Semantic Parsing for Evaluating Large Language Models: Separating Linguistic Abilities with YARN

Rémi de Vergnette, Maxime Amblard
Université de Lorraine, CNRS, Inria, LORIA, F-53999 Nancy, France
{remi.de-vergnette,maxime.amblard}@loria.fr

Abstract

We evaluate large language models (LLMs) through semantic parsing into YARN, a structured meaning representation that distinguishes predicate–argument structure from higher-level linguistic features such as tense, aspect, and modality. For evaluation, we employ SMATCHY, a fine-grained metric designed to assess different layers of meaning independently. Our experiments test multiple LLMs under varied conditions, including inference modes, linearization formats (JSON and logic-inspired CFG), and the presence or absence of auxiliary supervision via partial semantic parses. Results show that model performance is highly sensitive to both representational design and supervision, with no single configuration consistently outperforming the others. While some models gain from additional semantic information in prompts, others are negatively affected. A layer-wise analysis indicates that surface-level features such as temporality and negation are captured more reliably than deeper semantic phenomena like quantification. Consistent with prior work, our findings highlight the limited capacity of current LLMs to generate fully formal meaning representations.

Keywords: semantic parsing, language models, meaning representation, evaluation, YARN, Smatchy

1. Introduction

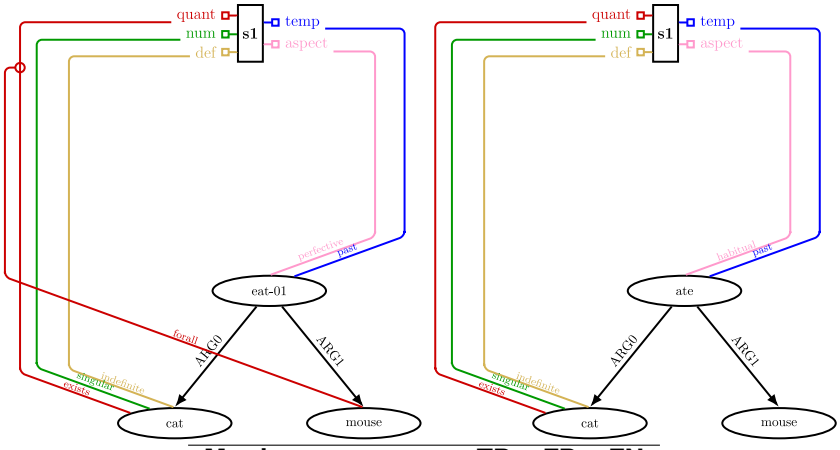
LLMs excel at handling surface-level linguistic patterns but show inconsistent performance on tasks requiring deeper semantic understanding. Semantic parsing maps natural language to formal meaning representations (Liang, 2016). Unlike syntax, where LLMs generalize well (Blevins et al., 2023), off-the-shelf models still perform poorly on formalisms like AMR (Banarescu et al., 2013), lagging behind specialized (Ettinger et al., 2023; Shi et al., 2025) or even pre-transformer parsers such as JAMR (Flanigan et al., 2014). Beyond practical use in knowledge base interaction (Kamath and Das, 2018), semantic parsing directly tests a model's ability to produce structured meaning, offering a clear probe of linguistic competence without task-specific shortcuts (Yuan et al., 2024). It thus serves as a middle ground between abstract linguistic benchmarks (Warstadt et al., 2020) and application-specific evaluations (Wein and Opitz, 2024).

Recent studies have used semantic parsing to assess LLMs' natural language understanding (Ettinger et al., 2023; Schneider et al., 2024), consistently finding that current models lag behind smaller, task-specific systems. However, such evaluations face key limitations. Most adopt a single linearization or encoding scheme for meaning representations, making it difficult to disentangle true linguistic competence from familiarity with a particular notation (see Figure), and introducing bias toward models fine-tuned on that format. Moreover, using broad formalisms like AMR (Banarescu et al., 2013) forces evaluations to conflate distinct phenomena—such as event identification, tense, and aspect—within one metric. Because standard graph-matching metrics aggregate these factors into a single score, they obscure which specific linguistic abilities are being tested. This reflects a broader limitation of most meaning representations for LLM evaluation: they either lack coverage of key semantic phenomena or encode them in ways that hinder automatic, multidimensional assessment—leaving fine-grained analysis largely dependent on manual inspection.

Beyond these issues, current evaluations often use widely available formalisms, raising concerns about potential data contamination.

