On the Role of Semantic Proto-roles in Semantic Analysis: What do LLMs know about agency?

Elizabeth Spaulding[1] and **Shafiuddin Rehan Ahmed**[2] and **James H. Martin**[1]
[1]University of Colorado Boulder, [2]Center for Advanced AI, Accenture
elizabeth.spaulding@colorado.edu, shafiuddin.r.ahmed@accenture.com, james.martin@colorado.edu

Abstract

Large language models (LLMs) are increasingly used in decision-making contexts, yet their ability to reason over event structure—an important component in the situational awareness needed to make complex decisions—is not well understood. By operationalizing proto-role theory, which characterizes agents via properties such as *instigation* and *volition* and patients via properties such as *change of state*, we examine the ability of LLMs to answer questions that require complex, multi-step event reasoning. Specifically, we investigate the extent to which LLMs capture semantic roles such as "agent" and "patient" through zero-shot prompts, and whether incorporating semantic proto-role labeling (SPRL) context improves semantic role labeling (SRL) performance in a zero-shot setting. We find that, while SPRL context sometimes degrades SRL accuracy in high-performing models (e.g., GPT-4o), it also uncovers an internal consistency between SPRL and SRL predictions that mirrors linguistic theory, and provides evidence that LLMs implicitly encode consistent multi-dimensional event role knowledge. Furthermore, our experiments support prior work showing that LLMs underperform human annotators in complex semantic analysis.

1 Introduction

Philosophies of ethics as far back as Aristotle (Freeland, 1985; Shaver, 1985; Weiner, 1995) define moral agents as entities which can be held responsible for their acts, and moral patients as entities which experience the effects of those acts, and are thus objects of concern. In linguistics, psychology, and neuroscience, agency and patiency are studied to understand how individuals perceive, represent, and communicate about actions and their consequences. With LLMs increasingly used for decision-making, further investigation into their capacities for moral reasoning and judgment has been called for (Jiang et al., 2025).

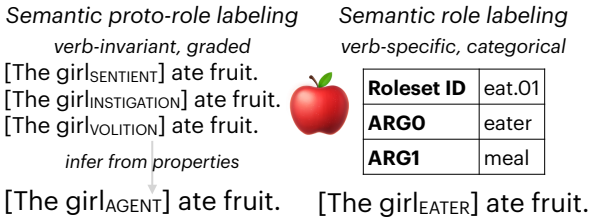

Figure 1: Semantic proto-role labeling captures broad properties across verbs, while semantic role labeling captures verb-specific role information. Evidence from neuroscience suggests that both systems are employed complementarily in human language processing.

In response, we examine LLMs through the relationship between semantic proto-role labeling (SPRL) and semantic role labeling (SRL), two sentence-level semantic tasks. Traditional SRL systems rely on fixed sets of categorical roles such as "agent", "patient," and "instrument." Proto-role theory (Dowty, 1991) challenges this view by proposing that the participants of an event are best characterized by a set of graded properties. Dowty's prototypical agent ("proto-agent") exhibits properties such as volition, sentience, causality, and movement, whereas proto-patients are more likely to undergo a change of state or be affected by the action.

Additionally, while SRL requires the entities in a sentence to be labeled with verb-specific semantic roles (agent of hitting/hitter, patient of hitting/hittee), SPRL properties are invariant across verbs. Neuroscience research has shown evidence that both verb-specific and verb-invariant event roles are encoded in the brain, and could play complementary roles in processing (Frankland and Greene, 2020).

SPRL and SRL, then, can provide a benchmark for an LLM's semantic reasoning capabilities as compared to a human's, a proxy for an LLM's situational awareness, and scaffolding for fine-grained interpretability of moral judgments. We ground our

Findings of the Association for Computational Linguistics: ACL 2025, pages 12027–12048

experiments in the following research questions: (1) does providing LLMs with semantic proto-role labels as context for semantic role labeling result in accuracy gains compared to no SPRL context?, (2) do errors in semantic role labeling correlate with errors in semantic proto-role labeling?, and (3) how does an LLM perform as the SRL and SPRL tasks increase in complexity? Our results enable us to make the following contributions:

- SPRL context often degrades SRL performance, especially in the highest-performing language model we prompted (GPT-4o), but sometimes provides a non-negligible boost to performance in smaller models.
- SRL errors do not overwhelmingly co-occur with SPRL errors, but SRL errors do overwhelmingly co-occur with SPRL properties that were deemed *not applicable* to the argument at annotation time.
- GPT-4o performs badly on the prompt variant with the most steps per test instance: an end-to-end SRL and SPRL pipeline. This suggests that the findings that GPT cannot perform complex and detailed semantic analysis of event roles (Ettinger et al., 2023; Bonn et al., 2024) generalize to larger datasets. Additionally, smaller LMs almost always performed better on less complex tasks (e.g. prompts that required only a single token for a response) versus more complex tasks (e.g. prompts that required multiple responses be produced in JSON format). However, GPT-4o performed better in a prompt variant that elicited roles for all arguments for a single predicate, versus only one argument and predicate, suggesting that some added complexity helps to further contextualize a task and boosts performance.

Additionally, while our findings did not produce evidence for positive prompt-level interactions between SRL and SPRL, we found robust evidence that GPT-4o classifies PropBank-style semantic roles consistently with Dowty's proto-role theory, suggesting that it encodes a hierarchy of event role structure similar to what has been proposed by linguistic theory. There is also faint evidence of such structures emerging in a much smaller LM, the 3b-parameter version of Llama 3.2.

2 Background and Related Work

PropBank and SRL PropBank (Kingsbury and Palmer, 2002; Gildea and Palmer, 2002; Palmer et al., 2005) is a lexical resource that separates verbs into coarse senses, each with a set of semantic roles (the "roleset," e.g. "eat.01" in Fig. 1). While the 6 core roles are verb-specific, ARG0 usually corresponds to a prototypical *agent* (the argument which makes the action happen) and ARG1 corresponds to *patient* (the argument which receives the action). Thus, these arguments can be generalized to an extent across verbs and verb senses. Semantic role labeling (SRL) is a classification task that can use PropBank as its vocabulary, so that a system learns to label sentences for their predicates, arguments, and semantic role labels. Importantly, a proficient PropBank-style SRL system can consistently label the role of verb participants across different syntactic alternations of the same verb (e.g., "I broke [the computer]$_{ARG1}$" and "[The computer]$_{ARG1}$ broke"). Various neural methods, including graph-based (Zhou et al., 2022; Liu et al., 2023) and syntax-aware (Fei et al., 2021; Zhang et al., 2022) approaches, have achieved impressive performance, and SRL has been leveraged for its promise in multimodal understanding (Sadhu et al., 2021; Bhattacharyya et al., 2023) and situation and narrative modeling (Ash et al., 2024; Balashankar et al., 2023).

SPRL Proto-role theory (Dowty, 1991) offers an alternative to categorical role inventories by focusing on the finer-grained properties of a prototypical agent (e.g., *volition*, *sentience*) and properties of a prototypical patient (e.g., *change of state*, *change of possession*). For example, in the sentence "The boy threw a rock," categorical role inventories assign argument "boy" the role Agent, and argument "rock" the role Patient. Work on decompositional semantics[1] has formulated the task of semantic proto-role labeling (SPRL) as the assignment of 14 different binary properties to arguments (Reisinger et al., 2015). Semantic proto-role labeling assigns *volition*, *sentience*, and *instigation* to "boy" and *change of state*, *change of location*, and *was used* to "rock." (All SPR labels, as well as their definitions, can be found in Table 7.) Previous work in SPRL has explored fine-tuned language models, attention-based ensembling, and other neural approaches for label classification (Teichert et al., 2017; Rudinger et al., 2018; Opitz and Frank, 2019; Tenney et al., 2019; Stengel-Eskin et al., 2020, 2021; Spaulding et al., 2023). Sadeddine et al. (2024) offer a comprehensive survey of datasets, parsers, and applications of

[1] http://decomp.io/

(a) Proto-agent properties

	instigated			volition			aware			sentience			chg-loc		
Predictions	T	F	N/A	T	F	N/A	T	F	N/A	T	F	N/A	T	F	N/A
ARG0	90.5	21.2	4.9	77.5	10.7	3.3	64.8	9.1	4.1	62.7	10.8	3.5	29.6	32.1	26.5
ARG1	6.0	58.1	50.7	15.2	66.3	51.7	23.5	69.2	52.9	25.6	65.7	49.8	53.2	47.8	42.8
ARG2	2.0	14.6	29.6	4.6	16.5	29.8	7.8	15.5	28.8	7.8	16.6	31.1	10.5	14.0	20.5
ARG3-5	0.1	1.1	2.0	0.2	1.3	2.1	0.3	1.4	2.0	0.3	1.4	2.3	0.9	1.0	1.5

(b) Proto-patient properties

	chg-poss			chg-state			created			destroyed		
Predictions	T	F	N/A	T	F	N/A	T	F	N/A	T	F	N
ARG0	7.1	33.3	27.7	8.3	36.0	32.6	0.9	35.0	11.1	5.7	34.7	11.1
ARG1	79.2	47.2	42.6	70.8	43.8	37.8	57.6	47.0	42.2	78.7	46.3	42.2
ARG2	10.9	13.7	19.6	16.2	13.7	19.3	33.6	11.8	33.3	10.1	12.7	33.3
ARG3-5	2.1	0.9	1.4	0.8	1.1	1.4	0.7	0.9	2.2	1.9	0.9	2.2

Table 1: % of predicted properties that co-occur with GPT-4o's predicted ARGn on the Ontonotes dataset, for the *All-args* prompt variant with "n/a" SPRL responses allowed. Denominator for percentages is the total number of arguments that property and value were assigned by the model. Columns do not add up to 100 because the model occasionally outputs an SRL prediction that is not any of the numbered ARGs.

tasks such as SRL and SPRL.

