Analyzing the Role of Semantic Representations in the Era of Large Language Models

Zhijing Jin* MPI & ETH jinzhi@ethz.ch
Yuen Chen* UIUC yuenc2@illinois.edu
Fernando Gonzalez* ETH fer.adauto@gmail.com
Jiarui Liu CMU jiarui@cmu.edu
Jiayi Zhang University of Michigan jiayizzz@umich.edu
Julian Michael NYU julianjm@nyu.edu
Bernhard Schölkopf MPI bs@tue.mpg.de
Mona Diab CMU mdiab@cs.cmu.edu

Abstract

Traditionally, natural language processing (NLP) models often use a rich set of features created by linguistic expertise, such as semantic representations. However, in the era of large language models (LLMs), more and more tasks are turned into generic, end-to-end sequence generation problems. In this paper, we investigate the question: what is the role of semantic representations in the era of LLMs? Specifically, we investigate the effect of Abstract Meaning Representation (AMR) across five diverse NLP tasks. We propose an AMR-driven chain-of-thought prompting method, which we call AMRCoT, and find that it generally hurts performance more than it helps. To investigate what AMR may have to offer on these tasks, we conduct a series of analysis experiments. We find that it is difficult to predict which input examples AMR may help or hurt on, but errors tend to arise with multi-word expressions, named entities, and in the final inference step where the LLM must connect its reasoning over the AMR to its prediction. We recommend focusing on these areas for future work in semantic representations for LLMs.[1]

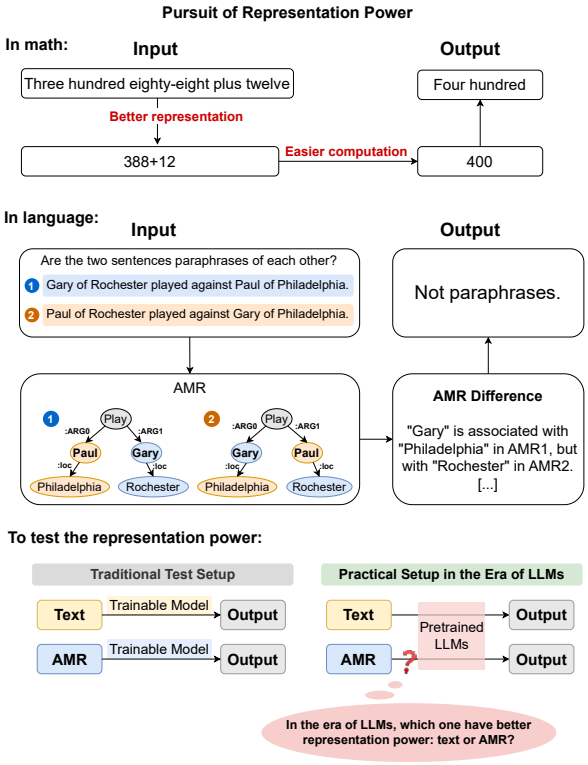

Figure 1: The role of representation power in different fields. Analogous to Arabic numbers for math, AMR is designed to efficiently and explicitly represent the semantic features of text. Existing work using AMR is concerned with trainable models, whereas we investigate the use of AMR in the modern practical setup of pre-trained LLMs.

1 Introduction

Formal representations of linguistic structure and meaning have long held an important role in the construction and evaluation of NLP systems. Semantic representations such as Abstract Meaning Representation (AMR; Banarescu et al., 2013) are designed to distill the semantic information of text to a graph consisting of the entities, events, and states mentioned in a text and the relations between them. Existing studies have shown the benefits of representations like AMR in a variety of NLP tasks, such as paraphrase detection (Issa et al., 2018), machine translation (Song et al., 2019), event extraction (Garg et al., 2015; Huang et al., 2018), code generation (Yin and Neubig, 2017), and others (Dohare and Karnick, 2017; Jangra et al., 2022; Wolfson et al., 2020; Kapanipathi et al., 2021). By explicitly representing the propositional structure of sentences, AMR removes much of the information from text that is irrelevant to semantic tasks, while surfacing the most important information (entities, relations, etc.), rendering them easier to operate on. In theory, this implies that using AMR as an intermediate representation should make it easier for a model to learn to perform such tasks, in the same way that a representation like Arabic numerals aids with arithmetic (see Figure 1).

*Equal contribution.
[1]Our code: https://github.com/causalNLP/amr_llm.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3781–3798
June 16-21, 2024

However, learning to produce and operate over representations like AMR is nontrivial, especially since AMR data is limited. In contrast, modern NLP systems based on large language models (LLMs) learn to directly manipulate text very effectively (Ignat et al., 2024), not only achieving high performance on a variety of tasks without using intermediate formal representations (Brown et al., 2020), but also achieving gains by directly using informal *textual* intermediates in methods such as chain-of-thought (CoT) prompting (Wei et al., 2022). Due to economic concerns (Zhao et al., 2022; Samsi et al., 2023; Patterson et al., 2021; Sharir et al., 2020), there is a growing trend to utilize readily available pre-trained LLMs in various application scenarios, without allocating additional resources for training or fine-tuning the models. These trends raise the question:

> *What is the role of semantic representations in the era of LLMs, when no training or finetuning is involved?*

Motivated by this question, we propose a theoretical formulation of representation power, and what it means to have an ideal representation for text, using ideas from Kolmogorov complexity (Solomonoff, 1964; Kolmogorov, 1965). Our key observation is that making use of even a very strong intermediate representation requires optimizing the model with regard to that representation; however, when using (out of the box) pretrained LLMs, the optimal representation will be the one which the LLM can *most effectively use* on the basis of its pretraining, which might shift away from the optimal representation for a learnable mapping to the output space. In short, the *a priori* ideal representation for a task is not necessarily the ideal representation for an LLM to use.

Given this, we empirically study how good AMR is as an intermediate representation for LLMs. Specifically, we answer the following three questions: (1) Does AMR help LLM performance? (2) When does AMR help, and when doesn't it? (3) Why does it help or not help in these cases?

On a diverse set of five NLP tasks, our experiments show that the contribution of AMR in the era of LLMs is not as great as that in the traditional setup where we can optimize the model for the representation. AMR causes a slight fluctuation of performance by -3 to +1 percentage points. However, we find that AMR is helpful for a subset of samples. We also find that the next step for using AMR to improve performance is likely not improving AMR parser performance, but improving the LLM's ability to map AMR representations into the output space.

In summary, the contributions of our work are as follows:

1. We are the first to investigate how semantic representations such as AMR can be leveraged to help LLM performance in the practical situation where no training is involved;
2. We propose a formalization of representation power for intermediate representations of language, and comprehensive experimental studies investigating *whether*, *when*, and *why* AMR can help when performing semantic tasks with LLMs;
3. We present thorough analysis experiments and results reflecting on the contribution of traditional linguistic structures such as semantic representations in the current wave of LLMs, and point out potential areas for improvement.

2 Formalizing Representation Power

In this section, we propose a framework to formulate representation power in both the pre-LLM era, where we do not outsource the training of the models, and the LLM era, where a lot of practical settings are to optimize the representation with regard to given fixed LLMs.

2.1 Notation

Suppose we have a dataset $D := \{(x_i, y_i)\}_{i=1}^{N}$ consisting of N pairs of input x_i and corresponding output y_i. Given the task to learn the $x \mapsto y$ mapping, we can consider it as a two-stage modeling process: the first step is to convert the raw input x into a good representation r by the representation model $g : x \mapsto r$, and the second step is to perform the computation that takes the representation r and predicts the output y by a computation model $h : r \mapsto y$. In this way, we decompose the resulting overall $x \mapsto y$ modeling process into

$$p(y|x) = p_g(r|x)p_h(y|r) \,, \tag{1}$$

where the overall probabilistic model $p(y|x)$ is turned into first the representation step $p_g(r|x)$ to generate the representation r given the input x, and

then the computation step $p_h(y|r)$ to perform operations on the intermediate representation r to derive the output y.

2.2 Problem Formulation

Let us draw some intuition from the math example in Figure 1. To represent numbers, the first representation choice r_1 is the English expression, such as "Three hundred eighty-eight plus twelve," and the second, intuitively stronger representation r_2 is the same calculation in Arabic numbers, "388+12". For our research question of whether AMR demonstrates stronger representation power than raw text, we formulate the question as follows:

- Representation choices: $\mathcal{R} = \{\text{text}, \text{amr}\}$ (which is an instance of text and its semantic representation, a common question of interest in linguistics);
- Representation power: some properties of the function $h : r \mapsto y$.

The next question becomes theorizing *what* properties of the computation model $h : r \mapsto y$ we are optimizing for, for which we will introduce two formulations, one in the pre-LLM era and the other in the LLM era.

2.3 Ideal Representations in the Pre-LLM Era

Continuing on with the Arabic number example: Our claim is that the representation r_2 (i.e., the Arabic number) is better than r_1 (i.e., the English expression) because the computation for $h_2 : r_2 \mapsto y$ is simpler than $h_1 : r_1 \mapsto y$, as measured by *Kolmogorov complexity*, or *algorithmic entropy* (Solomonoff, 1964; Kolmogorov, 1965), which is a theoretical construct of the complexity of an algorithm in bits.[2] Intuitively, the shortest program specifying an algorithm to take English expressions like "Three hundred eighty-eight plus twelve" as input and produce "Four hundred" as output should be longer than for the one taking "388+12" as input and "400" as output, since the former requires more complicated string manipulation to achieve the same effect.