This paper uses YARN (laYered meAning RepresentatioN)(Pavlova et al., 2024; Pavlova, 2025), a meaning representation formalism based on Abstract Meaning Representation AMR (Banarescu et al., 2013) that disentangles two distinct dimensions: predicate-argument relations and more subtle linguistic phenomena such as tense, aspect, modality, and discourse coherence. YARN is sufficiently novel to avoid significant online presence while remaining grounded in established linguistic theory. We pair this with SMATCHY (de Vergnette et al., 2025), a fine-grained evaluation metric for assessing performance across specific aspects of the representation. By evaluating off the shelf LLMs on parsing toward the YARN formalism, we provide a way to make a nuanced answer : By testing multiple linearizations of YARN and evaluating different linguistic dimensions with the SMATCHY metric family, we can probe how well models capture the

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 9745–9755
11–16 May 2026.



Metric	TP	FP	FN
smatchy_general	10	2	3
smatchy_predicative	5	1	1
smatchy_feature	4	1	2
smatchy_temp	2	0	0
smatchy_aspect	1	1	1
smatchy_num	2	0	0
smatchy_def	2	0	0
smatchy_quant	2	0	1

Figure 1: Gold reference for YARN representation of "Some cat ate every mouse", alongside a candidate parse from an LLM ; While temporality, number and definitiveness are accurately extracted, and quantification is partially correct, aspect and event labelling are misrepresented. This is reflected in the SMATCHY breakdown table, showing True Positives (TP), False Positives (FP), and False Negatives (FN) for each submetric.

various phenomena encoded in the representation. Strong performance on a given phenomenon in at least one linearization indicates some level of understanding of that aspect of meaning. Consistent performance across several linearizations, however, would suggest a deeper grasp of the underlying semantic structure encoded by YARN, rather than mere familiarity with a specific format.

Our approach aligns with a broader line of research evaluating the linguistic competence of Language Models through linguistically informed structured probing tasks, targeting syntax (Warstadt et al., 2019), semantics (Ettinger, 2020), and compositional generalization (Elazar et al., 2021; Conklin et al., 2023). While such methods provide valuable insights into implicit linguistic knowledge, they often rely on isolated templates or classification-based setups. In contrast, our work complements these approaches by requiring full structured semantic representations, enabling finer-grained assessment of meaning-level phenomena through a formalism like YARN.

Our experimental setup evaluates different LLMs in both standard and reasoning-based inference modes, compares performance across two distinct YARN linearizations, and analyzes the impact of auxiliary supervision. This design disentangles the effects of reasoning ability, representational format, and external guidance, providing a more nuanced picture of LLM semantic parsing abilities than prior work. Prior work suggests that off-the-shelf LLMs do not reliably parse natural language into semantic formalisms. Rather than treating this as a binary limitation, we focus on characterizing how and why failures occur, and on identifying practical indications for what an effective semantic parsing pipeline leveraging LLMs should look like.

The remainder of this paper is organized as follows: we first present the YARN formalism in Section 2, then the evaluation protocol in Section 3. We evaluate the results in Section 4, and conclude in Section 5.

2. YARN Formalism

Meaning representations typically fall into two main categories: (i) logic-based approaches grounded Fregean semantics (Frege, 1967), and (ii) graph-based representations inspired by conceptual graphs (Sowa, 1984) and widely used in knowledge representation.

YARN belongs to the second category. It is a graphical meaning representation formalism derived from Abstract Meaning Representation (AMR), while preserving logical expressivity. Unlike AMR, YARN does not encode underspecification by de-

fault. It clearly separates predicate–argument structures (depicted as graphs with black nodes and edges) from feature annotations (represented as colored layers that modify or constrain meaning).

Figure 2 illustrates this layered architecture. Layers annotate information distinct from the sentence's predicate–argument structure, capturing nuanced aspects that shape meaning while remaining largely independent of world knowledge. Examples range from the temporality layer encoding tense to the quantification layer, which structures to the logical interpretation of the sentence.

Layers in YARN are not always independent; they can interact and modify one another to capture interactions. For instance, the negation of a modal expression ("I cannot find the passage") differs semantically from the possibility of negation ("I may not find the passage"), as illustrated in Figure 2. In YARN, such differences are modeled through distinct layer orderings. Furthermore, certain feature layers can themselves give rise to new layers, allowing the representation of scope-sensitive phenomena such as nested quantification or tense–aspect interactions (e.g., future-in-the-past vs. past-in-the-future).

The main advantage of YARN lies in its modular layer design, which allows the selection of specific linguistic features for evaluation. A YARN annotation can be restricted to a subset of features, enabling targeted assessments of individual linguistic competencies. This makes YARN especially well suited for disentangled, fine-grained semantic evaluation of language models.

2.1. SMATCHY Metric

The SMATCHY metric family (de Vergnette et al., 2025) extends the widely used SMATCH metric (Cai and Knight, 2013) for evaluating AMR structures. Like SMATCH, it follows a match-then-score approach: given a reference YARN structure R and a candidate C, both are decomposed into sets of atomic elements (generalized nodes and relations). The algorithm then identifies an optimal alignment between elements of R and C, to computes precision, recall, and F1-score.