LLMs The largest and most successful LLMs have shown mastery in what Mahowald et al. (2024) call formal linguistic competence—the knowledge of rules and statistical regularities of language—but often fail in functional linguistic competence, or the ability to successfully apply language to real-world situations. Relevantly, functional linguistic competence includes robust situation modeling: the ability to keep track of entities, the relations between them, and their participation in various events across time. In the domain of event roles and relations, Bonn et al. (2024) assess the capabilities of GPT-3 and GPT-4 to do PropBank annotation and find that both perform far below reported human IAA (Bonial et al., 2017). Ettinger et al. (2023) find that GPT-3 and GPT-4 cannot produce AMR (Abstract Meaning Representation; Banarescu et al. 2013) annotations more complex than a core event structure in subject-verb-object form. Both studies, because of the time-intensive manual analysis required, reported results on very small sample sizes. Our study addresses similar questions at a larger sample size and provides evidence that the findings of Ettinger et al. (2023) and Bonn et al. (2024) generalize to larger data.

Previous work has found that pre-trained language models like BERT implicitly learn semantic properties, including SPRL (Tenney et al., 2019; Kuznetsov and Gurevych, 2020). Stengel-Eskin and Van Durme (2022) investigate the effect of providing SPRL context in prompts eliciting interpretations of sentences involving subject control clauses. They find that large language models contain SPR property knowledge but do not directly apply it to all situations. To date, however, there has been no fine-grained study of the capabilities of the most recent LLMs on both SPRL and SRL, and the relation between them.

3 Data

We evaluate SRL on two datasets using the standard test splits: the first one (SPR1 (Reisinger et al., 2015), $n = 1054$ predicate-argument pairs) is small, but contains SPRL annotations, which allows a fine-grained analysis of the effects of each proto-role property. The second dataset (an SRL test split from Ontonotes[2], $n = 44609$ predicate-argument pairs) is larger but lacks gold-standard SPRL annotations. We use Ontonotes to ascertain whether our analysis from SPR1 might generalize to a larger dataset with a more diverse set of topics and semantic and syntactic constructions.

SPRL results are typically reported on two English-language datasets: SPR1 and SPR2 (White et al., 2016). SPR1 contains 4,912 Wall Street Journal sentences from PropBank annotated by a single annotator based on a set of 16 proto-role properties. 9,738 arguments were annotated for the likelihood (on a Likert scale from 1 to 5) that a property holds for that argument. SPR2 contains 2,758 English Web Treebank (Bies et al., 2012) sentences annotated for a smaller set of 14 properties using a revised, streamlined protocol with two-way

[2]https://github.com/propbank/propbank-release

redundancy in annotation. The sets of properties in SPR1 and SPR2 are slightly different from one another, and neither maps one-to-one to Dowty's original 10 proto-role entailments. Table 7 shows the properties that were annotated for SPRL. In SPR2, reported inter-annotator agreement is generally acceptable for each property, with White et al. (2016) reporting a Spearman's rank correlation coefficient of 0.617 and Spaulding et al. (2023) reporting Cohen's $\kappa \geq 0.64$ for all properties when computed over the binarized labels, with an average $\kappa = 0.75$.

Previous work (Opitz and Frank, 2019; Rudinger et al., 2018; Teichert et al., 2017; Tenney et al., 2019), formulates SPRL as a 16 (SPR1) or 14 (SPR2) way multi-label binary classification problem and map Likert labels {1, 2, 3} to 0, and {4, 5} to 1. Previous work additionally maps judgments labeled "inapplicable" to 0. In our work, we use standard train/dev/test splits provided in the data where applicable: wherever we report results, those are based off of test sentences.

4 Experimental Setup

We experiment on a variety of open-weight models ranging from 3 billion to 8 billion parameters: Llama 2 and 3 (Touvron et al., 2023; Grattafiori et al., 2024), Qwen2.5 (Qwen et al., 2025), and Tülu 3 (post-trained using Llama 3.1 as a foundation; Lambert et al. 2025). We also experiment on open-weight, open-data models OLMo 1 and 2 (Groeneveld et al., 2024; OLMo et al., 2025) and Pythia (Biderman et al., 2023). Because the OLMo and Pythia authors release their data to the public, we could confirm that those models were not trained on data contaminated with the data we test on (Elazar et al., 2024). Finally, we perform a broad range of experiments that require a longer context window and a more complex array of tasks on GPT-4o (OpenAI, 2024).

We evaluate on a variety of prompt templates with and without certain context. Our main focus is the model's ability to reason over the semantics of a sentence in a zero-shot setting, and we use PropBank and proto-role theory to validate whether its reasoning is consistent. We utilize three different zero-shot prompt templates for SRL, and one prompt for SPRL alone:

- **Pipeline:** The model is sequentially prompted through six SRL components: predicate span identification, predicate sense disambiguation, argument span identification, (optionally) semantic proto-role labeling, and semantic role labeling. Previous output is provided as context in later prompts.
- **All-args-per-prompt:** The model receives oracle-provided predicate spans, sense, argument spans, and roleset details, and must output a JSON with both SPRL and SRL labels for all constituents of a single predicate.
- **One-arg-per-prompt:** The task is simplified by requiring the model to output only a single semantic role label for one argument using optional gold SPRL context.
- **SPRL-annotate:** The model is given a true/false prompt for each argument-predicate-property triplet, similar to the human annotation protocol used in SPR1/2 (Table 7).

We evaluate all prompt variants exhaustively on GPT-4o as it is a large-scale, powerful language model capable of advanced reasoning. We also evaluate smaller LMs on SPR1 using the *One-arg* prompt variant.[3] We additionally vary the context and instructions provided to the model in the following ways:

- **SPRL context:** Depending on the prompt, the model can receive SPRL context either through its previous predictions (*Pipeline*), gold labels (*One-arg*), or explicit instructions to generate SPRL output alongside SRL output (*All-args*). We additionally vary the specific SPRL properties included in each prompt: *instigation* and *change-of-location*, *volition* and *change-of-state*, all SPRL properties in both SPR1 and SPR2 (i.e., the intersection of the two property sets; SPR1∩SPR2), and all SPRL properties in SPR1.
- **Oracle predicate spans, predicate senses, and argument spans:** All prompt variants (except *Pipeline*) provide the model with oracle predicate spans, senses, and argument spans. In *Pipeline*, the model must use its own predictions.
- **Allowing "N/A" as a SPRL response:** In some All-args experiments, the model is explicitly instructed to label inapplicable properties as "n/a" rather than restricting the response to True/False.

[3]For *All-arg* and *Pipeline* prompts, due to length and complexity, early experiments on smaller LMs were computationally expensive and yielded poor results. Thus, GPT-4o is the main focus of those prompt experiments. Additionally, the token limit for the *One-arg* and *SPRL-annotate* prompts was very short for the smaller LMs for computational efficiency.

(a) F1 on SPR1 and SPR2 for *SPRL-annotate*.

	SPR1		*SPR2*	
Model	**Micro**	**Macro**	**Micro**	**Macro**
Qwen2.5 (7b) (I)	4.9	4.6	4.9	4.7
Pythia (6.9b)	5.6	5.5	16.0	15.2
Llama 2 (7b)	37.7	32.2	43.8	37.8
OLMo 2 (7b) (I)	49.1	40.1	46.1	39.6
Llama 3.1 (8b) (I)	47.2	41.8	54.0	48.2
Tulu 3 (8b)	47.4	39.8	54.9	47.6
Llama 3.2 (3b) (I)	48.2	42.1	55.6	49.3
OLMo 2 (7b)	50.8	45.9	60.7	55.1
GPT-4o	61.3	51.9	68.0	57.2
n/a skipped in eval ↓				
Qwen2.5 (7b) (I)	9.6	7.2	7.3	7.1
Pythia (6.9b)	11.1	10.5	23.4	22.2
Llama 2 (7b)	51.5	46.1	53.8	46.8
OLMo 2 (7b) (I)	70.2	58.5	58.9	51.2
Llama 3.1 (8b) (I)	71.6	65.2	68.5	62.1
Tulu 3 (8b)	67.1	59.6	68.0	60.4
Llama 3.2 (3b) (I)	72.2	64.9	70.9	63.4
OLMo 2 (7b)	81.9	74.8	79.6	72.6
GPT-4o	81.1	69.9	79.2	69.2

(b) GPT-4o F1 on SPR1. (These prompts also elicited SRL.)

	Micro-F1	**Macro-F1**
Prompt template	*SPR1, n/a = False*	
All-args	70.68	60.65
All-args, w/ PB GL	68.8	57.23
	Micro-F1	**Macro-F1**
	SPR1, n/a skipped	
All-args	78.73	64.21
All-args, w/ PB GL	79.01	64.43
All-args, n/a allowed	75.9	60.56

Table 2: SPRL results evaluated in two modes: first, collapsing "n/a" annotations to *False*, and second, discluding "n/a" annotations from the evaluation set entirely.

- **With PropBank annotation guidelines:** Since a human annotator would typically have access to detailed, PropBank-specific instructions, a variant of the All-args template includes an excerpt from the PropBank annotation guidelines as a system prompt.

Further experiment details can be found in §A.