We also use this notion of Kolmogorov complexity to quantify the power of representations for language. The intuition is that powerful representations are those that significantly simplify the complexity of the computation model h. Hence, the optimal representation function $g^* \in \mathcal{G}$ from the set $\mathcal{G}$ of possible functions should satisfy

$$g^* = \operatorname*{argmin}_{g \in \mathcal{G}} \min_{h \in \mathcal{H}} K(h) \,, \tag{2}$$

$$\text{where } h : g(x) \mapsto y \,, \tag{3}$$

and the optimal representation r^* is

$$r^* = g^*(x) \,. \tag{4}$$

Here, note that given each representation function, we optimize over all possible computation models h from the hypothesis space $\mathcal{H}$ to achieve the minimal Kolmogorov complexity $K(h)$.

If the representation enables the computation model h to have a low Kolmogorov complexity, it usually results in several good properties, such as that learning h has smaller generalization risks, requires fewer data samples, has smaller empirical risks, and results in more robustness, as introduced in Jin et al. (2021). Various studies explore the theoretical foundations for the above claims by connecting Kolmogorov complexity with statistical learning theory. For example, Vapnik (1995, §4.6.1) shows that an upper bound of Kolmogorov complexity, called "compression coefficient," can bound the generalization error in classification problems; Shalev-Shwartz and Ben-David (2014, Eq. 3) and Goldblum et al. (2023) show that generalization error is upper bounded by training error plus a term involving the Kolmogorov complexity of the hypothesis space.

This is, we argue, the implicit framework behind many previous studies showing AMR as a better representation than the raw text sequence by demonstrating its better performance (Turian et al., 2010), data efficiency (Liu et al., 2021), and robustness and domain transferability (Li et al., 2016; Jin et al., 2021). A crucial element of these studies is that they train models customized explicitly for the AMR representation, optimizing h over the hypothesis space $\mathcal{H}$.

2.4 Representation Power in the LLM Era

As mentioned previously, in the era of LLMs, we are moving towards the paradigm where the model training is usually outsourced, and during the inference stage, i.e., for most use cases, the model weights are fixed. Formally, this means two differences from the previous setting: (1) the hypothesis space $\mathcal{H}$ is collapsed to a size of one, containing

[2] Formally, Kolmogorov complexity is the length of the shortest program which produces a string.

only the fixed function h_{LLM}, (2) the optimization constraint in Eq. (3) that h can map the representation to the ground truth y is not necessarily guaranteed, namely that h could lead to $\hat{y}$, with certain estimation error.

Therefore, the key measure of representation power in the LLM era naturally shifts from simplicity of the computation model h—which aids optimization towards low estimation error—to low estimation error itself, i.e., $\mathbb{E}[\delta(\hat{y}, y)]$, where δ is the error function. This change results in a shift from the double optimization over both r and h to the optimization only of r with regard to the fixed h_{LLM}:

$$g^*_{\text{LLM}} = \underset{g \in \mathcal{G}}{\text{argmin}}\, \mathbb{E}[\delta(\hat{y}, y)]\,, \tag{5}$$

$$= \underset{g \in \mathcal{G}}{\text{argmin}}\, \mathbb{E}[\delta(h_{\text{LLM}}(g(x)), y)]\,, \tag{6}$$

where the optimal representation r^*_{LLM} becomes

$$r^*_{\text{LLM}} = g^*_{\text{LLM}}(x)\,. \tag{7}$$

This framework can also be used to explain the success, for example, of CoT prompting (Wei et al., 2022; Nye et al., 2021) in terms of how the intermediate representation generated by CoT better unlocks the power of LLMs.

Comparing Eqs. (2) to (4) with Eqs. (5) to (7), we can see that the ideal best representation r^* is not necessarily equal to the representation r^*_{LLM} that works well with LLMs, so there remains a need for experiments to fill in this knowledge gap.

It is also worth noting that for any learned representation function g, errors in $p_g(r|x)$ relative to $p_{g^*}(r|x)$ may cascade into the computation step $p(y|r)$, harming the final output. We investigate this concern in Section 6.1.

3 Designing the AMRCoT Experiments

We introduce an AMR-driven prompting method which we call AMRCoT, and investigate its performance on five datasets with five LLMs.

3.1 Dataset Setup

We test AMRCoT on paraphrase detection (Zhang et al., 2019), machine translation (Bojar et al., 2016), logical fallacy detection (Jin et al., 2022a), event extraction (Garg et al., 2015), and text-to-SQL generation (Yu et al., 2018). We select these tasks as they hinge on complex sentence structures and most of them are reported to have benefited from AMR in the pre-LLM era (Issa et al., 2018; Song et al., 2019; Garg et al., 2015; Yin and Neubig, 2017).

Dataset	Task	Test Size
PAWS	Paraphrase Detection	8,000
WMT16	Translation	5,999
Logic	Logical Fallacy Detection	2,449
Pubmed45	Event Extraction	5,000
SPIDER	Text2SQL Code Generation	8,034

Table 1: Tasks and datasets used.

For each dataset, we first take the entire original test set, and if it has fewer than 5,000 examples, we also include the development or training set. Data statistics are in Table 1 and details on test set construction are in Appendix A.1.

3.2 AMRCoT Prompt Design

To test the utility of AMR with LLMs, we draw inspiration from the CoT prompt design (Wei et al., 2022; Nye et al., 2021), together with CoT variants on causal (Jin et al., 2023) and moral reasoning tasks (Jin et al., 2022b), which enables models to answer an initially difficult question with the help of assistive intermediate steps to render the task easier.

We propose AMRCoT, in which we supplement the input text with an automatically-generated AMR and condition the LLM on the input text and AMR when generating the answer. If AMR has a stronger representation power than the raw text, then providing AMR as an assistive intermediate step should improve the performance of LLMs.

We compare AMRCoT to directly querying the LLMs, denoted BASE. An example prompt pair is shown in Table 2, and all prompts for all datasets

BASE	Please translate the following text from English to German. Text: `{sentence1}` Translation:
AMRCoT	You are given a text and its abstract meaning representation (AMR). Text: `{sentence1}` **AMR:** **`{amr1}`** Please translate the text from English to German. *You can refer to the provided AMR if it helps you in creating the translation.* Translation:

Table 2: Example BASE and AMRCoT prompt (for the translation task). We serialize AMRs with the commonly used Penman notation (Patten, 1993).

are in Appendix A.3.

3.3 Language Models

Since our experiments require models that can reasonably understand and reason over the symbols in AMRs, we find that only the instruction-tuned GPT models, from *text-davinci-001* to GPT-4 are capable of processing it, but not the open-sourced models such as LLaMa and Alpaca, at the time we conducted our research. For reproducibility, we set the text generation temperature to 0 for all models, and we use the model checkpoints from June 13, 2023 for GPT-3.5 and GPT-4, namely *gpt-3.5-turbo-0613* and *gpt-4-0613*.

3.4 Addressing Research Questions

4 Q1: Does AMR Help LLMs?

First, we are interested in the utility of AMR as an intermediate representation for LLMs. Specifically, we answer the following subquestions: what is the overall effect of AMR as a representation on LLMs' performance (Section 4.1)? Does the effect vary case by case (Section 4.2)? And how does the effect change with using various LLMs with different levels of capabilities (Section 4.3)?

4.1 Overall Effect of AMR

We first evaluate the overall effect of AMR as a representation to assist LLMs. Following the setup in Section 3, Table 3 shows performance on our five NLP tasks. Comparing AMRCoT to the BASE method which directly queries LLMs, AMR does not have an overall positive impact on performance. The performance fluctuates between a slight drop (-1 to -3 in most tasks) and a slight increase (+0.61 in the case of Text-to-SQL code generation).

Dataset	Task	BASE	ΔAMRCoT
PAWS	Paraphrase Detection	78.25	-3.04
WMT	Translation	27.52	-0.83
Logic	Fallacy Detection	55.61	-0.49
Pubmed45	Event Extraction	39.65	-3.87
SPIDER	Text2SQL	43.78	+0.61

Table 3: Across the five tasks, we report the baseline performance (BASE), and the additional impact of AMRCoT (ΔAMRCoT), using GPT-4. See statistical significance tests in Appendix D.1.

4.2 Helpfulness of AMR in Some Cases

Using AMR hardly changes overall performance, but this could be either because it does not change model predictions or because it helps in roughly as many cases as it hurts. To explore which is the case, we calculate the percentage of examples which are helped and hurt by AMRCoT, shown in Table 4. We count a sample as helped by AMR if its prediction improves (i.e., the output changes from incorrect to correct in classification tasks, or its score increases in text generation tasks), and hurt by AMR if its prediction degrades; the rest of the examples are considered unchanged.