SMATCHY extends SMATCH by enabling fine-grained filtering.Users can limit scoring to:

- A specific element type set : predicates (nodes), arguments(edges), feature modifiers(colored edges), or
- A particular (set of) annotation layer(s) (e.g., temporality, aspect, modality).

These filters can also be combined. For example, the SMATCHY-FOL metric evaluates performance on elements relevant to first-order logic interpretation,

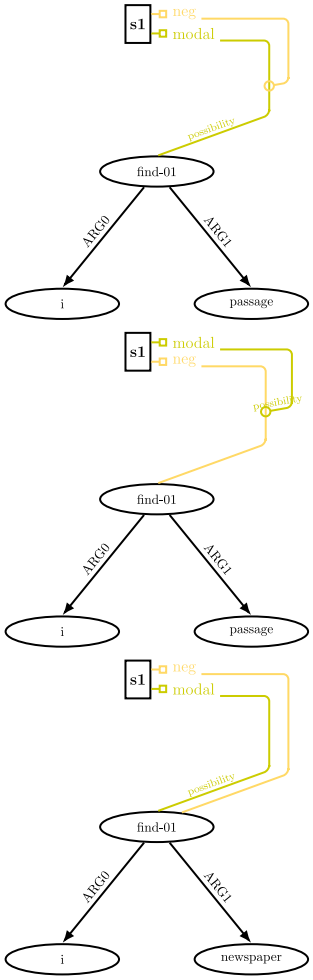


Figure 2: Interaction between modality and negation: (1) Negation of possibility ("I cannot find the passage"), (2) Possibility of negation ("I may not find the passage"), and (3) Underspecified layering that encodes ambiguity.

including predicates, arguments, and the negation and quantification layers.

SMATCHY does not enforce interpretive equivalences. In other words, structurally distinct annotations that may be logically equivalent under certain theories (e.g., commutative quantifier orderings) are treated as different. SMATCHY is a strictly structural metric, providing a conservative lower bound on semantic adequacy—useful for tasks where exact structural correspondence matters, such as logical inference or contradiction detection.

2.2. Linearizing YARN

To interface with text-only models, we define two linearization strategies for YARN graphs:

JSON-Based Linearization This method closely follows the original formal mathematical definition of YARN, introduced by Pavlova et al. (2024). It explicitly encodes graph structure through nodes and edges, serialized in a JSON format. An example for the first sentence in Figure 2 is shown below:

```
{
"labels": {
"f": "find-01",
...
},
"s": [ "s1", ..., delim]
"v": [ "f", ..., delim]
...
"e": {
"e1": [ "f", "ARG0", "i", delim]
...
},
"l": { "l1": ["modal",
    ↪ "possibility", "", delim],
"h": { "h1": ["neg", "", "l1", delim],
...
}
```

Where "s","v", "e", and "l" /"h" represent sets of event nodes, nodes, edges, layers edges. Each element is identified by a unique key (e.g., "f" for the predicate "find-01"). This linearization takes advantage of the widespread presence of structured data formats in LLM pretraining corpora, making the graph topology explicitly encoded.

CFG Linearization The second approach is inspired by logical formalisms. It separates predicate–argument structures from feature components, representing the latter as scope-taking operators enclosed in parentheses. For the same example:

Predicative part:

$$\text{ARG0}(f, i) \wedge \text{ARG1}(f, n) \wedge \text{find-01}(f) \\ \wedge\ \text{i}(i) \wedge \text{passage}(n)$$

Feature part:

$$s_1 := [(H\langle\text{neg}\rangle\, H\text{-modal-possibility}\rangle(f)]$$

This linearization is more token-efficient and makes both the semantic structure and operator scope more transparent. However, it is entirely unfamiliar to LLMs, providing an opportunity to test how different but equivalent ways of encoding the same formalism affect their performance.

Together, these two linearizations reflect the dual nature of YARN — combining the characteristics of both a graph-based and a logic-inspired representation.

2.3. Dataset

We use a set of 100 YARN annotations published alongside (Pavlova, 2025), based on short sentences from the Parallel Universal Dependencies English treebank Nivre et al. (2015). The sentences contain an average of 7.22 words ranging from 3 ("Drop the mic") to 10 (" These plant families are still present in Papua New Guinea."). Annotated data is publicly available on https://github.com/YARN-World, and can be explored on semantics.grew.fr through the use of GREW (Guillaume et al., 2012) requests.

These annotations serve as reference data for evaluation. An additional set of five human-annotated sentences, covering both complex and simple examples, is used as few-shot prompts.

