive_example_sentence_1>
AMR: <positive_example_amr_1>
Paraphrase: <positive_example_paraphrase_1>

<more positive examples>

Sentence: <negative_example_sentence_1>
AMR: <negative_example_amr_1>
Paraphrase: <negative_example_paraphrase_1>
Explanation: <negative_example_explanation_1>

<more negative examples>

Sentence: <test_sentence>
AMR: <test_sentence_amr>
Paraphrase:

C Human Evaluation Instruction

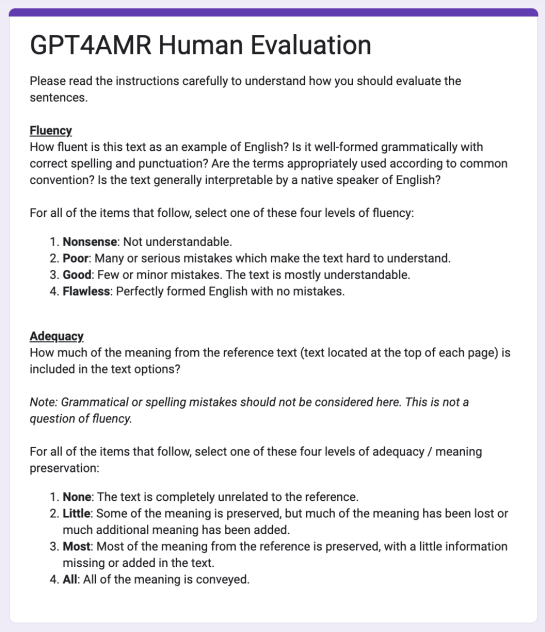

Figure 4: Human evaluation instructions that specify the scale of assessing fluency and adequacy.

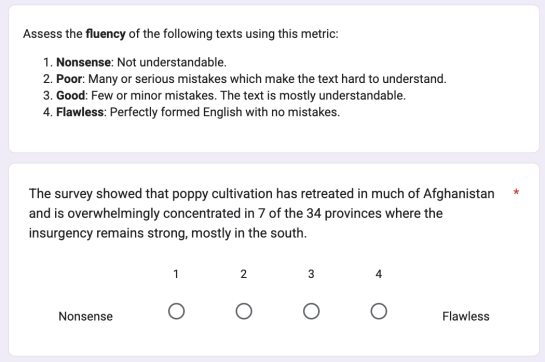

Figure 5: Instructions for evaluating sentence fluency and a sample question.

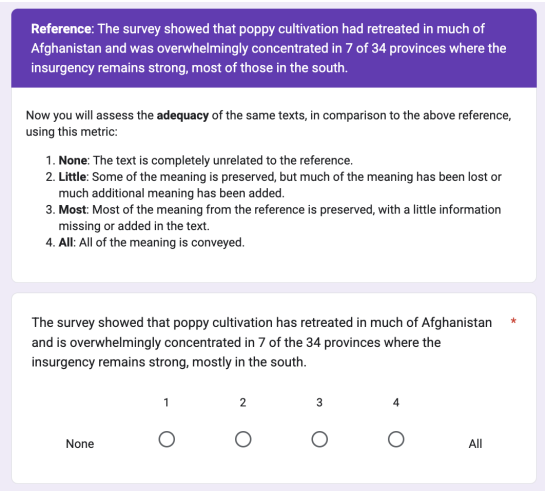

Figure 6: Instruction for evaluating sentence adequacy and a sample question.

D More Automatic Metrics Results

BLEU												
		GPT-4o mini									GPT-4.1	Qwen3
		No AMR							AMR-augmented		AMR-augmented	
		Zero-Shot				Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	Simple	RP1	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**48.236**	17.220	23.247	28.375	32.567	38.185	38.481	40.162	41.423	43.273	46.311	44.768
SPRING	**42.337**	16.630	21.657	26.148	29.667	34.759	34.672	36.884	36.963	39.077	41.154	39.732
BiBL	**47.997**	17.585	23.190	28.594	32.820	39.039	39.132	41.156	41.500	43.824	46.311	44.539
StructAdapt	45.181	17.350	23.185	28.727	32.372	37.973	37.797	39.866	40.905	42.783	**45.483**	44.056

Table 3: BLEU results on the AMR2.0 dataset.