Dataset	% Helped	% Hurt	% Unchanged
PAWS	16.48	20.16	63.36
WMT	16.45	21.17	62.38
Logic	1.96	2.45	95.59
Pubmed45	4.84	11.66	83.5
SPIDER	4.94	4.33	90.72

Table 4: Percentage of test samples that are helped (% Helped), hurt (% Hurt), or unchanged (% Unchanged) when we change from BASE to AMRCoT using GPT-4.

As shown in Table 4, AMR can change a significant proportion of answers, with 36.64% changed on PAWS, and 37.62% changed on WMT. On its face, the lack of overall improvement from AMR supports the current concern in the NLP community that traditional linguistics might have little role to play in improving the performance of NLP systems in the era of LLMs (Ignat et al., 2024). However, as there is a substantial subset of the data where AMR helps, if these improvements come from certain systematically identifiable subsets of the data, then this could provide clues for how structures such as AMR may potentially be leveraged to improve overall performance. We investigate this question further in Sections 5 and 6.

4.3 AMR's Effect on Models with Different Capabilities

Figure 2 shows the results of our experiments on models of varying capability, from text-davinci-001, -002, -003, to GPT-3.5 and GPT-4. Overall, AMRCoT hurts performance for most tasks and models, again with Text-to-SQL being the exception, at least for text-davinci-003 and GPT-4. In some cases, less capable models degrade more when using AMR, which might be due to their limited ability to comprehend AMR and reason over its special symbols. This is consistent with our preliminary observations that none of the non–instruction-tuned earlier GPT models, or the less capable models such as LLaMa and Alpaca, comprehend AMR or reason over them well.

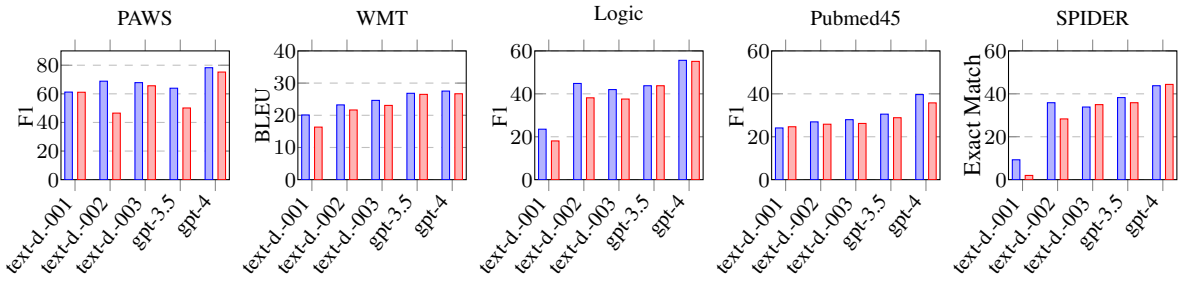

Figure 2: Performance of BASE (in purple) and AMRCOT (in red) on 5 datasets across 5 model versions: text-davinci-001|-002|-003, GPT-3.5 and GPT-4.

5 Q2: When Does AMR Help/Hurt?

The previous section shows that AMR is helpful or harmful for different samples. Now we investigate the conditions under which it helps or harms performance, in particular whether this can be predicted from features of the input text. We first illustrate a case study in Section 5.1, where AMR's lack of ability to capture the semantic equivalence of multi-word expressions (MWEs) hinders paraphrase detection. Then, we perform two systematic interpretability studies: First, we treat linguistic features as our hypotheses, and extract features with high correlation with AMR helpfulness (Section 5.2); second, we directly train classifiers to learn AMR helpfulness (Section 5.3).

5.1 Case Study: AMR's Shortcomings on MWEs

AMR has its unique advantages and limitations, from which we can interpret what cases it can help, and what cases not. One such limitation of AMR is its lack of ability to capture MWEs such as idiomatic expressions, which makes it overlook certain semantic equivalences for paraphrase detection. Consider the example in Figure 3. Here, the proper paraphrase for the MWE *swan song* is not "bird song," but "final performance." However, the AMRs for the three sentences do not reflect this; the AMR for the "swan song" sentence is structurally and lexically more similar to the "bird song" AMR than the one for the "final performance" variant.

Given this intuition, we qualitatively study whether AMR systematically fails on texts that contain more MWEs. We run AMRCOT on a self-composed dataset of paraphrase detection involving slang, assuming slang has more MWEs. Since our experiments need annotations for both slang paraphrase pairs and AMRs, we compose two datasets, GoldSlang-ComposedAMR, and GoldAMR-ComposedSlang. For GoldSlang-ComposedAMR, we use the curated slang paraphrase pairs by Tayyar Madabushi et al. (2021) and generate their AMRs with an off-the-shelf parser (Drozdov et al., 2022). For GoldAMR-ComposedSlang, we use gold AMRs from the LDC AMR 3.0 corpus (Banarescu et al., 2013), and compose slang paraphrases using a combination of manual annotation and assistance from GPT-4. The data curation steps and data statistics are in Appendix B.1.

Original Sentence with MWE
*Her **swan song** disappointed her fans.*
(z0 / disappoint-01
  :ARG0 (z1 / **song**
    :mod (z2 / **swan**)
    :poss (z3 / she))
  :ARG1 (z4 / fan
    :poss z3))

Paraphrase Candidate 1
(✗ Not a paraphrase.)

*Her **bird song** disappointed her fans.*
(z0 / disappoint-01
  :ARG0 (z1 / **song**
    :mod (z2 / **bird**)
    :poss (z3 / she))
  :ARG1 (z4 / fan
    :poss z3))

Paraphrase Candidate 2
(✓ A paraphrase.)

*Her **final performance** disappointed her fans.*
(z0 / disappoint-01
  :ARG0 (z1 / **perform**-02
    :ARG0 (z2 / she)
    :mod (z3 / **final**))
  :ARG1 (z4 / fan
    :poss z2))

Figure 3: An example showing the failure of AMR for paraphrase detection when the original sentence involves a MWE. This example is from our GoldSlang-ComposedAMR dataset.

Table 5 shows evaluation results, where AMRCOT causes a large drop in performance compared to BASE, more substantial than the slight fluctuation of -3 to +1 percentage points shown previously in

Dataset	BASE	ΔAMRCOT
GoldSlang-ComposedAMR	86.83	-6.63
GoldAMR-ComposedSlang	77.69	-8.78

Table 5: AMRCOT results in a large drop in performance on slang-comprising paraphrase detection data.

Top 5 Positive Features	Pearson Correlation
Adj POS Tag Frequency	0.0393
Avg. Word Complexity	0.0343
# Adjuncts	0.0337
Max Word Complexity	0.0316
Avg. Word Frequency	0.0271

Table 6: The top five features with the highest positive correlation coefficients to AMR helpfulness: the frequency of adjectives among all the words (*Adj* POS Tag Frequency), average word complexity level by the age of acquisition (Kuperman et al., 2012), number of adjuncts, maximum word complexity level by the age of acquisition, and average word frequency.

Top 5 Negative Features	Pearson Correlation
# Named Entities	-0.0630
% of Tokens Containing Digits	-0.0281
# Proper Nouns	-0.0258
# Third Person Singular Pronouns	-0.0236
# Quantifier Phrase Modifiers	-0.0222

Table 7: The top five features with the highest negative correlation coefficients to AMR helpfulness: the number of named entities, percentage of tokens containing digits, number of proper nouns (e.g., London), number of third person singular pronouns (e.g., he), and number of quantifier phrase modifiers. See detailed explanations of features in Appendix C.

Table 3. It is very likely that, due to the shortcomings of AMR on MWEs, AMRCoT mostly distracts the model, yielding worse performance.

5.2 Large-Scale Text Feature Analysis

The case study above provides a precise insight into a special case when AMR does not work. To systematically explore a larger set of hypotheses, we perform a feature analysis over the input texts. We formulate the contribution of AMR as the *AMR helpfulness* score, which is the per-example performance difference between AMRCoT and BASE, ranging between -100% and 100%, where a negative value means that AMR hurts performance on the example, and a positive value means that AMR improves performance.

For each input, we compute a comprehensive set of linguistic features, including 139 features on the text representation, and 4 features derived from the AMR. Specifically, we obtain 55 features using the Text Characterization Toolkit (TCT) (Simig et al., 2022), which is specifically designed to facilitate the analysis of text dataset properties, 17 different part-of-speech (POS) tags, 44 dependency tags, and 61 other hand-crafted features, which characterize the semantic and syntactic complexity of the input text, such as the number of arguments vs. adjuncts (Haspelmath, 2014).

Tables 6 and 7 show the Pearson correlation between each linguistic feature and the AMR helpfulness score. Overall, the correlation of each individual feature to the AMR helpfulness score is not strong, either because these features do not explain much about AMR helpfulness, or because it requires a combination of multiple features. Though the correlations are weak, the top correlated features in Table 6 align with our intuition that AMR should be helpful for semantically complex sentences: AMR is most helpful for samples with more adjectives, complex words, and adjuncts. In Table 7, the top negative feature, the number of named entities, echoes the finding in our previous MWE case study in Section 5.1, and we systematically show that AMR is most harmful on samples with many named entities, tokens containing digits, and proper nouns.