3. Protocol

Varied conditions We evaluate the capacity of different LLMs to perform semantic parsing into the YARN formalism under varying conditions. Our experimental design manipulates three factors: the choice of LLM, the YARN linearization format, and the presence of auxiliary supervision.

The evaluated LLMs include Qwen3 (Yang et al., 2025)(tested in both Thinking and standard modes), LLaMA 4 (Scout and Maverick) (Meta AI, 2025), and Mixtral 8x7B (Mistral AI, 2023).

We design a custom prompt for each linearization format. Auxiliary supervision is provided by including, in the corresponding format, the predicate–argument structure of the reference semantic representation without any feature annotations. This setup simulates the output of an upstream AMR-like parser that produces a partial semantic parse, which the model must then enrich with additional linguistic features.

This setup allows us to isolate the model's ability to capture the more subtle semantic phenomena represented in YARN, independently of its predicate identification performance. Predicates correspond to PropBank concepts (Kingsbury and Palmer, 2002) rather than English surface words, making their identification a distinct task that relies on lexical and resource knowledge. In contrast, feature modifiers primarily use natural English words and draw from a smaller vocabulary, making them easier to include explicitly in prompts and to illustrate through few-shot examples.

We instruct the model to annotate the following semantic layers: temporality (e.g., past), aspect (e.g., habitual), quantification (e.g., existential), definiteness (definite vs. indefinite), negation, and number (singular vs. plural). The detailed use and interpretation of these layers for semantic annotation are thoroughly defined in (Pavlova, 2025).

Prompting We employ few-shot prompting with five examples. The model is instructed to generate

the YARN representation within an XML tag to facilitate automatic extraction. Sampling parameters are kept at their default settings.

Each prompt follows the structure below:

- Instruction: Explains the task, the YARN formalism, and the possible values for each feature.
- Format specification: Defines the linearization format to be used and the XML tag for the output.
- Examples: Provides five example sentence–representation pairs in the specified format.
- Test instance: Presents the target sentence to be parsed, optionally including its predicate–argument structure.

This standardized prompt design ensures consistent guidance across all experimental settings, enabling controlled comparisons between models and configurations.

Analysis Results are analyzed using SMATCHY. When a model fails to produce output conforming to the required linearization scheme, the result is treated as an empty annotation.

We use several variants of the SMATCHY metric:

- SMATCHY-GENERAL: Global similarity over the full representation.
- SMATCHY-PA/ SMATCHY-FEATURE: Evaluation limited to the predicate–argument structure (resp. feature layers).
- SMATCHY-{temporality, aspect, negation}: Evaluation restricted to a single feature layer.

F1 scores are computed on the full test set to ensure comparability across models and conditions.

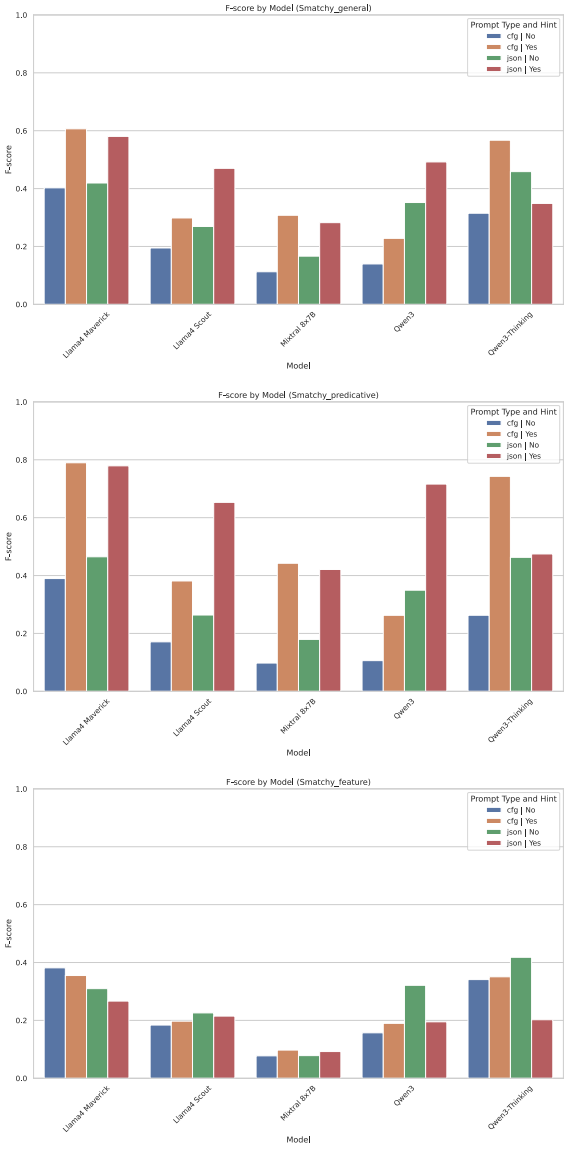


Figure 3: F1 scores for different LLMs across linearization formats and auxiliary supervision conditions, evaluated using the SMATCHY-GENERAL (overall performance) SMATCHY-PA (performance on the predicative part) and SMATCHY-FEATURE (performance on the feature part) metrics.