4.1 Metrics

SRL: For *All-args* and *One-arg* variants, we use accuracy and limit our evaluation to core (numbered) roles only (ARGs 0-5). To calculate accuracy, we count a "hit" every time a model produces the correctly numbered role and divide hits by all numbered roles in the evaluation set. For *Pipeline*, we use *exact match* accuracy: predicate span, predicate sense, argument span, and role label must all be exactly correct for a hit.

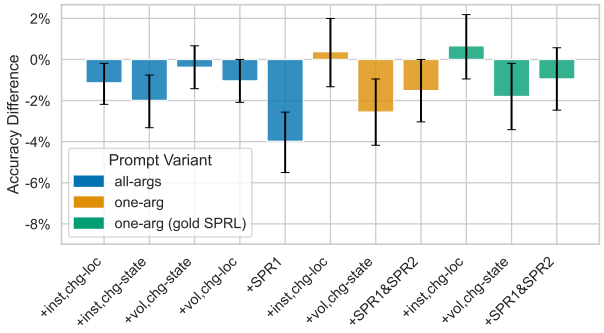

Figure 2: Effect of SPRL context on SRL accuracy, GPT-4o on SPR1. The y-axis shows the change in accuracy when providing the model with SPRL context (with SRL-alone prompts as the 0% baseline).

SPRL: We use the micro- and macro-F1 strategies in previous work: In macro-F1, F1 is computed first per property and then averaged, and in micro-F1, F1 is computed and averaged over all properties at once. We provide F1 values by (a) treating "n/a" as *False* and (b) skipping "n/a" completely (and thus, evaluating over a different set of arguments than previous work). In variants in which the model is allowed to produce "n/a" as SPRL output, the "n/a"-annotated arguments are still skipped: that is, we do not give it credit for getting the "n/a" value "correct." If the model outputs "n/a" on an annotation that is *True* or *False*, we penalize the model as if it had given the opposite value for output.

5 Results and discussion

One-arg prompt variant results on SPR1 are aggregated in Table 3. See Table 4 for GPT-4o results across all prompt variants on Ontonotes. For GPT-4o, the most effective prompt variants are *All-args* and *Pipeline (oracle)*, with the *One-arg* prompting method consistently performing the worst out of all oracle variants. The pure *Pipeline* variant, in which the model is fed its own predictions for each step of the pipeline, achieved less than 25% accuracy on both datasets. The poor *Pipeline* performance is due to errors earlier in the pipeline—see Appendix §B. This seems to further the evidence (Ettinger et al., 2023; Bonn et al., 2024) that LLMs underperform human annotators in complex, multi-step semantic analysis.

Why did the *All-args* and *Pipeline (oracle)* prompts yield better results than *One-arg*? One possible explanation is that all of the arguments being in the prompt context for the former two variants, as opposed to only one in the latter, forces the

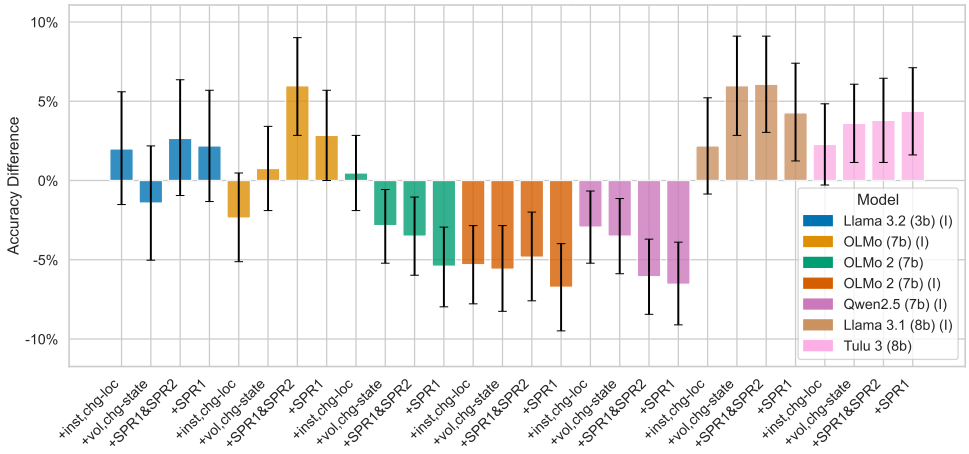

Figure 3: Effect of SPRL context on SRL accuracy, smaller LMs on SPR1.

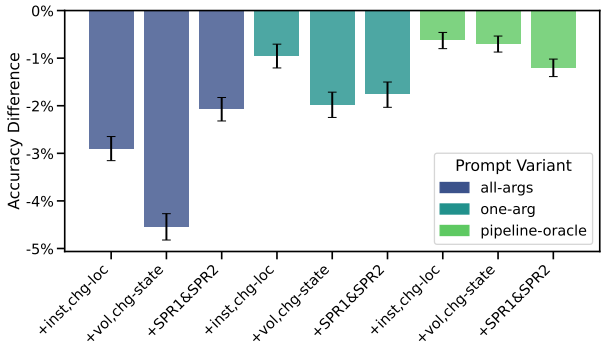

Figure 4: Effect of SPRL context on SRL accuracy, GPT-4o on Ontonotes.

model to consider all arguments at once and prevent it from assigning, for example, agent-of-action to two different arguments, but more experimentation would be necessary to isolate this effect.

The SPRL results (Table 2) contextualize the SRL results: first, when n/a is not evaluated, GPT-4o performs better on the simpler *SPRL-annotate* prompts than any of the *All-args* prompts, in which it is concurrently predicting SRL. However, in the *n/a = False* evaluation paradigm, the more complex multi-task *All-args* yields much better results.

LMs that perform well on SPRL tend to perform well on SRL, with two notable exceptions: Qwen2.5 performed the worst on SPRL while having the highest SRL accuracy out of all the other small LMs. On the other hand, Llama 3.2 (3b) exhibited competitive SPRL performance while lagging behind in SRL.

Interestingly, models tend to perform better on SPR2, *opposite* the trend in previous work, which has been explained by the SPR2 training set containing (1) less data than SPR1 or (2) more complex predicates and arguments. That zero-shot prompting methods yield better results on SPR2 than SPR1 seem to confirm that (1) was the case. In general, all zero-shot prompts we tested underperformed previous work, suggesting that fine-tuning or few-shot prompting would be a more fruitful method for utilizing LLMs for this task.

5.1 The effects of SPRL context on SRL

We plot the differences in SRL accuracy with 95% confidence intervals in Figures 2 (SPR1) and 4 (Ontonotes) for GPT-4o. These results correspond to the more detailed GPT-4o results in Tables 3 and 4. For the GPT-4o experiments on Ontonotes, a McNemar's test ($p < 0.05$) indicated a significant difference between the SRL-alone vs. SPRL-context prompt variants for all ARG0, ARG1, and ARG2 responses. The figures show that SPRL context mostly has a negative effect of SRL accuracy for GPT-4o, regardless of whether the SPRL context is in the form of gold labels (in the case of *One-arg*) or explicit instructions to generate SPRL output synchronously with SRL output (*All-args*). We additionally plot the differences in SRL accuracy with 95% confidence intervals for the smaller LMs that surpass the random baseline in Figure 3. We observe that the smaller LMs *can* benefit from SPRL context, but not consistently across models. In fact, among those that surpass the random baseline for SRL accuracy, only the Llama 3.1-based models (Llama 3.1 and Tulu 3) saw an unequivocal benefit from SPRL context. For those two models, we do see that eliciting the fuller sets of properties (SPR1 or SPR1∩SPR2) tends to result in a bigger boost.

Evaluating only well-formatted responses. A possible explanation for a drop in performance

Model	SPRL Context	Acc. (%)
Random Baseline	none	36.91
Llama 3.2 (3b) (I)	none	39.18
	instigate,chg-loc	41.18 (+2.0)
	volition-chg-state	37.76 (−1.42)
	SPR1∩ SPR2	41.84 (+2.66)
	SPR1	41.37 (+2.19)
Pythia (6.9b)	none	15.56
	instigate,chg-loc	16.7 (+1.14)
	volition-chg-state	11.67 (−3.89)
	SPR1∩ SPR2	12.71 (−2.85)
	SPR1	14.71 (−0.85)
OLMo (7b) (I)	none	45.07
	instigate,chg-loc	42.69 (−2.38)
	volition-chg-state	45.83 (+0.76)
	SPR1∩ SPR2	51.04 (+5.97)
	SPR1	47.91 (+2.84)
Llama 2 (7b)	none	25.33
	instigate,chg-loc	33.87 (+8.54)
	volition-chg-state	36.53 (+11.2)
	SPR1∩ SPR2	39.56 (+14.23)
	SPR1	37.57 (+12.24)
OLMo 2 (7b)	none	54.08
	instigate,chg-loc	54.55 (+0.47)
	volition-chg-state	51.23 (−2.85)
	SPR1∩ SPR2	50.57 (−3.51)
	SPR1	48.67 (−5.41)
OLMo 2 (7b) (I)	none	53.23
	instigate,chg-loc	47.91 (−5.32)
	volition-chg-state	47.63 (−5.6)
	SPR1∩ SPR2	48.39 (−4.84)
	SPR1	46.49 (−6.74)
Qwen2.5 (7b) (I)	none	61.01
	instigate,chg-loc	58.06 (−2.95)
	volition-chg-state	57.5 (−3.51)
	SPR1∩ SPR2	54.93 (−6.08)
	SPR1	54.46 (−6.55)
Llama 3.1 (8b) (I)	none	51.23
	instigate,chg-loc	53.42 (+2.19)
	volition-chg-state	57.21 (+5.98)
	SPR1∩ SPR2	57.31 (+6.08)
	SPR1	55.5 (+4.27)
Tulu 3 (8b)	none	59.3
	instigate,chg-loc	61.57 (+2.27)
	volition-chg-state	62.9 (+3.6)
	SPR1∩ SPR2	63.09 (+3.79)
	SPR1	63.66 (+4.36)
GPT-4o	none	87.0
	instigate,chg-loc	87.67 (+0.67)
	volition,chg-state	85.2 (−1.8)
	SPR1∩ SPR2	86.05 (−0.95)

Table 3: Effect of SPRL context on SRL accuracy across several different models on SPR1 ($n = 1054$ predicate-argument pairs) on the *One-arg* prompt variant. (I) indicates instruct-tuned.

when eliciting SPRL is the higher likelihood for the model to output a badly-formatted JSON string, due to an increased number of JSON key-value pairs the model must produce. Automatic evaluation would not capture the knowledge in a badly-formatted string. In qualitative analysis, we saw some instances of the model exhibiting correct judgment, but making a formatting error, and thus, incurring a penalty. We take a subset of responses that are well-formatted and evaluate only on those (Table 11a). However, even when evaluating on only well-formatted responses, GPT-4o mostly performs worse when asked to elicit SPRL. Interestingly, that is mostly true for Llama-3.2 (3b), except for ARG0, in which the SPRL-included prompts always perform better.