5.3 AMR Helpfulness Prediction as a Learning Task

Now we analyze the upper-bound predictability of AMR helpfulness from the input, both on the basis of our linguistic features and text input itself. Specifically, we train models to predict AMR helpfulness as a binary classification task where the positive class is the case where AMR helps, and the negative class is the rest. Merging all five datasets together, we have a binary classification dataset of 19,405 training samples, 4,267 development samples, and 5,766 test samples, with positive labels composing 10.38% of the dataset.

As shown in Table 8, classifiers based on linguistic features achieve an F1 score of up to 32.67%. BERT-based deep learning models improve by up to 1.16 F1 scores, with substantial increases in recall. For interpretability, we run Shapley feature attribution method (Fryer et al., 2021) and find that

Model	F1	Acc	P	R
Random Baseline	16.14	49.95	9.65	49.16
Using Linguistic Features				
Random Forest	**32.67**	81.93	25.72	44.75
XGBoost	30.08	78.47	22.06	47.27
Ensemble	30.42	77.59	21.85	50.00
Using the Free-Form Text Input				
BERT	**33.83**	79.70	25.00	52.28
RoBERTa	33.29	80.36	25.11	49.38

Table 8: Classification performance of various models on AMR helpfulness. We report the F1, precision (P), and recall (R) of the positive class, as well as the accuracy (Acc). See the implementation details of the models in Appendix A.5.

words that signal the existence of clauses tend to have high importance for the classifier, such as "what," "how," "said," and "says." These results do not provide a clear explanation of when AMR can help, but give a starting point, and we welcome future research to continue exploring the potential benefits of AMR. The fact that AMR helpfulness is challenging to predict even for BERT models may indicate either that we need more data to learn the features that predict this, or that a substantial portion of the changes that AMR makes to model predictions correspond to noise (i.e., help or hurt in unpredictable ways).

6 Q3: Why Does AMR Help/Hurt?

To understand why AMR helps or hurts when it does, we look into the following subquestions: (1) how does parser-generated AMR work compared with gold AMR (Section 6.1)? (2) what is the representation power of AMR versus text when the other is ablated (Section 6.2)? And (3) how does AMR help in each step of the reasoning process (Section 6.3)?

6.1 Gold vs. Parser-Generated AMR

First, we investigate whether there are cascading errors before the CoT process, due to mistakes in the parser-generated AMR. For example, the reported performance of Drozdov et al. (2022) is 83% on AMR 3.0 (Banarescu et al., 2013). To assess this, we compare AMRCoT performance when using predicted versus gold AMRs. Testing this requires data with gold AMR annotations as well as gold labels for some downstream NLP task we can evaluate the models on. To this end, we take the intersection of the AMR 3.0 dataset (Banarescu et al., 2013) with Ontonotes 5.0 (Pradhan et al., 2011), which contains 131 sentences that have both gold AMR and named entity recognition (NER) annotations. We list the intuition of why AMR can be helpful for NER in Appendix B.2.

Using this AMR-NER dataset, we compare the performance of AMRCoT with gold AMR versus parser-generated AMR on NER, shown in Table 9. Both lead to similar results, with a difference of less than two percentage points (which is not statistically significant, with $p = 0.627$ by t-test). The test set is unfortunately too small to reliably detect an effect of reasonable size, due to the lack of available data with both gold AMR and NLP task annotations; this result is also specific to NER, which may not have all of the relevant features for understanding the effect of gold versus automatically produced AMRs. However, the fact that the observed effect size is very small constitutes some evidence that improving the predicted AMRs would likely not play a huge role in increasing downstream performance with current models.

Dataset	BASE	AMR	ΔAMRCoT
AMR-NER	60.51	Gold	+0.03
		Parser	+1.91

Table 9: Model performance on the AMR-NER data using the gold AMR (Gold) and parser-generated AMR (Parser). We report the BASE performance, and the change of performance by AMRCoT (ΔAMRCoT) in terms of F1 scores.

6.2 Ablating the AMR/Text Representation

As discussed in Section 2, AMR and text representations are two different surface forms for expressing sentence semantics, but one representation may be more useful to the LLM than the other. To test this, we conduct an ablation study removing either the original text or the AMR and measuring performance (see Appendix A.6). To avoid the potential for cascading errors from the parsing process, we use the AMR-NER dataset with the gold AMR.

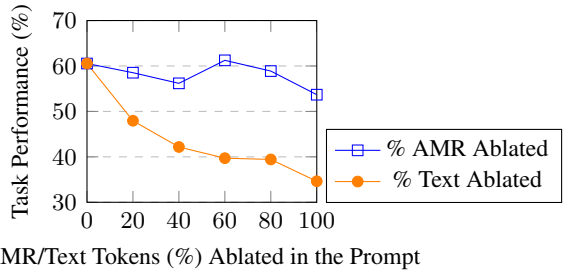

Figure 4: Ablation studies of AMR and text representations in the prompt on the AMR-NER dataset using GPT-4. Starting from the AMRCoT prompt with the complete text and AMR, we randomly drop out a certain portion of tokens in the text/AMR, and see the effect on the task performance.

Results In addition to previous results contrasting AMRCoT, which provides both the text and AMR in the prompt, and BASE with the text-only input, we show the results of a more granular analysis in Figure 4, where we randomly drop out text and AMR tokens and measure the effect on task performance. Similar to the above, we find that dropping AMR marginally decreases performance, and dropping text much more drastically degrades LLM performance, showing the greater utility of text as a representation for LLMs. We also conduct

the same ablation study on 1,000 random samples from the WMT dataset using predicted AMRs in Appendix D.3, where the observations are similar.

6.3 Checking the Step-By-Step Reasoning

To better understand how LLMs use AMR, we directly examine the step-by-step reasoning process produced by AMRCoT with GPT-4. We randomly select 50 samples from the PAWS dataset and manually annotate the correctness of each step in the reasoning process. For paraphrasing on PAWS, the steps (and our evaluations) are as follows:

1. Produce the AMR for the input sentences using Drozdov et al. (2022)'s structured BART model. Instead of manually annotating correctness of these AMRs, we defer to their reported performance of 82.6 SMATCH scores on the AMR 3.0 dataset.
2. Provide the AMRs to GPT-4 in the paraphrasing task prompt using AMRCoT, and then instruct it to list all the commonalities and differences of the AMRs. Our manual check finds that GPT-4 achieves a 97% F1 score (with 95% precision, 98% recall) at listing these.
3. GPT-4 then outputs a final decision on whether the sentences are paraphrases. We evaluate that its judgment in this step is consistent with the reasoning in the prior step 80% of the time.

Even though GPT-4 was able to correctly enumerate the relevant features of the AMRs, it still had trouble synthesizing this information into a correct paraphrasing judgment. These mistakes as well as the potential for cascading errors may explain why AMRCoT achieves a performance of 75.21% on PAWS, which is a slight drop from the BASE performance of 78.25%. Overall, this provides further evidence of the advantages that free-form text itself has as a representation for LLMs to operate on.

7 Related Work

Semantic Representations Traditionally, NLP models often represent text by features developed on the basis of linguistic expertise, among which semantic representations such as AMR (Banarescu et al., 2013) are used to abstract away the surface form of the text and distill the most important elements of its meaning. In the past, such representations have helped with a variety of NLP tasks, such as semantic parsing (Kuhn and De Mori, 1995), machine translation (Wu and Fung, 2009; Wong and Mooney, 2006), and text summarization (Liu et al., 2018). Recent research also looks into whether LLMs already incorporate a good understanding of semantic representations (e.g., Staliūnaitė and Iacobacci, 2020; Blevins et al., 2023).

Chain-of-Thought Prompting Rapid advancement in LLMs has led to a new paradigm of performing NLP tasks by eliciting model behavior via instructions and examples using prompting (Brown et al., 2020; Raffel et al., 2020). Chain-of-thought (CoT) prompting (Wei et al., 2022; Nye et al., 2021), which pairs input examples with step-by-step explanations of how to produce their respective outputs, has been shown to improve LLMs' performance at various reasoning tasks (Yu et al., 2023), and its variants have also shown success in various scenarios, such as CAUSALCOT for causal reasoning (Jin et al., 2023), and MORALCOT for moral reasoning (Jin et al., 2022b). Our work proposes a way to bridge linguistic representations with text by AMRCoT, providing an AMR as an intermediate representation for the LLM to reason over. Our results are mixed, demonstrating the relative advantage that unstructured, free-text representations have for language models pretrained on large amounts of natural language data.

8 Conclusion

In this work, we analyze the role of semantic representations in the era of LLMs. In response to the ongoing paradigm shift in the NLP community, we show that AMR in general is not yet a representation immediately fit for pre-trained LLMs. However, our study shows that AMR still helps on some samples. We also suggest that a potential direction to enhance AMR's contribution to LLMs is to improve the understanding of LLMs over the schemes and symbols of AMR, and map it to the reasoning of the respective NLP task. This work presents an effort to bridge the traditionally rich linguistic structures with the strength of LLMs.

Limitations and Future Work

This work explores one form of linguistic representation of text. In the future, we welcome more exploration on various other linguistic representations using the methodology presented in this work. Moreover, we explore one intuitive way of prompting the model. Future work is welcome to

explore different ways of prompting to make the AMR information more accessible and useful to the model.