4. Evaluation

4.1. General Observations

As shown in Tables 1 and 2, we observe strong effects of both the linearization format and the presence of auxiliary supervision on model performance. However, these effects are not consistent across models, suggesting that different LLMs exhibit distinct sensitivities and failure modes when parsing into YARN.

As noted in prior work (Ettinger et al., 2023), a substantial proportion of model errors arises from failures to adhere to the expected output format, resulting in representations that cannot be parsed. This issue occurs most frequently with the CFG linearization, which is less structured and therefore more difficult to generate correctly. In contrast, the JSON format—being more explicit—produces fewer invalid outputs but tends to exhibit subtler structural errors, such as references to nonexistent nodes or omissions of required elements like the root “S” node.

A few general trends can be identified:

Fitrst, auxiliary supervision generally enhances overall parsing performance, particularly as reflected in higher SMATCHY-GENERAL F1 scores. This improvement stems primarily from increased accuracy in the predicate–argument structure

model	prompt	pa_hint	SmatchY-General	SmatchY-PA	SmatchY-Feature	non valid
Llama4 Maverick	cfg	no	0.40	0.39	0.39	0.20
		yes	0.61	0.79	0.37	0.25
	json	no	0.43	0.46	0.35	0.13
		yes	0.60	0.78	0.32	0.16
Llama4 Scout	cfg	no	0.20	0.17	0.19	0.63
		yes	0.30	0.38	0.21	0.65
	json	no	0.27	0.26	0.23	0.22
		yes	0.47	0.65	0.22	0.26
Mixtral 8x7B	cfg	no	0.11	0.10	0.09	0.68
		yes	0.31	0.44	0.09	0.62
	json	no	0.17	0.18	0.08	0.45
		yes	0.29	0.42	0.12	0.62
Qwen3	cfg	no	0.14	0.11	0.15	0.80
		yes	0.23	0.26	0.19	0.80
	json	no	0.35	0.35	0.31	0.15
		yes	0.49	0.72	0.20	0.31
Qwen3-Thinking	cfg	no	0.32	0.26	0.35	0.43
		yes	0.57	0.74	0.35	0.39
	json	no	0.47	0.46	0.44	0.13
		yes	0.35	0.47	0.21	0.67

Table 1: Results of semantic parsing into Yarn using different LLMs, linearization formats, and auxiliary supervision conditions. Scores are reported using the SmatchY metric family: general similarity (SmatchY-General), predicate-argument structure (SmatchY-PA), and feature layers (SmatchY-Feature). The proportion, of non valid structure that were produced is indicated in the "non valid" column.

model	prompt	pa_hint	SmatchY-temp	SmatchY-def	SmatchY-neg	SmatchY-quant
Llama4 Maverick	cfg	no	0.74	0.50	0.26	0.40
		yes	0.67	0.54	0.29	0.35
	json	no	0.71	0.47	0.41	0.29
		yes	0.72	0.33	0.49	0.24
Llama4 Scout	cfg	no	0.38	0.31	0.06	0.29
		yes	0.39	0.27	0.18	0.23
	json	no	0.59	0.37	0.42	0.18
		yes	0.56	0.31	0.53	0.22
Mixtral 8x7B	cfg	no	0.34	0.06	0.06	0.07
		yes	0.37	0.08	0.00	0.02
	json	no	0.24	0.03	0.24	0.04
		yes	0.32	0.05	0.15	0.12
Qwen3	cfg	no	0.26	0.26	0.06	0.23
		yes	0.28	0.28	0.06	0.28
	json	no	0.62	0.47	0.35	0.45
		yes	0.55	0.21	0.49	0.24
Qwen3-Thinking	cfg	no	0.53	0.50	0.22	0.37
		yes	0.60	0.52	0.32	0.39
	json	no	0.68	0.68	0.63	0.47
		yes	0.39	0.31	0.45	0.12

Table 2: Layer-wise results for selected feature layers (temporality, definiteness, negation, quantification) using the SmatchY metric family

(SmatchY-PA). While there are also gains in the feature structure, as indicated by SmatchY-Feature and feature-specific SmatchY metrics, these improvements are comparatively smaller. Notably,