Within-model SRL-SPRL agreement. Does an LM's SRL output align with its SPRL output? That is, if an LM outputs a proto-agent SPR label, what is the probability that it concurrently outputs a proto-agent SRL label? Our findings suggest that a model's semantic role labeling decisions are principled on Dowty's proto-roles. See Table 1, which shows the likelihood of an SPRL prediction to co-occur with an SRL prediction. For example, the leftmost cell of the top table indicates that 90.5% of all arguments labeled *True* for *instigation* were also labeled ARG0, while only 6% of arguments labeled *True* for *instigation* were labeled as an ARG1. We observe the tendency of a positive response for a proto-agent property to co-occur with an ARG0 response, for a negative response for a proto-agent property to co-occur with an ARG1 response, and vice-versa for proto-patient properties (albeit with a weaker effect), suggesting the model encodes an event hierarchy similar to what is theorized in linguistics.[4]

We also observe what seems to be a very faint but similar effect even in the smallest LM prompted, Llama 3.2 (3b) (Table 9). Among the arguments it assigned numbered roles, Llama 3.2 (3b) shows a small preference for matching proto-agent properties with ARG0. (The effect all but disappears for proto-patient properties, which are classified at a much lower F1 across models.)

Explaining SRL errors with SPRL errors? We hypothesized that eliciting SPRL with SRL would help us to understand *why* a model gets an SRL example wrong. Could a concurrent error in SPRL help us pinpoint the precise dimension of the semantics that led a model to produce an SRL error?

[4]We also observe, like Reisinger et al. (2015), that Dowty's *movement* (here, *change of location*) property does not tend to correlate with ARG0.

	GPT-4o Accuracy (% correct)						
	ARG0	ARG1	ARG2	ARG3	ARG4	ARG5	All core roles
Prompt template	n = 13754	n = 22036	n = 7959	n = 451	n = 398	n = 11	n = 44609
One arg per prompt	78.94	93.14	89.33	74.72	95.98	100.0	87.93
+instigate,chg-loc	77.88 (−1.06)	93.41 (+0.27)	85.09 (−4.24)	74.5 (−0.22)	95.98 (+0.0)	100.0 (+0.0)	86.97 (−0.96)
+volition,chg-state	76.39 (−2.55)	93.61 (+0.47)	81.4 (−7.93)	74.28 (−0.44)	95.48 (−0.5)	90.91 (−9.09)	85.94 (−1.99)
+SPR1∩ SPR2	76.57 (−2.37)	94.73 (+1.59)	79.23 (−10.1)	74.28 (−0.44)	95.23 (−0.75)	90.91 (−9.09)	86.16 (−1.77)
All args per prompt	95.04	94.4	89.67	74.28	94.97	90.91	93.55
+instigate,chg-loc	95.32 (+0.28)	91.01 (−3.39)	82.17 (−7.5)	77.16 (+2.88)	94.22 (−0.75)	90.91 (+0.0)	90.65 (−2.9)
+volition,chg-state	94.69 (−0.35)	89.48 (−4.92)	78.18 (−11.49)	78.49 (+4.21)	94.22 (−0.75)	90.91 (+0.0)	89.0 (−4.55)
+SPR1∩ SPR2	94.73 (−0.31)	92.14 (−2.26)	84.62 (−5.05)	78.94 (+4.66)	94.72 (−0.25)	81.82 (−9.09)	91.48 (−2.07)
+SPR1∩ SPR2, allowing N/A	95.0 (−0.04)	90.83 (−3.57)	81.28 (−8.29)	75.61 (−1.55)	93.72 (−1.25)	90.91 (+0.0)	90.28 (−3.27)
Pipeline (oracle)	94.44	95.97	89.57	77.16	95.73	72.73	94.16
+instigate,chg-loc	93.78 (−0.66)	96.13 (+0.16)	86.74 (−2.83)	78.71 (+1.55)	93.47 (−2.26)	90.91 (+18.18)	93.53 (−0.63)
+volition,chg-state	93.31 (−1.13)	96.28 (+0.31)	86.77 (−2.8)	76.5 (−0.66)	95.48 (−0.25)	72.73 (+0.0)	93.45 (−0.71)
+SPR1∩ SPR2	93.31 (−1.13)	96.28 (+0.31)	84.02 (−5.55)	78.27 (+1.11)	91.96 (−3.77)	81.82 (+9.09)	92.95 (−1.21)

Table 4: Effect of eliciting SPRL along with SRL across prompt variants on the Ontonotes dataset. Columns are subsets of the dataset, stratified by the gold argument label. Results are reported as accuracy, i.e., the number of times the model correctly labels an ARG0 as ARG0, divided by the total number of ARG0s in the dataset.

We observe a surprising lack of error agreement: only 56% of all SRL errors co-occur with any SPRL error at all (Table 5a). The majority of SRL errors cannot be explained by an error in a majority of the SPRL properties. Notably, the property with the highest percentage of co-occurring error, *manipulated by another*, can potentially be explained by idiosyncratic annotation of that property in SPR1.[5]

While SPRL errors do not overwhelmingly co-occur with SRL errors, SPRL annotations that are marked as *not applicable* to the sentence *do* overwhelmingly co-occur: (Table 5b). For args 3-5, this correlation is understandable: 12 out of 12 of those arguments are prepositional phrases (e.g. "The difference in yield ... *widened* [to more than 5.5 percentage points]$_{ARG4}$") for which none of the definitions in Table 7 make sense. Similarly, the ARG2 errors are also often prepositions, adverbs, and numerical values or proportions (" ... retail sales *grew* [0.5%]$_{ARG2}$"). Among all errors, 37% of the arguments were prepositional or adverbial phrases. Of all the arguments with a gold patient/ARG1 role that GPT-4o missed, 50% were the subject of the sentence (normally where an agent would be).

5.2 Qualitative error analysis

While SPRL context does not overwhelmingly help the model in SRL classification, it allows a finer-grained qualitative error analysis on the model's semantic reasoning. We identify cases in which, even though the model's SRL output is in error, its SPRL output is consistent with its SRL decision, and can help explain the SRL decision with greater clarity.

(1) [They]$_{ARG1}$ would **break**, the wine would spill out, and the wineskins would be ruined.

(1-GPT) [They]$_{ARG0'}$ would **break**, ...

Sentence 1 contains an intransitive (unaccusative) alternation of the verb *break*: in this instance of the verb, there is no agent. In all prompt variants without SPRL context, GPT missed the syntactic cues indicating the lack of an agent, and it mistakenly classified the patient as an agent. Only prompts with SPRL context correctly classified "They" as the patient of the *break* verb.

(2) At Jefferies' trading room on Finsbury Circus, a stately circle at the edge of the financial district, [desktop computer screens]$_{ARG2}$ **displayed** the London market's major barometer–the Financial Times-Stock Exchange 100 Share Index.

(2-GPT) ... [desktop computer screens]$_{ARG0'}$ **displayed** the London market's major barometer...

In Sentence 2, ARG2 was wrongly classified in every single prompt variant, regardless of the context. Mostly, the model misclassified it as ARG0, the agent of the display action, even in variants where the model responds *False* for several proto-agent properties (*instigation*, *volition*, *awareness*, *sentience*) and *True* for several proto-patient properties (*stationary*, *manipulated by another*). One

[5] See White et al. (2016) for a discussion of the idiosyncrasy of the SPR1 annotator's *manipulated by another* responses.

SPRL errors	SRL errors *n = 100*
instigation	7.0% (7)
volition	3.0% (3)
awareness	5.0% (5)
sentient	3.0% (3)
change of location	4.0% (4)
exists as physical	4.0% (4)
existed before	18.0% (18)
existed during	2.0% (2)
existed after	11.0% (11)
changes poss	2.0% (2)
change of state	17.0% (17)
stationary	13.0% (13)
location of event	7.0% (7)
makes phys contact	6.0% (6)
manip by another	31.0% (31)
pred changed arg	13.0% (13)
Any of the above	56.0% (56)

(a) % of SRL errors that co-occur with SPRL errors.