In addition, some of our analyses are limited by a lack of annotated resources, so we were only able to show experimental results on hundreds of examples in some cases where gold AMR annotation is needed. This is a commonly known issue for AMR, which is expensive and requires a high level of linguistic expertise to annotate. This limitation makes the results less statistically significant than what we could have if there are more annotated AMRs available. In this work, we hope to strike a balance to still show some meaningful trends while trying to get the largest size of annotated data we can.

Moreover, if any future work has the resources to train an LLM specifically optimized for AMR as a representation, this would be the ideal setting to check out the upper bound of the power of AMR in the era of LLMs.

As for the limitations for specific parts of the paper, for example for the notion of gold AMRs in Section 6, although we use the AMR annotated by humans in the official Banarescu et al. (2013) dataset, it should be noted that such AMRs are not necessarily "perfect", as humans might also have a non-perfect inter-annotator agreement over some AMRs. And while SMATCH scores can be predictive, they may not perfectly reflect the quality of parser-generated AMRs (Opitz and Frank, 2022). These are both open research questions, and we use the AMRs released by the official source (Banarescu et al., 2013) as a proxy for ground-truth AMR.

Ethical Considerations

The datasets used in this paper are existing public datasets on general NLP tasks without any user-sensitive information. We are not aware of specific ethical concerns with the analysis in this study, which is a neutral investigation to understand the role of traditional linguistic structures such as semantic representations in the era of LLMs.

Acknowledgments

We thank Juri Opitz for his insightful suggestions on our AMR experiments based on profound domain expertise. We also appreciate Wendong Liang for insightful discussions on Komolgorov complexity, which is a foundation of the theoretical framework in this work. We thank Nils Heil for extracting the SQL schemes of the SPIDER dataset so that we can incorporate them in the prompt to improve our performance. This material is based in part upon works supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B; and by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645; Zhijing Jin is supported by PhD fellowships from the Future of Life Institute and Open Philanthropy.

Author Contributions

Mona Diab initiated the project idea based on her strong expertise in traditional linguistics, and an intuition that the semantic representations should help model efficiency, robustness, and interpretability. During the course of exploration by Zhijing Jin and Mona Diab together for over a year, they find that the AMR representations does not always help LLMs over multiple experimental setups and model implementations.

Zhijing further explores the theoretical formulation of representation power to provide the explanations behind the observed performance, together with the expertise of Bernhard Schoelkopf in causal representation learning. Julian Michael provided valuable insights and overview of the field of semantic representations, which brings the depth of the project to another level. Julian also provided constructive suggestions for improving the experiments and structuring the paper, and substantially improved the writing.

Yuen Chen and Fernando Gonzalez contributed substantially to scaling up all the experiments across multiple datasets and multiple model versions, and analyzing the results. Jiarui Liu and Jiayi Zhang helped with the training the BERT-based classifiers, and analyzing the Shapley values. Jiarui Liu conducted several important experiments for the camera-ready version of the paper, especially on checking the ceiling performance of AMRCoT with various prompt improvements and data setups.

References

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, LAW-ID@ACL 2013, August 8-9, 2013, Sofia, Bulgaria*, pages 178–186. The Association for Computer Linguistics. 1, 6, 8, 9, 10, 16

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics. 9

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics. 4

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2, 9

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM. 14

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee. 17

Shibhansh Dohare and Harish Karnick. 2017. Text summarization using abstract meaning representation. 1

Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramón Astudillo. 2022. Inducing and using alignments for transition-based AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1086–1098, Seattle, United States. Association for Computational Linguistics. 6, 8, 9, 16

Daniel Vidali Fryer, Inga Strümke, and Hien D. Nguyen. 2021. Shapley values for feature selection: The good, the bad, and the axioms. *CoRR*, abs/2102.10936. 7

Sahil Garg, A. G. Galstyan, Ulf Hermjakob, and Daniel Marcu. 2015. Extracting biomolecular interactions using semantic parsing of biomedical text. *ArXiv*, abs/1512.01587. 1, 4, 16

Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. 2023. The no free lunch theorem, Kolmogorov complexity, and the role of inductive biases in machine learning. *CoRR*, abs/2304.05366. 3

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538. 17

Martin Haspelmath. 2014. Arguments and adjuncts as language-particular syntactic categories and as comparative concepts. *Linguistic Discovery*, 12:3–11. 7

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics. 1, 16

Oana Ignat, Zhijing Jin, Artem Abzaliev, Laura Biester, Santiago Castro, Naihao Deng, Xinyi Gao, Aylin Gunal, Jacky He, Ashkan Kazemi, Muhammad Khalifa, Namho Koh, Andrew Lee, Siyang Liu, Do June Min, Shinka Mori, Joan Nwatu, Verónica Pérez-Rosas, Siqi Shen, Zekun Wang, Winston Wu, and Rada Mihalcea. 2024. Has it all been solved? Open nlp research questions not solved by large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. International Committee on Computational Linguistics. 2, 5

Fuad Issa, Marco Damonte, Shay B. Cohen, Xiaohui Yan, and Yi Chang. 2018. Abstract meaning representation for paraphrase detection. In *North American Chapter of the Association for Computational Linguistics*. 1, 4

Anubhav Jangra, Preksha Nema, and Aravindan Raghuveer. 2022. T-STAR: Truthful style transfer using AMR graph as intermediate representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8805–8825, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 1

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez, Max Kleiman-Weiner, Mrinmaya Sachan, and

Bernhard Schölkopf. 2023. CLadder: Assessing causal reasoning in language models. In *NeurIPS*. 4, 9

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. 2022a. Logical fallacy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7180–7198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 4

Zhijing Jin, Sydney Levine, Fernando Gonzalez, Ojasv Kamal, Maarten Sap, Mrinmaya Sachan, Rada Mihalcea, Josh Tenenbaum, and Bernhard Schölkopf. 2022b. When to make exceptions: Exploring language models as accounts of human moral judgment. In *NeurIPS*. 4, 9

Zhijing Jin, Julius von Kügelgen, Jingwei Ni, Tejas Vaidhya, Ayush Kaushal, Mrinmaya Sachan, and Bernhard Schölkopf. 2021. Causal direction of data collection matters: Implications of causal and anticausal learning for NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9499–9513, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 3

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics. 1

Andrei N Kolmogorov. 1965. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7. 2, 3

R. Kuhn and R. De Mori. 1995. The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):449–460. 9

Victor Kuperman, Hans Stadthagen-González, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44:978–990. 7, 17

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2016. Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1979–1985, Austin, Texas. Association for Computational Linguistics. 3

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman M. Sadeh, and Noah A. Smith. 2018. Toward abstractive summarization using semantic representations. *CoRR*, abs/1805.10399. 9

Zhiyuan Liu, Yankai Lin, and Maosong Sun. 2021. Representation learning for natural language processing. *CoRR*, abs/2102.03732. 3

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114. 4, 9

Juri Opitz and Anette Frank. 2022. Better Smatch = better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 32–43, Online. Association for Computational Linguistics. 10

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. 14, 17

Terry Patten. 1993. Book reviews: Text generation and systemic-functional linguistics: Experiences from English and Japanese. *Computational Linguistics*, 19(1). 4

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. 2

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics. 8

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67. 9

Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. 2023. From Words to Watts: Benchmarking the Energy Costs of Large Language Model Inference. 2

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press. 3

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training nlp models: A concise overview. 2

Daniel Simig, Tianlu Wang, Verna Dankers, Peter Henderson, Khuyagbaatar Batsuren, Dieuwke Hupkes, and Mona Diab. 2022. Text characterization toolkit (TCT). In *Proceedings of the 2nd Conference of the Asia-Pacific*

Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations, pages 72–87, Taipei, Taiwan. Association for Computational Linguistics. 7

Ray J Solomonoff. 1964. A formal theory of inductive inference. part ii. *Information and control*, 7(2):224–254. 2, 3

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. 1, 4

Ieva Staliūnaitė and Ignacio Iacobacci. 2020. Compositional and lexical semantics in RoBERTa, BERT and DistilBERT: A case study on CoQA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7046–7056, Online. Association for Computational Linguistics. 9

Student. 1908. The probable error of a mean. *Biometrika*, 6(1):1–25. 17

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic. Association for Computational Linguistics. 6, 16

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics. 3

Vladimir Naumovich Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer. 3

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*. 2, 4, 9

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *CoRR*, abs/2001.11770. 1

Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York City, USA. Association for Computational Linguistics. 9

Dekai Wu and Pascale Fung. 2009. Semantic roles for SMT: A hybrid two-pass model. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA, Short Papers*, pages 13–16. The Association for Computational Linguistics. 9

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada. Association for Computational Linguistics. 1, 4

Ping Yu, Tianlu Wang, Olga Golovneva, Badr AlKhamissi, Siddharth Verma, Zhijing Jin, Gargi Ghosh, Mona Diab, and Asli Celikyilmaz. 2023. ALERT: adapting language models to reasoning tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada. Association for Computational Linguistics. 9

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics. 4

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. 17

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. 4, 16

Dan Zhao, Nathan C. Frey, Joseph McDonald, Matthew Hubbell, David Bestor, Michael Jones, Andrew Prout, Vijay Gadepally, and Siddharth Samsi. 2022. A green(er) world for a.i. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 742–750. 2

A Experimental Details

A.1 Data Split Details

For the five datasets that we use in Section 5, we compose the test sets in the following way. To make the experimental results in our paper representative, we aim at composing a large test set for each task, ideally more than a size of 5,000 test samples. We sequentially check whether the test set is large enough, and if not, then we include the development set and the training set sequentially. Note that since our experiments are zero-shot, i.e., we do not train our models at all, any of the original test, development, or training sets can be used to report the performance on.