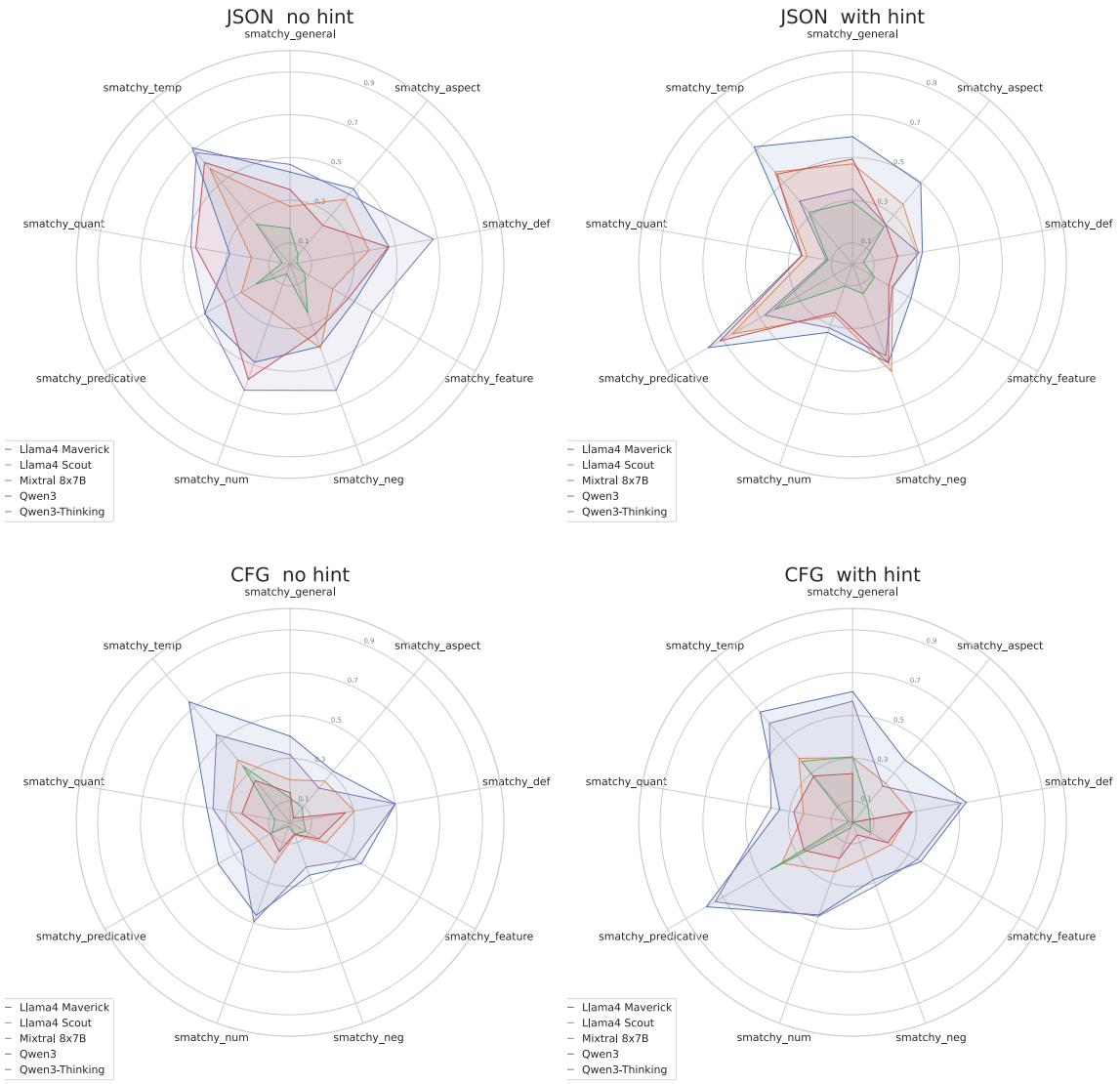


Figure 4: Fine-grained scores. Each radar chart corresponds to a specific combination of linearization format (JSON or CFG) and auxiliary supervision (with or without predicate-argument hints). Each axis represents a different feature layer (temporality, aspect, number, definiteness, quantification), allowing for a detailed comparison of model performance across linguistic competencies.

SMATCHY-PA scores remain below 1 even when supervision is provided, as some models still fail to produce fully valid structures.

Second, the JSON linearization generally yields better results for most models in the absence of supervision. However, for the best-performing models (Qwen3-Thinking and LLaMA4 Maverick) as well as the lowest-performing one (Mixtral 8×7B), the CFG format benefits more strongly from auxiliary supervision. In these cases, CFG achieves performance comparable to—or even surpassing—that of JSON. For the more challenging quantification layer, supervision in the JSON setting significantly reduces performance. This suggests that, although CFG is generally harder to parse, it may allow models to make better use of the provided predicate–argument structure. This advantage likely stems from the clearer separation between structural and feature components in the CFG format.