METEOR												
		GPT-4o mini									GPT-4.1	Qwen3
		No AMR							AMR-augmented		AMR-augmented	
		Zero-Shot				Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	Simple	RP1	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**78.633**	47.098	55.322	62.390	66.712	71.824	71.747	73.885	74.928	75.832	76.928	76.319
SPRING	**74.932**	46.609	53.377	60.666	63.942	68.703	68.394	70.608	71.797	72.631	73.699	72.740
BiBL	**78.274**	47.746	55.034	62.863	66.296	72.334	71.803	73.881	75.064	75.868	76.702	76.160
StructAdapt	75.566	47.288	55.237	63.252	66.712	71.870	71.242	73.415	74.470	75.303	**76.377**	75.689

Table 4: METEOR results on the AMR2.0 dataset.

chrf++												
		GPT-4o mini									GPT-4.1	Qwen3
		No AMR							AMR-augmented		AMR-augmented	
		Zero-Shot				Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	Simple	RP1	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**73.209**	45.872	52.903	59.724	63.399	66.258	66.497	68.297	69.442	70.507	72.139	71.005
SPRING	**69.212**	45.110	51.555	57.825	60.976	63.388	63.362	65.392	66.171	67.232	68.649	67.425
BiBL	**73.205**	46.369	53.236	59.913	63.664	66.822	66.905	62.787	69.652	70.728	72.238	71.035
StructAdapt	**71.889**	46.126	51.942	59.955	63.273	66.187	66.111	68.010	69.214	70.255	71.764	70.769

Table 5: chrf++ results on the AMR2.0 dataset.

BERTScore										
		GPT-4o mini							GPT-4.1	Qwen3
		No AMR					AMR-augmented		AMR-augmented	
		Zero-Shot		Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**87.958**	80.899	81.689	85.628	85.362	86.011	86.024	86.518	86.876	86.829
SPRING	**86.187**	80.008	80.709	84.364	84.237	84.964	84.976	85.340	85.614	85.362
BiBL	**87.945**	81.052	81.764	85.741	85.386	86.232	86.388	86.693	86.990	86.667
StructAdapt	84.068	81.118	82.173	85.613	85.386	85.950	86.127	86.430	**86.810**	86.499

Table 6: BERTScore results on the AMR3.0 dataset.

BLEU										
		GPT-4o mini							GPT-4.1	Qwen3
		No AMR					AMR-augmented		AMR-augmented	
		Zero-Shot		Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**47.818**	28.682	32.808	38.172	37.807	40.281	40.911	42.936	45.698	44.181
SPRING	**41.809**	26.880	30.591	35.050	34.737	36.740	37.480	39.177	41.392	39.755
BiBL	**47.565**	29.408	33.258	38.665	38.460	40.695	41.733	43.661	45.856	44.007
StructAdapt	42.733	28.612	32.438	37.359	37.342	39.016	40.540	41.999	**44.707**	42.886

Table 7: BLEU results on the AMR3.0 dataset.

METEOR										
		GPT-4o mini							GPT-4.1	Qwen3
		No AMR					AMR-augmented		AMR-augmented	
		Zero-Shot		Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**77.146**	62.390	65.626	71.393	71.142	73.112	74.011	75.047	75.752	75.188
SPRING	**73.660**	60.240	63.498	68.920	68.773	70.773	71.523	72.422	73.102	72.272
BiBL	**76.957**	62.322	65.573	71.818	71.378	73.532	74.481	75.149	75.887	75.002
StructAdapt	71.347	61.834	65.291	71.306	70.754	72.934	73.669	74.608	**75.358**	73.828

Table 8: METEOR results on the AMR3.0 dataset.

chrf++										
		GPT-4o mini							GPT-4.1	Qwen3
		No AMR					AMR-augmented		AMR-augmented	
		Zero-Shot		Five-Shot			Five-Shot		Five-Shot	
Model / Prompt	Baseline	RP1*	RP2	Sent+RP1*	AMR+RP1*	Neg+RP1*	Neg+RP1*	Neg+RP2	Neg+RP2	
AMRBART	**72.415**	59.568	63.114	65.711	65.655	67.655	68.681	69.771	71.368	70.080
SPRING	**68.374**	57.528	61.008	62.982	62.902	64.768	69.092	70.247	68.155	67.019
BiBL	**72.409**	60.011	63.443	66.081	66.044	68.073	69.092	70.247	71.491	70.279
StructAdapt	70.510	59.345	62.656	65.221	65.206	67.057	68.362	69.417	**70.779**	69.566

Table 9: chrf++ results on the AMR3.0 dataset.