As a result, for the PAWS dataset, we use its entire test set, which is large enough with 8,000 samples. For WMT16, since its test set has only 2,999 samples, we also include its development set with 3,000 samples, totaling 5,999 samples for our test. For the LOGIC dataset, as it is a relatively small dataset, we add up all its original test, development, or training sets to obtain 2,449 samples. For Pubmed45, which contains 25,360 unsplit samples, we randomly select 5,000 data points for our analysis. For SPIDER, as its test set is not released, and development set has only 1,034 samples, we also include its training set of 7,000 samples, totaling 8,034 samples for our test.

A.2 Evaluation Metrics

For evaluation, we report the performance of PAWS, Logic, and Pubmed45 by F1 scores, the performance of machine translation on the WMT16 dataset by BLEU scores (Papineni et al., 2002), and the performance of text-to-SQL generation using the official evaluation setup at `https://github.com/taoyds/test-suite-sql-eval`. To evaluate the generation quality of parser-produced AMRs, we report the SMATCH scores using the evaluation codes at `https://github.com/snowblink14/smatch`.

A.3 Prompts

We list the prompts for BASE and AMRCOT of all datasets in Tables 10 and 11, as well as the system prompts in Table 12.

Paraphrase Detection (PAWS)	
BASE	Paraphrase Detection: Determine if the following two sentences are exact paraphrases (rewritten versions with the same meaning) of each other. Sentence 1: `{sentence1}` Sentence 2: `{sentence2}` Answer [Yes/No] and then provide a brief explanation of why you think the sentences are paraphrases or not. Paraphrase:

AMRCOT	Paraphrase Detection: You are given two sentences and the abstract meaning representation (AMR) of each.
	Sentence 1: `{sentence1}`
	AMR 1:
	`{amr1}`
	Sentence 2: `{sentence2}`
	AMR 2:
	`{amr2}`
	Explain what are the commonalities and differences between the two AMRs. Then determine if the two sentences are exact paraphrases (rewritten versions with the same meaning) of each other and provide a brief explanation of why you think the sentences are paraphrases or not.
	Use the following format: Answer: [Yes/No]

Translation (WMT16)	
BASE	Please translate the following text from English to German. Text: `{sentence1}` Translation:
AMRCOT	You are given a text and its abstract meaning representation (AMR). Text: `{sentence1}` AMR: `{amr1}` Please translate the text from English to German. You can refer to the provided AMR if it helps you in creating the translation. Translation:
Logical Fallacy Detection (Logic)	
BASE	Text: `{sentence1}`. You must answer one option from the listed categories without additional text.
AMRCOT	Text: `{sentence1}` AMR: `{amr1}` You must answer one option from the listed categories without additional text.
Event Extraction (Pubmed45)	
BASE	This question aims to assess your proficiency in validating relationships between different entities in biomedical text. You will be presented with a sentence from an article and asked to determine whether the interaction between the entities mentioned in the sentence is valid or not. You should respond with a single digit, either "0" if the interaction is invalid, "1" if it is valid, or "2" if swapping the positions of any two entities would make the interaction valid. Please note that you are required to provide only one of these three responses. Text: `{sentence1}` Interaction: `{interaction}`
AMRCOT	This question aims to assess your proficiency in validating relationships between different entities in biomedical text. You will be presented with a sentence from an article and its abstract meaning representation (AMR) and asked to determine whether the interaction between the entities mentioned in the sentence is valid or not. You should respond with a single digit, either "0" if the interaction is invalid, "1" if it is valid, or "2" if swapping the positions of any two entities would make the interaction valid. Please note that you are required to provide only one of these three responses. Text: `{sentence1}` AMR: `{amr1}` Interaction: `{interaction}`

Table 10: Prompts for PAWS, WMT16, Logic, and Pubmed45.

A.4 Example Data Samples

To get a better sense of how the data samples look, we provide some example (text, AMR) pairs in Appendix D.4.

A.5 Implementation Details

As for the experimental details, for the BERT and RoBERTa models, we use the weighted cross entropy loss, with a batch size of 16, learning rate of 1e-5, and dropout of 0.1, and train for five epochs until convergence. For the XGBoost classifier (Chen and Guestrin, 2016), we use the default hyperparameters, and set the random seed to 0, and the class weight proportional to the class ra-

Text2SQL (SPIDER)	
BASE	Write an SQL query that retrieves the requested information based on the given natural language question. Remember to use proper SQL syntax and consider any necessary table joins or conditions. Question: `{sentence1}` Query:
AMRCOT	Write an SQL query that retrieves the requested information based on the given natural language question and its abstract meaning representation (AMR). Remember to use proper SQL syntax and consider any necessary table joins or conditions. Question: `{sentence1}` AMR: `{amr1}` Query:
Named Entity Recognition (AMR-NER)	
BASE	The following is a named entity recognition task. Please extract all the named entities of the following types from the given sentence. TYPE="CARDINAL": Numerals that do not fall under another type, e.g., "one", "ten" TYPE="DATE": Absolute or relative dates or periods. E.g., "the summer of 2005", "recent years" TYPE="EVENT": Named hurricanes, battles, wars, sports events, etc. E.g., "Olympiad games" TYPE="FAC": Buildings, airports, highways, bridges, etc. E.g., "Disney", "the North Pole" TYPE="GPE": Countries, cities, states. E.g., "Hong Kong", "Putian" TYPE="LAW": Named documents made into laws. E.g., "Chapter 11 of the federal Bankruptcy Code" TYPE="LOC": Non-GPE locations, mountain ranges, bodies of water. E.g., "Mai Po Marshes", "Asia" TYPE="MONEY": Monetary values, including unit. E.g., "$ 1.3 million", "more than $ 500 million" TYPE="NORP": Nationalities or religious or political groups. E.g., "Chinese", "Buddhism" TYPE="ORDINAL": E.g., "first", "second", etc. TYPE="ORG": Companies, agencies, institutions, etc. E.g., "Eighth Route Army", "the Chinese Communist Party" TYPE="PERCENT": Percentage, including "%". E.g., "25 %" TYPE="PERSON": People, including fictional. E.g., "Zhu De", "Saddam Hussein" TYPE="PRODUCT": Objects, vehicles, foods, etc. (Not services.) E.g., "iPhone", "Coke Cola" TYPE="QUANTITY": Measurements, as of weight or distance. E.g., "23 sq. km" TYPE="TIME": Times smaller than a day. E.g., "homecoming night" Sentence: `{sentence1}` Use json format for the response where each key is an entity type.
AMRCOT	The following is a named entity recognition task. Please extract all the named entities of the following types from the given sentence and its abstract meaning representation (AMR). TYPE="CARDINAL": Numerals that do not fall under another type, e.g., "one", "ten" TYPE="DATE": Absolute or relative dates or periods. E.g., "the summer of 2005", "recent years" TYPE="EVENT": Named hurricanes, battles, wars, sports events, etc. E.g., "Olympiad games" TYPE="FAC": Buildings, airports, highways, bridges, etc. E.g., "Disney", "the North Pole" TYPE="GPE": Countries, cities, states. E.g., "Hong Kong", "Putian" TYPE="LAW": Named documents made into laws. E.g., "Chapter 11 of the federal Bankruptcy Code" TYPE="LOC": Non-GPE locations, mountain ranges, bodies of water. E.g., "Mai Po Marshes", "Asia" TYPE="MONEY": Monetary values, including unit. E.g., "$ 1.3 million", "more than $ 500 million" TYPE="NORP": Nationalities or religious or political groups. E.g., "Chinese", "Buddhism" TYPE="ORDINAL": E.g., "first", "second", etc. TYPE="ORG": Companies, agencies, institutions, etc. E.g., "Eighth Route Army", "the Chinese Communist Party" TYPE="PERCENT": Percentage, including "%". E.g., "25 %" TYPE="PERSON": People, including fictional. E.g., "Zhu De", "Saddam Hussein" TYPE="PRODUCT": Objects, vehicles, foods, etc. (Not services.) E.g., "iPhone", "Coke Cola" TYPE="QUANTITY": Measurements, as of weight or distance. E.g., "23 sq. km" TYPE="TIME": Times smaller than a day. E.g., "homecoming night" Sentence: `{sentence1}` AMR: `{amr1}` Use json format for the response where each key is an entity type.

Table 11: Prompts for SPIDER and AMR-NER.

tio, namely setting the positive weight to be the inverse of the number of samples in the positive class divided by that of the negative class.