	GPT-4o Error on:				
Annotation = N/A:	ARG0	ARG1	ARG2	ARG3-5	All core roles
SPRL Property	n = 24	n = 40	n = 24	n = 12	n = 100
instigation	25.0% (6)	67.5% (27)	75.0% (18)	100.0% (12)	63.0% (63)
volition	20.83% (5)	70.0% (28)	95.83% (23)	100.0% (12)	68.0% (68)
awareness	20.83% (5)	62.5% (25)	95.83% (23)	100.0% (12)	65.0% (65)
sentient	58.33% (14)	77.5% (31)	100.0% (24)	100.0% (12)	81.0% (81)
change of location	54.17% (13)	67.5% (27)	91.67% (22)	100.0% (12)	74.0% (74)
exists as physical	45.83% (11)	62.5% (25)	87.5% (21)	100.0% (12)	69.0% (69)
existed before	0.0% (0)	7.5% (3)	37.5% (9)	66.67% (8)	20.0% (20)
existed during	0.0% (0)	7.5% (3)	33.33% (8)	66.67% (8)	19.0% (19)
existed after	0.0% (0)	15.0% (6)	54.17% (13)	66.67% (8)	27.0% (27)
changes possession	58.33% (14)	82.5% (33)	91.67% (22)	91.67% (11)	80.0% (80)
change of state	0.0% (0)	12.5% (5)	50.0% (12)	100.0% (12)	29.0% (29)
stationary	54.17% (13)	67.5% (27)	95.83% (23)	100.0% (12)	75.0% (75)
location of event	79.17% (19)	80.0% (32)	100.0% (24)	100.0% (12)	87.0% (87)
makes phys contact	50.0% (12)	65.0% (26)	95.83% (23)	100.0% (12)	73.0% (73)
manip by another	70.83% (17)	42.5% (17)	20.83% (5)	25.0% (3)	42.0% (42)
pred changed arg	8.33% (2)	27.5% (11)	75.0% (18)	91.67% (11)	42.0% (42)
Any of the above	100.0% (24)	100.0% (40)	100.0% (24)	100.0% (12)	100.0% (100)

(b) % of errors that co-occur with a *not applicable* property.

Table 5: Co-occurrence of GPT-4o's SRL errors with (5a) SPRL errors elicited in the same prompt, and with (5b) arguments that have been annotated as *not applicable* for that SPRL property. From the *All-args* prompt variant on SPR1, with SPRL not evaluated when the annotation is "n/a.".

possible explanation can still be provided by proto-role properties: several models responded *False* for the *location of event* proto-role property, and the PropBank description for ARG2 shown to the model is simply "location."

(3) [Richard Chamberlain]$_{ARG0}$ **dresses** as a "Mainland haole," tucking in a Hawaiian shirt and rolling up its long sleeves.

(3-GPT) [Richard Chamberlain]$_{ARG1'}$ **dresses** as a "Mainland haole," ...

Sentence 3 gives us another example in which proto-role properties seem to have helped GPT make an SRL decision: in the prompt variants without SPRL, the model misclassified ARG0 as ARG1 (proto-patient, the one wearing clothes) in the reflexive "dresses". However, in the variants in which SPRL were elicited, the model correctly classified ARG0 as well as assigning *True* to strongly proto-agent properties *instigation*, *volition*, *awareness*, and *sentience*, suggesting that the SPRL context encouraged the model to reason more deeply over the event roles.

6 Conclusion and future work

Our study investigated how LLMs reason about agency and patiency by examining the interplay between semantic proto-role labeling (SPRL) and semantic role labeling (SRL). We found that including SPRL context in prompts does not improve SRL accuracy, and in some cases even degrades performance. However, our experiments revealed a consistency between SPRL and SRL LLM output that mirrors Dowty's proto-role theory. In particular, we tested two models, GPT-4o and Llama 3 (3b), on synchronous SRL and SPRL judgments, and found that they both tend to assign proto-agent properties (e.g., instigation, volition) alongside ARG0 labels, suggesting that the latent event role encodings of these models align with linguistic theory.

Additionally, our experiments prompting GPT-4o through a full SRL-SPRL pipeline provide further evidence that LLMs are not yet capable of complex, multi-step semantic reasoning needed to annotate sentences for rich event structure (Ettinger et al., 2023; Bonn et al., 2024), suggesting that LLMs deployed in real-world settings may lack the advanced situation-modeling capabilities necessary to make complex decisions.

However, our results show that LLMs *do* encode something like a latent event role hierarchy, and we suspect that there are better ways to coax the models to employ it. Future research should explore, for example, few-shot learning or fine-tuning strategies to better harness LLMs' event role knowledge. Such improvements could not only improve semantic role labeling accuracy, but also deepen our understanding of how LLMs model event structure: a necessary component in developing robust AI systems with transparent situation-modeling and decision-making capabilities.

Limitations

Our study was limited in its focus on English. The models and data evaluated are both either all-English, or highly English-centric, and as such, our claims cannot extend across languages. The annotation schemes of the datasets also may not generalize well to other domains, limiting the generalizability of our findings. Fruitful future work could focus its efforts on evaluating multi-lingual LLMs on verb lexica in different languages.

Because small changes in wording or formatting could lead to different performance outcomes, our results are also influenced by the specific prompt formulations we used. While we attempted to capture a broad swath of prompts, much more experimentation needs to be done to understand what prompt-based methods truly can reveal regarding event role knowledge in LLMs.

We also recognize the possibility of data contamination (i.e. that the test sets we evaluated on were included as training data for an LLM), potentially inflating performance due to memorization rather than genuine semantic reasoning. We attempted to mitigate this risk by including open-data models in our analysis, but we cannot entirely rule out the possibility of contamination in models like GPT-4o. Nevertheless, we included GPT-4o and other closed-data models because of their ubiquity in real-world applications. Future work should consider evaluating on entirely novel data to remove this risk.

Ethical Considerations

Finally, we want to emphasize the risk of overestimation of LLM capabilities. If those interested in deploying LLMs interpret our results as evidence that LLMs truly "understand" agency or moral responsibility, they may use these models to support or automate decisions that have significant ethical implications. Such misuse could exacerbate issues like bias in real-world settings. We reiterate that our results (especially in the *Pipeline* setting) support the finding that language models are not yet capable of complex semantic analysis.

Acknowledgments

We thank the ARR reviewers for their helpful feedback and suggestions. We also gratefully acknowledge the support of the Center for Computational Language and Education Research (CLEAR) at the University of Colorado Boulder.

References

Elliott Ash, Germain Gauthier, and Philine Widmer. 2024. Relatio: Text semantics capture political and economic narratives. *Political Analysis*, 32(1):115–132.

Ananth Balashankar, Lakshminarayanan Subramanian, and Samuel P. Fraiberger. 2023. Predicting food crises using news streams. *Science Advances*, 9(9):eabm3449.

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.

Abhidip Bhattacharyya, Martha Palmer, and Christoffer Heckman. 2023. CRAPES:cross-modal annotation projection for visual semantic role labeling. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 61–70, Toronto, Canada. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank.

Claire Bonial, Kathryn Conger, Jena D. Hwang, Aous Mansouri, Yahya Aseri, Julia Bonn, Timothy O'Gorman, and Martha Palmer. 2017. Current directions in english and arabic propbank.

Julia Bonn, Harish Tayyar Madabushi, Jena D. Hwang, and Claire Bonial. 2024. Adjudicating LLMs as PropBank adjudicators. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 112–123, Torino, Italia. ELRA and ICCL.

David Dowty. 1991. Thematic Proto-Roles and Argument Selection. *Language*, 67(3):547–619. Publisher: Linguistic Society of America.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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.

Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. 2021. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 549–559, Online. Association for Computational Linguistics.

Steven M Frankland and Joshua D Greene. 2020. Two ways to build a thought: Distinct forms of compositional semantic representation across brain regions. *Cerebral Cortex*, 30(6):3838–3855.

Cynthia A. Freeland. 1985. Aristotelian actions. *Noûs*, 19(3):397–414.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 239–246, USA. Association for Computational Linguistics.

Aaron Grattafiori et al. 2024. The llama 3 herd of models. (arXiv:2407.21783). ArXiv:2407.21783 [cs].

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Liwei Jiang, Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jenny T. Liang, Sydney Levine, Jesse Dodge, Keisuke Sakaguchi, Maxwell Forbes, Jack Hessel, Jon Borchardt, Taylor Sorensen, Saadia Gabriel, Yulia Tsvetkov, Oren Etzioni, Maarten Sap, Regina Rini, and Yejin Choi. 2025. Investigating machine moral judgement through the delphi experiment. *Nature Machine Intelligence*, 7(1):145–160.

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).

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2025. Tulu 3: Pushing frontiers in open language model post-training. (arXiv:2411.15124). ArXiv:2411.15124 [cs].

Wei Liu, Songlin Yang, and Kewei Tu. 2023. Structured mean-field variational inference for higher-order span-based semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 918–931, Toronto, Canada. Association for Computational Linguistics.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 olmo 2 furious. (arXiv:2501.00656). ArXiv:2501.00656 [cs].

OpenAI. 2024. Gpt-4 technical report. (arXiv:2303.08774). ArXiv:2303.08774 [cs].

Juri Opitz and Anette Frank. 2019. An argument-marker model for syntax-agnostic proto-role labeling. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 224–234, Minneapolis, Minnesota. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. (arXiv:2412.15115). ArXiv:2412.15115 [cs].

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Rachel Rudinger, Adam Teichert, Ryan Culkin, Sheng Zhang, and Benjamin Van Durme. 2018. Neural-Davidsonian semantic proto-role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 944–955, Brussels, Belgium. Association for Computational Linguistics.

Zacchary Sadeddine, Juri Opitz, and Fabian Suchanek. 2024. A survey of meaning representations – from theory to practical utility. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2877–2892, Mexico City, Mexico. Association for Computational Linguistics.

Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. 2021. Visual semantic role labeling for video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5589–5600.