An exception is Qwen3-Thinking: in the CFG setting, auxiliary supervision actually reduces performance. Manual inspection reveals that the model frequently copies the provided structure without adding the required feature layers, leading to invalid or incomplete representations. It also often fails to produce the output within the requested XML tags, a problem that does not occur when no predicate–argument structure is provided. This suggests that the additional information confuses the model rather than assisting it. In contrast, the same model benefits substantially from supervision in the CFG format.

The last column of Table 1 shows that models fail to produce structurally valid outputs more often when additional supervision is supplied in the JSON setting, and that this trend is observed for every model tested. The effect is especially pronounced for the Qwen3 family model. An analysis of the outputs indicates that, in this situation, models tend to generate longer thinking traces, and instruction forgetting occurs. We hypothesize that this difference between the CFG and JSON settings is due to the intertwining of the predicate and layer elements in the JSON linearization. In contrast, CFG makes this distinction explicit, allowing the additional supervision to be used directly.

4.2. Layerwise Observations

Figure 4 presents feature-level performance, controlling for both the prompt format and the supervision condition.

Performance varies considerably across feature layers. Although no single performance profile emerges, temporality consistently achieves the highest scores across all models, with stable F1 scores around 0.7. Negation and definiteness are also handled reasonably well by the strongest models. These layers exhibit the least variation across hint settings and often yield better parsing accuracy than the predicate–argument structure when no assistance is provided. This trend likely reflects the fact that such features can often be identified directly from surface cues in the sentence.

In contrast, abstract features like quantification are poorly captured across all models. Even the best-performing systems fail to surpass an F1 score of 0.5 on this layer. This underscores a persistent limitation of current LLMs in modeling non-local or scope-sensitive phenomena, which cannot be reliably inferred from surface forms alone.

These findings suggest that LLMs are able to align words in the original sentence with corresponding concepts in the semantic parse and to refine these concepts using annotation layers informed by surface cues. However, their ability to perform deeper, more abstract semantic analysis remains limited.

4.3. Model comparison

The clear winner in terms of adhering to structural constraints and producing valid representations is the LLaMA4 Maverick model. It achieves the best overall performance, with Qwen3-Thinking performing comparably—or even slightly better—in most settings, except for the JSON linearization with supervision. Interestingly, Qwen3-Thinking surpasses or matches LLaMA4 Maverick on the quantification layer despite producing fewer valid parses. This contrast reveals distinct model profiles: Qwen3-Thinking tends to generate higher-quality parses when it strictly follows annotation guidelines, whereas LLaMA4 Maverick produces less accurate but more consistent outputs.

LLaMA4 Scout and Qwen3 (in non-Thinking mode) exhibit a noteworthy pattern, showing a pronounced drop in performance when switching to the CFG format.

5. Conclusion

We presented an evaluation framework for assessing the semantic parsing capabilities of LLMs using the YARN formalism, which separates predicate–argument structure from higher-level linguistic features. Our experiments evaluated multiple models across two linearization formats, both with and without auxiliary supervision, and analyzed overall as well as feature-specific performance using the SMATCHY metric suite.

We found that LLM performance varies considerably depending on the representation format and the presence of partial supervision. While supervision generally improves predicate–argument accuracy, its impact on feature parsing differs across models. Format compliance remains a major source of failure, and some models are even negatively affected by auxiliary input, particularly in the JSON setting. Layer-specific evaluation reveals that surface-aligned features such as temporality and negation are captured more reliably than abstract features like quantification. Overall, our findings highlight the critical importance of how semantic annotations are linearized, as the chosen format strongly influences whether LLMs can effectively leverage additional supervision or become hindered by it.

Our analysis points to a hierarchy in the types of semantic phenomena that LLMs can effectively handle. Features directly reflected in surface forms—such as temporality and number in English—are generally well captured. Event identification and predicate–argument structures are modeled with moderate success. In contrast, deeper semantic phenomena that require scope sensitivity, such as quantification, remain challenging across models. This pattern underscores a persistent gap between surface-level pattern recognition and deeper, compositional semantic understanding.

Beyond the specific task of semantic parsing, our results align with prior studies that leverage theoretical linguistic frameworks to probe and evaluate the semantic abilities of large language models (Scivetti et al., 2025; Ettinger et al., 2023; Ettinger, 2020)

Overall, while LLMs show progress in “learning to write semantics correctly”—with recent models like

LLaMA 4 producing a higher proportion of structurally valid outputs—effective semantic parsing through general-purpose transfer learning still appears out of reach. For the time being, specialized, fine-tuned smaller models remain the more reliable option for automated semantic parsing.