A.6 Details of Ablation Study

For text/AMR ablation experiments, we use AMRCOT prompt with portions of text/AMR string ablated. An example of ablating 100% of the AMR is as follows:

"You are given a text and its abstract meaning representation (AMR).
Text: The relationship between Obama and Netanyahu is not exactly friendly.
AMR:

PAWS	You are an NLP assistant whose purpose is to perform Paraphrase Identification. The goal of Paraphrase Identification is to determine whether a pair of sentences have the same meaning.
WMT16	You are an NLP assistant expert in machine translation from English to German.
Logic	You are an expert in logic whose purpose is to determine the type of logical fallacy presented in a text or complete the text with one of the following logical fallacies. 1) Faulty Generalization 2) False Causality 3) Circular Claim 4) Ad Populum 5) Ad Hominem 6) Deductive Fallacy 7) Appeal to Emotion 8) False Dilemma 9) Equivocation 10) Fallacy of Extension 11) Fallacy of Relevance 12) Fallacy of Credibility 13) Intentional Fallacy.
Pubmed45	You are a medical professional expert.
SPIDER	You are a language model designed to generate SQL queries based on natural language questions. Given a question, you need to generate the corresponding SQL query that retrieves the requested information from a database.
AMR-NER	You are an NLP assistant whose purpose is to perform named entity recognition (NER).

Table 12: System prompts for all datasets.

Please translate the text from English to German. You can refer to the provided AMR if it helps you in creating the translation. Translation:"

We also provide an example of ablating 100% of the text:

"You are given a text and its abstract meaning representation (AMR).
Text:
AMR:
(f / friendly-01 9
:ARG1 (r / relation-03 1
:ARG0 (p / person 3
:name (n / name 3
:op1 "Obama" 3))
:ARG2 (p2 / person 5
:name (n2 / name 5
:op1 "Netanyahu" 5)))
:mod (e / exact 8)
:polarity - 7)
Please translate the text from English to German. You can refer to the provided AMR if it helps you in creating the translation. Translation:"

B Data Collection

B.1 Composing the Slang-Involved Paraphrase Detection Dataset

Since our experiments need annotations for both slang paraphrase pairs and AMRs, we compose two datasets, GoldSlang-ComposedAMR, and GoldAMR-ComposedSlang. For GoldSlang-ComposedAMR, we use the curated slang paraphrase pairs by Tayyar Madabushi et al. (2021), and generate their AMRs with an off-the-shelf parser (Drozdov et al., 2022). For the other dataset, GoldAMR-ComposedSlang, we use gold AMRs from the LDC AMR 3.0 corpus (Banarescu et al., 2013), and compose slang paraphrases using a combination of human efforts and assistance from GPT-4.

Composing the GoldSlang-ComposedAMR Dataset We adapt a subset of the ASILM (Tayyar Madabushi et al., 2021), an idiomatic MWE dataset, into a paraphrase detection task. Each sentence in the subset containing idiomatic expressions is paired with a paraphrase (where the idiom is replaced with its literal semantic equivalent) and a non-paraphrase (where the idiom is replaced with a phrase of similar superficial meaning but differing semantic meaning). This results in a balanced paraphrase detection dataset with respect to ground truth labels.

Composing the GoldAMR-ComposedSlang Dataset A possible error in AMRCoT lies in the imperfection of parser-generated AMRs. To disentangle the harm caused by (1) incorrect AMRs produced by the parsers and (2) poor representation of slang expressions by AMRs, we handcrafted the GoldAMR-Slang-Para dataset. We first extract a subset from LDC-AMR3.0 (Banarescu et al., 2013) that involve slang expressions. Then, for each sentence, we replace the slang expression with an alternative expression of the same meaning, and a semantically different expression which seems literally similar, thus creating a paraphrase and non-paraphrase sentence, respectively. The corresponding AMRs can be derived from the original LDC-AMR3.0 AMRs with minimal modifications.

Specifically, we operationalize the process as follows. We first use gpt-3.5-turbo-0613 to identify 500 samples of slang usage from LDC-AMR3.0 with the following prompt:

Please evaluate the following sentence for the presence of slang expressions. A slang expression is a phrase or expression that is in the online slang dictionaries and has a meaning that is very different from its literal form. For instance, 'raining cats and dogs' is slang, while 'middle school' is not. Although 'middle school' is a compound phrase, it does not carry a meaning beyond its literal interpretation. Here is the sentence for your analysis: premise. Please format your response as follows: 'Yes or No, slangs.'
If there's no slang used, just answer 'No'. If there are multiple slang expressions, please separate them with a semicolon (';'). Remember, the idioms we are interested in are those that, when taken literally, would have a completely different semantic meaning.

Then we mannually check whether the extracted expressions are slang and are appropriate. Consistent with the spirit of (Zhang et al., 2019), we use the following prompt to query gpt-3.5-turbo-0613 to generate one paraphrase and one non paraphrase of each sentence.

Rewrite the following sentence in two ways Sentence: sentence 1. Replacing "slang" with its intended meaning. 2. Replacing "slang" with its literal meaning, such that the sentence loses its original meaning. Do not change anything else

Lastly, for each pair of (original_sentence, (non)paraphrase_sentence), we give (original_sentence, original_amr, (non)paraphrase_sentence) to gpt-3.5-turbo-0613, and ask it to generate (non)paraphrase_amr by minimally modifying the original_amr. The prompt is as follows:

"The AMR of the sentence 'og_sentence' is og_amr
What is the AMR of the sentence 'paraphrase'? Modified the given AMR to fit the sentence 'hypothesis' and words not present in the sentence 'hypothesis' should not appear in your AMR. Start you response with '('."

B.2 Intuition of Why AMR Might Be Helpful for NER

For some intuition of why we choose the NER task out of the OntoNotes 5.0 dataset, it has already been shown in existing work that AMR can help event extraction (Garg et al., 2015; Huang

et al., 2018), which is also a type of named entities. Specifically, the graphical structure and typed tags of AMR make it easy to identify named entities. For instance, in the sentence "the top money funds are currently yielding well over 9%" in the AMR-NER dataset, we discover the AMR substructure "(p / percentage-entity 10 :value 9)," which makes it easy to identify "9%" as a named entity of type *percent*.

C Explanation of Linguistic Features

In our analysis, we delve into specific linguistic features that exhibit strong correlations with AMR helpfulness, as detailed in Tables 6 and 7.

Number of Adjuncts This feature involves counting words that serve as modifiers to nouns, pronouns, verbs, and other parts of speech. Adjuncts typically provide additional context or emphasis but can be omitted without altering the core meaning of the sentence. For example, in "John really likes apples," the word "really" is an adjunct, modifying the verb "likes."

Word Complexity We assess word complexity using the *age of acquisition* metric, following the methodology of Kuperman et al. (2012).

Number of Quantifier Phrase Modifiers This feature quantifies the modifiers within quantifier phrases that adjust the head, or primary element, of the phrase. An illustration of this can be seen in the sentence "About 5000 people attended the conference," where "about" modifies the quantifier "5000." This concept is further explained by de Marneffe and Manning (2008).

D Additional Experiments

D.1 Statistical Significance Tests

We conduct statistical significance tests for the experiments in the main paper comparing BASE and AMRCOT, including Tables 3, 5 and 9. Using the Student's t-test (Student, 1908), we report the significance test results in Table 13. The results echo our earlier observation that the changes by AMRCOT is mostly small-scale fluctuations, as the differences are statistically insignificant in most cases.

Dataset	BASE	ΔAMRCOT	Sig. (p)
PAWS	78.25	-3.04	✓ (5.209e-8)
WMT	27.52	-0.83	✗ (0.0716)
Logic	55.61	-0.49	✗ (0.7300)
Pubmed45	39.65	-3.87	✓ (0.0309)
SPIDER	43.78	+0.61	✗ (0.4362)
GoldSlang-ComposedAMR	86.83	-6.63	✓ (0.0014)
GoldAMR-ComposedSlang	77.69	-8.78	✗ (0.1309)
AMR-NER (Gold AMR)	60.51	+0.03	✗ (0.9935)
AMR-NER (Parser AMR)	60.51	+1.91	✗ (0.6227)

Table 13: For all the experiments comparing BASE and AMRCOT using GPT-4 mentioned in the main text, we calculate whether the difference ΔAMRCOT is statistically significant (Sig.) using t-test (Student, 1908) by the threshold $p = 0.05$, and report the actual p values.

Model	BERTScore		spBLEU	
	BASE	ΔAMRCOT	BASE	ΔAMRCOT
text-d.-001	90.48	-1.09	30.29	-4.19
text-d.-002	91.00	-0.30	33.14	-1.67
text-d.-003	91.37	-0.25	34.75	-1.55
GPT-3.5	91.70	-0.06	37.09	-0.43
GPT-4	91.79	-0.08	37.71	-0.72

Table 14: Performance on WMT by additional metrics, BERTScore and spBLEU.

D.2 Additional Evaluation Results for Machine Translation

In the main paper, we mainly report the performance of machine translation using the standard evaluation metric BLEU (Papineni et al., 2002). Recent studies has proposed new metrics to evaluate the quality of machine translation, such as BERTScore (Zhang et al., 2020) and spBLEU (Goyal et al., 2022), so we also report the model performance according to these two additional metrics in Table 14. We use the version of spBLEU built from the Flores-200 dataset. [3] The performance trend is consistent with Section 4, where AMR has a marginal effect on the baseline LLM performance.