Kelly G. Shaver. 1985. *The Attribution of Blame*. Springer, New York, NY.

Elizabeth Spaulding, Gary Kazantsev, and Mark Dredze. 2023. Joint end-to-end semantic proto-role labeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 723–736, Toronto, Canada. Association for Computational Linguistics.

Elias Stengel-Eskin, Kenton Murray, Sheng Zhang, Aaron Steven White, and Benjamin Van Durme. 2021. Joint universal syntactic and semantic parsing. *Transactions of the Association for Computational Linguistics*, 9:756–773.

Elias Stengel-Eskin and Benjamin Van Durme. 2022. The curious case of control. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11065–11076, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Elias Stengel-Eskin, Aaron Steven White, Sheng Zhang, and Benjamin Van Durme. 2020. Universal decompositional semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8427–8439, Online. Association for Computational Linguistics.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew Gormley. 2017. Semantic proto-role labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. (arXiv:2307.09288). ArXiv:2307.09288 [cs].

Bernard Weiner. 1995. *Judgments of responsibility: A foundation for a theory of social conduct.* Judgments of responsibility: A foundation for a theory of social conduct. Guilford Press, New York, NY, US.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Yu Zhang, Qingrong Xia, Shilin Zhou, Yong Jiang, Guohong Fu, and Min Zhang. 2022. Semantic role labeling as dependency parsing: Exploring latent tree structures inside arguments. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4212–4227, Gyeongju, Republic of

Korea. International Committee on Computational Linguistics.

Shilin Zhou, Qingrong Xia, Zhenghua Li, Yu Zhang, Yu Hong, and Min Zhang. 2022. Fast and accurate end-to-end span-based semantic role labeling as word-based graph parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4160–4171, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

A Experiment details

This section provides precise details for each prompting experiment, exact model names, and specific parameters used when prompts were run. Exact model configurations, prompts, and output provided upon request, and samples can be found in the released code used for the experiments.

A.1 Prompt variants

See Table 6 for examples of each prompt variant.

Pipeline: The model is prompted to provide output for every single component in a six-component SRL pipeline: predicate span identification, predicate sense disambiguation, argument span identification, (optionally) semantic proto-role labeling, and semantic role labeling. Each component is separately processed via a prompt which includes, as context, the prompt and output for the previous component. At the end of the pipeline, the model has access to the full pipeline starting from the raw sentence, including all of the previous predicate span, argument span, and SPRL output, but only has to label one argument per prompt. See further details on this prompt variant in A.2 and an example in Figure 7.

All-args-per-prompt: The model is prompted to provide a JSON with the SPRL and SRL output for all constituents of a single predicate. The predicate spans, predicate sense, argument spans, and roleset information are all provided to the model by an oracle: its only task is to select the correct semantic role label (and/or semantic proto-role label). A sample prompt for this variant can be found in Figure 6. This prompt will allow us to study the effect of *eliciting* SPRL context from the model itself, without providing gold SPR labels.

One-arg-per-prompt: To evaluate the model on a prompt that requires a simpler output format than a JSON-formatted string, this prompt only elicits the semantic role label for a single argument. SPRL is elicited in a separate prompt from SRL, and, in settings where SPRL is included, it is presented as optional but helpful context. See Figure 5 for an example.

We additionally provide a random baseline for the *One-arg* variant, in which the semantic role label is randomly chosen from the set of all numbered argument labels in the training set, weighted by the proportion each is found in the training set (so that the random choices are proportional with the labels in the training set).

SPRL-annotate: We evaluate all models on a simple true/false prompt, designed to be similar in format to the questions posed to the SPR1 and SPR2 annotators (Table 7). The models are only prompted on one argument-predicate-property triplet at a time, as in the human annotation protocol. Figure 8 shows an example of this prompt.

Prompt name	Prompt example
One-arg-per-prompt	Figure 5
All-args-per-prompt	Figure 6
Pipeline	Figure 7
SPRL-annotate	Figure 8

Table 6: Prompt index

A.2 Components of SRL pipeline

The pipeline prompts are separated into six separate queries to the model, listed below. Each query's output is then chained to the beginning of the next prompt as additional context for the model. In the *Pipeline (oracle)* setting, items 1-4 are not processed through the model. Instead, we process items 1-4 with gold data as an oracle, and chain the prompts as in the non-oracle setting.

1. Predicate span retrieval
   Input: `The dog barked.`
   Output: `The dog <PRED>barked</PRED>.`
2. Roleset identification **given predicate span** from predicate output
   Input: `The dog <PRED>barked</PRED>.` + *a list of rolesets for lemma "bark" and their descriptions*
   Output: `bark.01`
3. Argument span retrieval **given predicate span** from predicate output
   Input: `The dog <PRED>barked</PRED>.`
   Output: `<ARG>The dog</ARG> <PRED> barked</PRED>.`

4. Semantic proto-role labeling **given argument and predicate span** from argument output

   Input: `<ARG>The dog</ARG> <PRED> barked</PRED>.` + *a list of semantic proto-role properties and their descriptions*

   Output: *A JSON string with the listed semantic proto-role properties as keys and booleans as values*

5. Semantic role labeling **given a roleset, argument span, and predicate span** from argument output

   Input: `<ARG>The dog</ARG> <PRED> barked</PRED>.` + *a list of roles and role descriptions for bark.01*

   Output: `ARG0`

6. Semantic role labeling **given SPRL, roleset, argument span, and predicate span** from SPRL output

   Input: `<ARG>The dog</ARG> <PRED> barked</PRED>.` + *a list of roles and role descriptions for bark.01* + *a JSON string with semantic proto-role properties as keys and booleans as values*

   Output: `ARG0`

A.3 Prompt context

SPRL context: SPRL context is provided to the model in various ways, depending on the prompt template: in *Pipeline*, the model has access to its previous predictions; in *One-arg*, the model is given SPRL context from gold labels; and in *All-args*, the model is given instructions to produce SPRL output concurrently with SRL output. In every prompt variant, we ablate SPRL context to ascertain the effect of the context on the model's decision-making.

Oracle predicate spans, predicate senses, and argument spans: All prompt template variants provide the model with oracle predicate spans, predicate senses, and argument spans except *Pipeline*. And even in *Pipeline*, the model has access to all of its previous predicate span, predicate sense, and argument span predictions at the end of the pipeline, when the model is finally prompted to make its SRL prediction. None of our experiments prompt the model to make concurrent predicate span, predicate sense, and argument span predictions along with SPRL and SRL predictions.

Allowing "N/A" as a SPRL response: A potential weakness of SPRL modeling is the handling of *not applicable* annotations. In most previous work, properties annotated *not applicable* are converted to *False* in the data, and systems are evaluated on their abilities to classify those instances as *False*. For some experiments in the *All-args* prompt variant, we introduce an additional variant in which the model is explicitly instructed to mark properties as "n/a" if that property doesn't apply to the argument. And, while SPRL performance is not our main concern, when we evaluate, we skip over properties that are marked "n/a" in the annotations.

With PropBank annotation guidelines: An additional concern in evaluating the model on its ability to produce PropBank-style SRL is the lack of detailed, PropBank-specific instruction that a human annotator would have access to, especially because the information found in the PropBank rolesets presented to the model can appear cryptic to a non-expert. While we think it is likely that GPT-4o's training data included public resources such as the PropBank annotation guidelines, we include in our experiments a variant on the *All-args* template that includes a relevant excerpt of the PropBank annotation guidelines[6] as a system prompt. Specifically, we include Section 1.1, up to and including Table 1.1, and Section 1.3.1, on choosing ARG0 vs. ARG1.

A.4 Parameters and model details

We use `gpt-4o-2024-05-13` for all GPT experiments. For the rest of the experiments, we access models through HuggingFace, using the following model codes:

- meta-llama/Llama-2-7b-hf-Instruct
- meta-llama/Llama-2-7b-hf
- meta-llama/Llama-3.1-8B-Instruct
- meta-llama/Llama-3.2-3B-Instruct
- allenai/OLMo-2-1124-7B-Instruct
- allenai/OLMo-2-1124-7B
- allenai/OLMo-7b-Instruct-hf
- EleutherAI/pythia-6.9b
- Qwen/Qwen2.5-7B-Instruct
- allenai/Llama-3.1-Tulu-3-8B

[6] https://github.com/propbank/propbank-documentation

We instantiate each model according to their default configurations—the only parameter we change is the `max_length` on generation, which we set to `model_max_position_embeddings + 4`. Runtimes varied between experiments (number of parameters, prompt length, and number of samples in the datasets all affecting the runtime), but the experiments on the models with the most parameters and the largest datasets took ∼20 GPU hours to run on a single NVIDIA TITAN XP or NVIDIA TITAN RTX and the experiments on the models with the fewest parameters took as little as ∼20 minutes to run. Exact runtimes per experiment can be provided on request.

B Pipeline performance

The *Pipeline* prompt variant exhibited perhaps shockingly low accuracy because of the strict nature of the evaluation metrics for the span retrieval components within the SRL pipeline. A semantic role label is only counted correct if the model is able to achieve an *exact match* on every single component: retrieval of the *exact* predicate span, classification of the predicate sense, and retrieval of the *exact* argument span, as well as the correct role label. Below is the gold span annotation (for the argument span retrieval step, before being prompted for the role label) for a test sentence:

(4) To express its determination, [the Chinese securities regulatory department]$_{ARG}$ **compares** [this stock reform]$_{ARG}$ [to a die that has been cast]$_{ARG}$.