While sensitivity to how semantics is represented is a meaningful result in itself, our evaluation inherently targets two intertwined abilities: a model's general understanding of what meaning representations mean, and its capacity to produce highly constrained formal structures. One could argue that the latter remains too underdeveloped for our approach to fully assess the former. In other words, conclusions about the linguistic competence of LLMs are limited by a broader, non-linguistic weakness in generating complex structured outputs—particularly graph-based linearizations. This supports hybrid approaches where an LLM handles general language understanding, while generation is delegated to a specialized, constraint-aware model, as in (Zhan et al., 2025) for SQL parsing.

In future work, experimental setups where models are asked to evaluate or incrementally complete existing annotations, rather than generate them from scratch, could mitigate format adherence issues and yield a more reliable measure of linguistic competence. A similar approach has been successfully employed to probe syntactic comprehension in (Warstadt et al., 2019).

Ethical considerations

This study raises no specific ethical concerns beyond those inherited from semantic representation and large language models. Potential issues relate mainly to ontological bias in meaning representations. All resources used are public and non-sensitive. We emphasize transparency in methodology and the importance of linguistic diversity to avoid underrepresentation of low-resource languages.

Limitations

The study is limited by the small dataset size and the use of English-only examples, which restricts generalization. Results depend partly on models' ability to produce well-formed outputs rather than purely on linguistic competence. Evaluations were conducted without fine-tuning, and future work should explore multilingual settings and larger, more varied datasets.

6. Bibliographical 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*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

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.

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.

Aaron Conklin, Michael Frank, and Ellie Pavlick. 2023. What do llms know about linguistic theory? *arXiv preprint arXiv:2310.01672*.

Rémi de Vergnette, Maxime Amblard, and Bruno Guillaume. 2025. Evaluation framework for layered meaning representation. In *Proceedings of the Sixth International Workshop on Designing Meaning Representations*, pages 38–48, Prague, Czechia. Association for Computational Linguistics.

Yanai Elazar, Atticus Geiger, Ellie Pavlick, and Yoav Goldberg. 2021. Measuring and improving compositional generalization in language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Gottlob Frege. 1967. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. In Jean van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, pages 1–82. Harvard University Press. Originally published in 1879.

Bruno Guillaume, Guillame Bonfante, Paul Masson, Mathieu Morey, and Guy Perrier. 2012. Grew : un outil de réécriture de graphes pour le TAL (Grew: a graph rewriting tool for NLP) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 5: Software Demonstrations*, pages 1–2, Grenoble, France. ATALA/AFCP.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Percy Liang. 2016. Learning executable semantic parsers for natural language understanding. *Commun. ACM*, 59(9):68–76.

Meta AI. 2025. The llama 4 herd: The beginning of a new era of natively multimodal intelligence. Meta AI Blog.

Mistral AI. 2023. Mixtral of experts. Blog post.

Siyana Pavlova. 2025. *Tools and methods for semantically annotated corpora*. Ph.D. thesis, Université de Lorraine.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2024. YARN is All You Knit: Encoding Multiple Semantic Phenomena with Layers. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 66–76, Torino, Italia. ELRA and ICCL.

Phillip Schneider, Manuel Klettner, Kristiina Jokinen, Elena Simperl, and Florian Matthes. 2024. Evaluating large language models in semantic parsing for conversational question answering over knowledge graphs.

Wesley Scivetti, Melissa Torgbi, Austin Blodgett, Mollie Shichman, Taylor Hudson, Claire Bonial, and Harish Tayyar Madabushi. 2025. Beyond memorization: Assessing semantic generalization in large language models using phrasal constructions.

Liang Shi, Zhengju Tang, Nan Zhang, Xiaotong Zhang, and Zhi Yang. 2025. A survey on employing large language models for text-to-sql tasks. *ACM Comput. Surv.*, 58(2).

John F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. 2024. Do LLMs overcome shortcut learning? an evaluation of shortcut challenges in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12188–12200, Miami, Florida, USA. Association for Computational Linguistics.

Zeyu Zhan, E. Haihong, and Min Song. 2025. Leveraging large language models for enhanced text-to-sql parsing. *IEEE Access*.

7. Language Resource References

Nivre, Joakim and Bosco, Cristina and Choi, Jinho and de Marneffe, Marie-Catherine and Dozat, Timothy and Farkas, Richárd and Foster, Jennifer and Ginter, Filip and Goldberg, Yoav and Hajič, Jan and Kanerva, Jenna and Laippala, Veronika and Lenci, Alessandro and Lynn, Teresa and Manning, Christopher and McDonald, Ryan and Missilä, Anna and Montemagni, Simonetta and Petrov, Slav and Pyysalo, Sampo and Silveira, Natalia and Simi, Maria and Smith, Aaron and Tsarfaty, Reut and Vincze, Veronika and Zeman, Daniel. 2015. *Universal Dependencies 1.0*. ISLRN 586-682-285-530-1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).