D.3 Larger-Scale Ablation Study Using WMT

We understand that the ablation study in Section 6.2 is on a small scale (in order to use the gold annotated data). As an alternative tradeoff to regress a bit on the data quality, but aim at a larger scale, we also conduct the same ablation study on 1,000 random samples from the WMT dataset using predicted AMRs. Our results in Figure 5 also confirm that text has a more instrumental role for LLMs.

[3] https://github.com/facebookresearch/flores/tree/main/flores200

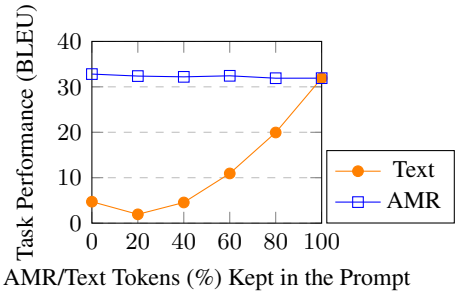

Figure 5: Ablation studies of AMR and text representations in the prompt on 1,000 random samples of the WMT dataset using GPT-4. Starting from the AMRCoT prompt with the complete text and AMR, we randomly drop out a certain portion of tokens in the text/AMR, and see the effect on the task performance.

D.4 Few-Shot Experiments

In addition to the main results of the overall effect of AMR in Section 4 using zero-shot prompting, we also check whether adding few-shot examples to the prompt will help. Since the experiments are very costly, we conduct a small-scale preliminary check running few-shot AMRCoT on 200 random samples from PAWS using `gpt-3.5-turbo-0613`. As AMRs are lengthy, and PAWS is a binary classification task, we select one random example with the positive label, and another with the negative label, totaling an average prompt length of 371 tokens. The resulting F1 score is 63.67, close to BASE performance of 63.92, serving as a preliminary observation that the few-shot setting might not change our observations much.

Paraphrase Detection (PAWS)	
Text	Sentence 1: The defendant then broke into the house and tried unsuccessfully to remove and open the vault. Sentence 2: The defendant then broke into the house and tried unsuccessfully to open the safe and then to remove them.
AMR	AMR1: (a / and 7 :op1 (b / break-02 3 :ARG0 (d / defendant 1) :ARG1 (h / house 6)) :op2 (t2 / try-01 8 :ARG0 d :ARG1 (a2 / and 12 :op1 (r / remove-01 11 :ARG0 d :ARG1 (v / vault 15)) :op2 (o / open-01 13 :ARG0 d :ARG1 v)) :ARG1-of (s / succeed-01 9 :ARG0 d :polarity - 9)) :time (t / then 2)) AMR2: (a / and 7 :op1 (b / break-02 3 :ARG0 (d / defendant 1) :ARG1 (h / house 6)) :op2 (t2 / try-01 8 :ARG0 d :ARG0-of (s2 / succeed-01 9 :polarity - 11) :ARG1 (a2 / and 14 :op1 (o / open-01 11 :ARG0 d :ARG1 (s / safe 13)) :op2 (r / remove-01 17 :ARG0 d :ARG1 s))) :time (t / then 2))
Translation (WMT16)	
Text	Obama receives Netanyahu
AMR	(r / receive-01 1 :ARG0 (p / person 0 :name (n / name 0 :op1 "Obama" 0)) :ARG1 (p2 / person 2 :name (n2 / name 2)) :rel (e / Netanyahu 2))
Logical Fallacy Detection (Logic)	
Text	On my walk to work this morning, a woman on her bike nearly ran me off the sidewalk. I hadn't realized that cyclists were so aggressive and rude!
AMR	(m2 / multi-sentence 19 :snt1 (n / near-02 13 :ARG2 (r3 / run-10 14 :ARG0 (b / bike 12 :poss (w2 / woman 9)) :ARG1 (i / i 15) :ARG2 (s / sidewalk 18) :time (w / walk-01 2 :ARG0 i :ARG2 (w3 / work-01 4 :ARG0 i) :time (d / date-entity 6 :dayperiod (m / morning 6) :mod (t / today 5))))) :snt2 (r / realize-01 23 :ARG0 (i2 / i 20) :ARG1 (h / have-degree-91 27 :ARG1 (p / person 25 :ARG0-of (c / cycle-01 25)) :ARG2 (a2 / and 29 :op1 (a / aggressive 28) :op2 (r2 / rude-01 30 :ARG1 p)) :ARG3 (s2 / so 27)) :polarity - 22))
Event Extraction (Pubmed45)	
Text	Examination of activated phospho-MEK levels revealed that the FBm and S729 mutations had no effect on MEK activation induced by the high-activity V600E B-Raf protein; however, the FBm and S729A mutations increased and decreased, respectively, the abilities of the intermediate G466A and kinase-impaired D594G B-Raf proteins to activate MEK (Fig. 4B), indicating a correlation between the transformation potential of these proteins and their ability to activate ERK cascade signaling in vivo.
AMR	(c3 / contrast-01 35 :ARG1 (r / reveal-01 7 :ARG0 (e4 / examine-01 0 :ARG1 (l / level 6 :ARG1-of (a / activate-01 2) :mod (e / enzyme 5 :name (n / name 5 :op1 "MEK" 5) :ARG1-of (p / phosphorylate-01 3)))) :ARG1 (a6 / affect-01 17 :ARG0 (m / mutate-01 14) :ARG1 (a2 / activate-01 20 :ARG1 (p3 / protein 19 :name (n5 / name 19 :op1 "MEK" 19)) :ARG1-of (i4 / induce-01 21 :ARG0 (p4 / protein 33 :name (n2 / name 5 :op1 "FBm" 10 :op1 "V" 27 :op2 600 28 :op3 "B-Raf" 30) :ARG0-of (a5 / activity-06 26 :ARG1-of (h / high-02 24))))) :polarity - 16) :ARG1-of (d2 / describe-01 76 :ARG0 (f / figure 73 :ARG0-of (i3 / indicate-01 78 :ARG1 (c4 / correlate-01 80 :ARG1 (p2 / potential 84 :domain (t2 / transform-01 83 :ARG0 (p7 / protein 87 :mod (t / this 86)))) :ARG2 (c2 / capable-01 90 :ARG1 p7 :ARG2 (a4 / activate-01 92 :ARG0 p7 :ARG1 (s / signal-07 95 :ARG0 (e3 / enzyme 93 :name (n7 / name 93 :op1 "ERK" 93 :op2 "cascade" 94))) :manner (v / vivo 97)))))) :mod 4B 75))) :ARG2 (a7 / and 45 :op1 (i2 / increase-01 44 :ARG0 (m2 / mutate-01 43 :ARG1 (e2 / enzyme 19 :name n2)) :ARG1 (c / capable-01 51 :ARG1 (a8 / and 58 :op1 m2 :op1 (p5 / protein 68 :name n2) :op2 (m3 / mutate-01 43 :ARG1 p3) :op2 (p6 / protein 68 :name (n6 / name 68 :op1 "G" 55 :op2 466 56 :op3 "A" 57) :ARG2-of (i / impair-01 61 :ARG1 (k / kinase 59))))) :ARG2 (a3 / activate-01 70 :ARG0 a8 :ARG1 e2))) :op2 (d / decrease-01 46 :ARG0 m3 :ARG1 c)) :rel (n3 / name 5 :op1 "D" 62 :op2 "594" 63) :rel (n4 / name 12 :op1 "S" 12 :op2 "729" 13) :rel (f2 / figure 73) :rel (i5 / intermediate 54))
Text2SQL (SPIDER)	
Text	List the number of all matches who played in years of 2013 or 2016.
AMR	(l / list-01 0 :ARG0 (y / you 0) :ARG1 (n / number 2 :quant-of (m / match-03 5 :ARG0-of (p / play-01 7 :time (o / or 12 :op1 (d / date-entity 11 :year 2013 11) :op2 (d2 / date-entity 13 :year 2016 13))) :mod (a / all 4))) :mode imperative 0)
Named Entity Recognition (AMR-NER)	
Text	Then, in the guests' honor, the speedway hauled out four drivers, crews and even the official Indianapolis 500 announcer for a 10-lap exhibition race.
AMR	(h / haul-01 10 :purpose (h2 / honor-01 6 :ARG1 (g / guest 4)) :purpose (r / race-02 29 :ARG1-of (e3 / exhibit-01 28) :ARG3 (l / lap 27 :quant 10 25)) :ARG0 (s / speedway 9) :ARG1 (a / and 14 :op1 (p / person 13 :quant 4 12 :ARG0-of (d / drive-01 13)) :op2 (c / crew 15) :op3 (p2 / person 15 :ARG0-of (a2 / announce-01 22 :ARG1 (e2 / event 21 :name (n / name 20 :op1 "Indianapolis" 20 :op2 500 21))) :mod (e / even 17) :mod (o / official 19))) :direction (o2 / out 11) :time (t / then 0))

Table 15: Example text-AMR pairs for each task.