Compare the gold annotation to the output from GPT-4o:

(4-GPT) [To express its determination, the Chinese securities regulatory department]$_{ARG'}$ **compares** [this stock reform]$_{ARG}$ to [a die that has been cast]$_{ARG'}$.

Out of the three arguments in the gold annotation, GPT only successfully produced the exact span for one of them—a 33% accuracy. Table 8 provides a fine-grained breakdown of the accuracy of the GPT-4o output on the *Pipeline* prompt variant, contextualizing the very low accuracy shown in Table 4.

Because the model was only able to correctly retrieve 26% of the argument spans in the Ontonotes dataset, it did not even see the other 74% and was automatically penalized for them, which is why the accuracy was so low. Although these experiments show a very low performance overall in the span retrieval tasks, we suspect that GPT-4o is capable of better performance. For instance, trying few-shot approaches, changing the potentially-confusing HTML-style <PRED> and <ARG> tags, or trying a different style of prompt completely could yield better results.

C Full results

We provide results on all model runs, broken down per argument, in Table 10 for GPT-4o and Llama-3.2 (3b).

C.1 Non-compliance with formatting requirements

We perform some evaluations in order to disentangle the effects of non-compliance with the formatting requirements specified by the prompt (i.e., a model failing to correctly format its response). For the *All-args* prompt template, see Table 11 shows accuracy on *only* parsable output. For the 3b-parameter Llama-3.2 output, we additionally perform "fuzzy role matching," which allows "agent" and "patient" as correct responses for ARG0 and ARG1, respectively, because of the large number of responses that used those words instead of the required ARG0 and/or ARG1. As such, we are able to evaluate the actual knowledge within Llama without giving it undue penalty for minor formatting mistakes. We see that the model performs far better under the fuzzy role matching evaluation method.

D Licenses

All data used for prompting and evaluation is under a CC BY-SA 4.0 license. All work was consistent with the intended use of data and models, which have the following licenses and terms: Apache 2.0 for Qwen2.5, OLMo 1 and 2, and Pythia. Llama Community License for Llama 2 and 3 (details here). OpenAI terms of use (here) for GPT-4o. Tulu 3 is subject to the licenses of the models used to train it: Llama Community License, Gemma Terms of Use, and Qwen License Agreement (details here).

Prompt	Identify the Semantic Role for the predicate <PRED>put</PRED> and arg <ARG>We</ARG>. Below are the Semantic Roles for <PRED>put</PRED>: ARG0: putter. ARG1: thing put. ARG2: attribute of arg1. Respond with only the role label, and nothing else, like so: Example: <ARG>The dog</ARG> <PRED>barked</PRED>. Response: ARG0 Text for labeling: <ARG>We</ARG> <PRED>put</PRED> some orders together . Response:
Response	ARG0

(a) SRL only, *one-arg* prompt variant.

Prompt	Identify the Semantic Role for the predicate <PRED>put</PRED> and arg <ARG>We</ARG>. Consider the fact that <ARG>We</ARG> has the following properties in making your decision: instigation: ARG caused PRED to happen. change_of_location: ARG changed location during PRED. Below are the Semantic Roles for <PRED>put</PRED>: ARG0: putter. ARG1: thing put. ARG2: attribute of arg1. Respond with only the role label, and nothing else, like so: Example: <ARG>The dog</ARG> <PRED>barked</PRED>. Response: ARG0 Text for labeling: <ARG>We</ARG> <PRED>put</PRED> some orders together .
Response	ARG0

(b) SRL with SPRL context, *one-arg* prompt variant.

Figure 5: *One-arg-per-prompt* template variant. This prompt asks only for the model to pick a semantic role out of a list of semantic roles and provides the SPRL context from the model's silver predictions, but not in a pipeline context. It only asks for one argument at a time.

Prompt

```
Identify the Semantic Roles for the arg(s) <ARG>We</ARG>, <ARG>some orders</ARG>, <ARG>together</ARG> with respect to the predicate
<PRED>put</PRED> in the following sentence:

<ARG>We</ARG> <PRED>put</PRED> <ARG>some orders</ARG> <ARG>together</ARG> .


1. Analyze each Argument for Semantic Proto-Role Properties. Below are the properties:
volition: ARG chose to be involved in PRED. true/false
change_of_location: ARG changed location during PRED. true/false


2. Label each Argument for its Semantic Role. Below are the Semantic Roles for <PRED>put</PRED>:
ARG0: putter.
ARG1: thing put.
ARG2: attribute of arg1.


JSON Format:
{
    "text": "We put some orders together .",
    "predicates": {
        "put": {
            "start_char": 3,
            "arguments": {
                "We": {
                    "start_char": 0,
                    "volition": true/false,
                    "change_of_location": true/false,
                    "role": "Identified Semantic Role Label for <ARG>We</ARG>"

                },"some orders": {
                    "start_char": 7,
                    "volition": true/false,
                    "change_of_location": true/false,
                    "role": "Identified Semantic Role Label for <ARG>some orders</ARG>"

                },"together": {
                    "start_char": 19,
                    "volition": true/false,
                    "change_of_location": true/false,
                    "role": "Identified Semantic Role Label for <ARG>together</ARG>"

                }
            }
        }

    }
}


Original text for labeling:

We put some orders together .

Output only the JSON.
```

Response

```
```json{"text": "...", "predicates": {"put": {"start_char": 3, "arguments": {
  "volition": true,
  "change_of_location": false,
  "role": "ARG0" ...
}}```
```

(a) *All-args* prompt variant, eliciting both SRL and SPRL at the same time.

Prompt

```
Identify the Semantic Roles for the arg(s) <ARG>We</ARG>, <ARG>some orders</ARG>, <ARG>together</ARG> with respect to
the predicate <PRED>put</PRED> in the following sentence:

<ARG>We</ARG> <PRED>put</PRED> <ARG>some orders</ARG> <ARG>together</ARG> .


Below are the possible Semantic Roles for <PRED>put</PRED>:
ARG0: putter.
ARG1: thing put.
ARG2: attribute of arg1.


JSON Format:
{
    "text": "We put some orders together .",
    "predicates": {
        "put": {
            "start_char": 3,
            "arguments": {
                "We": {
                    "start_char": 0,
                    "role": "Identified Semantic Role Label for <ARG>We</ARG>"
                },"some orders": {
                    "start_char": 7,
                    "role": "Identified Semantic Role Label for <ARG>some orders</ARG>"
                },"together": {
                    "start_char": 19,
                    "role": "Identified Semantic Role Label for <ARG>together</ARG>"
                }
            }
        }
    }
}


Original text for labeling:

We put some orders together .

Output only the JSON.
```

Response

```
```json{"text": "We put some orders together .",
        "predicates": {
          "start_char": 3, ...}
```

(b) *All-args* prompt variant, eliciting only SRL.

Figure 6: *All-args-per-prompt* template variants. This prompt asks the model to pick a semantic role concurrently with semantic proto-role labeling properties, and output them all in JSON format.

Prompt	Identify predicates with <PRED></PRED> tags on the original sentence. Format the response inline with the original sentence, and only respond with the labeled sentence, like so: Example: The dog barked. Response: The dog <PRED>barked</PRED>. Text for labeling: MANY AFFLUENT people place personal success and money above family . Response:
Response	MANY AFFLUENT people <PRED>place</PRED> personal success and money above family.

Previous prompt + response + new prompt

Prompt	Identify predicates with <PRED></PRED> tags on the original sentence. Format the response inline with the original sentence, and only respond with the labeled sentence, like so: Example: The dog barked. Response: The dog <PRED>barked</PRED>. Text for labeling: MANY AFFLUENT people place personal success and money above family . Response: MANY AFFLUENT people <PRED>place</PRED> personal success and money above family. Identify arguments for the predicate <PRED>place</PRED> with <ARG></ARG> tags on the original sentence. Format the response inline with the original sentence, and only respond with the labeled sentence, like so: Example: The dog <PRED>barked</PRED>. Response: <ARG>The dog</ARG> <PRED>barked</PRED>. Text for labeling: MANY AFFLUENT people <PRED>place</PRED> personal success and money above family. Response:
Response	<ARG>MANY AFFLUENT people</ARG> <PRED>place</PRED> <ARG>personal success and money</ARG> above family.

(a) Predicate and argument prompts.

Prompt	(previous text omitted)...</PRED> personal success and money above family. Response: <ARG>MANY AFFLUENT people</ARG> <PRED>place</PRED> <ARG>personal success and money</ARG> above family. Identify SPRL properties for the predicate <PRED>place</PRED> and arg <ARG>MANY AFFLUENT people</ARG>. Give judgements for each SPRL property below: volition: ARG chose to be involved in PRED. change_of_state: ARG was altered or somehow changed during or by the end of PRED. JSON Format: { "MANY AFFLUENT people": { "volition": true/false, "change_of_state": true/false } } Text for labeling: <ARG>MANY AFFLUENT people</ARG> <PRED>place</PRED> personal success and money above family. Output only the JSON. Response:
Response	```json{"MANY AFFLUENT people": { "volition": true, "change_of_state": false }}```

Previous prompt + response + new prompt

Prompt	(previous text omitted)...Response: ```json{ "MANY AFFLUENT people": { "volition": true, "change_of_state": false }}``` Identify the Semantic Role for the predicate <PRED>place</PRED> and arg <ARG>MANY AFFLUENT people</ARG>. Below are the Semantic Roles for <PRED>place</PRED>: ARG0: putter. ARG1: thing put. ARG2: where put. Respond with only the role label, and nothing else, like so: Example: <ARG>The dog</ARG> <PRED>barked</PRED>. Response: ARG0 Text for labeling: <ARG>MANY AFFLUENT people</ARG> <PRED>place</PRED> personal success and money above family. Response:
Response	ARG0

(b) SPRL and SRL prompts.

Figure 7: *Pipeline* prompts. The above represents a single SRL instance from start (predicate ID) to end (SRL). In the SRL-only setting, the SPRL step is skipped completely so that the prompt gives no SPRL-related context. In the *Pipeline (oracle)* setting, the predicate, roleset, and argument output is taken from the gold data as an oracle, but the prompt format is the exact same.

Prompt	TRUE or FALSE: “MANY AFFLUENT people” was altered or somehow changed during or by the end of “place” in the following sentence: [MANY AFFLUENT people] [place] personal success and money above family . Response:
Response	FALSE

Figure 8: SPRL-annotate prompt example, designed to be similar to questions posed to annotators in White et al. (2016).

Role Property	**Proto-role**	**Definition**	**Dataset**
instigation	agent	*Arg* caused *Pred* to happen	1&2
volition	agent	*Arg* chose to be involved in *Pred*	1&2
awareness	agent	*Arg* was aware of being involved in *Pred*	1&2
sentient	agent	*Arg* was sentient	1&2
change of location	agent	*Arg* changed location during *Pred*	1&2
exists as physical	agent	*Arg* existed as a physical object	1
existed before	[depends]	*Arg* existed before *Pred* began	1&2
existed during	[depends]	*Arg* existed during *Pred*	1&2
existed after	[depends]	*Arg* existed after *Pred* stopped	1&2
created*	patient	Infer from (–existed before and +existed after)	1&2
destroyed*	patient	Infer from (+existed before and –existed after)	1&2
change of possession	patient	*Arg* changed possession during *Pred*	1&2
change of state	patient	*Arg* was altered or somehow changed during or by the end of *Pred*	1&2
stationary	patient	*Arg* was stationary during *Pred*	1
location of event	peripheral	*Arg* described the location of *Pred*	1
makes physical contact	agent	*Arg* made physical contact with someone or something else involved in *Pred*	1
was used	patient	*Arg* was used in carrying out *Pred*	2
manipulated by another	patient	*Arg* was used in carrying out *Pred*	1
predicate changed argument	patient	*Pred* caused a change in *Arg*	1
was for benefit	patient	*Pred* happened for the benefit of *Arg*	2
partitive	patient	Only a part of portion of *Arg* was involved in *Pred*	2
change of state continuous	patient	The change in *Arg* happened throughout *Pred*	2

Table 7: Definitions of the proto-role properties used in the original annotations in both Reisinger et al. (2015) and White et al. (2016). These were included in zero-shot prompts. **Created* and *destroyed* were not labeled directly by annotators; instead, those labels are inferred from different combinations of the *existed [before/during/after]* labels.

	GPT-4o Accuracy (% correct)						
Prompt template	ARG0 n = 13754	ARG1 n = 22036	ARG2 n = 7959	ARG3 n = 451	ARG4 n = 398	ARG5 n = 11	All core roles n = 44609
Pipeline	31.39	22.41	12.92	8.87	6.78	0.0	23.2
+instigate,chg-loc	33.6 (+2.21)	24.34 (+1.93)	12.94 (+0.02)	9.53 (+0.66)	12.81 (+6.03)	0.0 (+0.0)	24.9 (+1.7)
+volition,chg-state	33.34 (+1.95)	24.4 (+1.99)	13.38 (+0.46)	10.2 (+1.33)	12.56 (+5.78)	0.0 (+0.0)	24.93 (+1.73)
+SPR1∩ SPR2	33.43 (+2.04)	24.46 (+2.05)	12.9 (−0.02)	9.53 (+0.66)	12.56 (+5.78)	0.0 (+0.0)	24.9 (+1.7)

(a) Effect of SPRL on SRL accuracy for GPT-4o on the full SRL-SPRL pipeline, in which the model must predict everything and is penalized on errors earlier in the pipeline.

	# preds or args evaluated			*Accuracy*	
Task	$N \in$ ***Onto***	$M \in$ ***GPT-retrieved***	**# correct**	**Full-pipeline** (out of N)	**Single task** (out of M)
Pred. span retrieval	24,167	n/a	14,022	55.72%	n/a
Pred. sense disambiguation	24,167	14,022	12,472	49.56%	88.95%
Arg. span retrieval	44,609	26,497 (=1.9 args per pred)	11,809	26.47%	44.60%
SRL	44,609	11,809	10,351	23.20%	96.96%

(b) Fine-grained SRL pipeline results on GPT-4o. The full-pipeline accuracy (what was used in Table 4) shows the score for a task, penalizing for misses in previous tasks. The single-task accuracy shows the score without penalizing the model for previous misses; i.e., the model is only scored on the subset of data that it got correct in the previous component of the pipeline.

Table 8: Performance of GPT-4o on the full SRL-SPRL pipeline.

(a) Proto-agent properties

	instigated		volition		aware		sentience		chg-loc	
	T	F	T	F	T	F	T	F	T	F
ARG0	32.9	24.4	39.4	21.5	31.9	23.9	50.0	23.7	46.7	25.0
ARG1	2.6	14.9	0.8	16.9	6.9	14.9	0.0	14.2	6.7	13.4
ARG2	0.0	3.3	0.0	3.7	1.7	3.2	0.0	3.1	0.0	2.9
ARG3-5	0.0	1.0	0.0	1.2	0.0	1.1	0.0	1.0	0.0	0.9
other	61.8	52.3	57.5	52.4	51.7	54.1	47.5	54.0	40.0	53.9

(b) Proto-patient properties

	chg-poss		chg-state		created		destroyed	
	T	F	T	F	T	F	T	F
ARG0	0.0	25.6	34.1	23.9	33.3	24.5	7.7	26.0
ARG1	0.0	13.1	10.2	13.8	7.6	14.0	23.1	13.0
ARG2	0.0	2.9	1.1	3.2	3.0	2.8	0.0	2.9
ARG3-5	0.0	0.9	0.0	1.1	0.0	1.0	0.0	0.9
other	0.0	53.6	51.1	54.0	47.0	54.5	53.8	53.6

Table 9: % of predicted properties that co-occur with Llama-3.2-3B's predicted ARGn on SPR1, for the *All-args* prompt variant with "n/a" SPRL responses not allowed. Denominator for percentages is the total number of arguments that property and value were assigned by the model. Columns do not add up to 100 because the model often outputs an SRL prediction that is not any of the numbered ARGs.

	GPT-4o Accuracy (% correct)				
	ARG0	ARG1	ARG2	ARG3-5	All core roles
Prompt template	n = 411	n = 476	n = 122	n = 45	n = 1054
Model must predict everything and is penalized on errors earlier in the pipeline					
Pipeline	29.44	23.95	4.92	2.22	22.96
+instigate,chg-loc	29.2 (−0.24)	24.16 (+0.21)	4.1 (−0.82)	2.22 (+0.0)	22.87 (−0.09)
+instigate,chg-state	29.44 (+0.0)	23.95 (+0.0)	4.1 (−0.82)	2.22 (+0.0)	22.87 (−0.09)
+volition,chg-state	29.44 (+0.0)	24.16 (+0.21)	4.1 (−0.82)	2.22 (+0.0)	22.96 (+0.0)
+volition,chg-loc	29.2 (−0.24)	24.58 (+0.63)	4.92 (+0.0)	2.22 (+0.0)	23.15 (+0.19)
+inst,chg-loc,chg-state	29.2 (−0.24)	24.37 (+0.42)	4.1 (−0.82)	2.22 (+0.0)	22.96 (+0.0)
+SPR1∩ SPR2	29.44 (+0.0)	24.37 (+0.42)	4.92 (+0.0)	2.22 (+0.0)	23.15 (+0.19)
Predicate spans, predicate senses, and argument spans provided					
One arg per prompt	79.56	92.44	86.89	97.78	87.0
+instigate,chg-loc *	82.48 (+2.92)	91.18 (−1.26)	87.7 (+0.81)	97.78 (+0.0)	87.67 (+0.67)
+volition,chg-state *	75.67 (−3.89)	93.28 (+0.84)	81.97 (−4.92)	95.56 (−2.22)	85.2 (−1.8)
+SPR1∩ SPR2 *	78.1 (−1.46)	93.07 (+0.63)	81.15 (−5.74)	97.78 (+0.0)	86.05 (−0.95)
+instigate,chg-loc *	84.43 (+4.87)	89.92 (−2.52)	84.43 (−2.46)	95.56 (−2.22)	87.38 (+0.38)
+volition,chg-state *	73.97 (−5.59)	92.86 (+0.42)	81.97 (−4.92)	97.78 (+0.0)	84.44 (−2.56)
+SPR1∩ SPR2 *	76.16 (−3.4)	93.28 (+0.84)	82.79 (−4.1)	95.56 (−2.22)	85.48 (−1.52)
All args per prompt	97.08	94.75	87.7	86.67	94.5
+instigate,chg-loc	97.81 (+0.73)	92.65 (−2.1)	85.25 (−2.45)	82.22 (−4.45)	93.36 (−1.14)
+instigate,chg-state	96.35 (−0.73)	92.02 (−2.73)	83.61 (−4.09)	86.67 (+0.0)	92.5 (−2.0)
+volition,chg-state	97.81 (+0.73)	94.12 (−0.63)	85.25 (−2.45)	84.44 (−2.23)	94.12 (−0.38)
+volition,chg-loc	96.35 (−0.73)	93.07 (−1.68)	89.34 (+1.64)	82.22 (−4.45)	93.45 (−1.05)
+SPR1	94.16 (−2.92)	91.6 (−3.15)	80.33 (−7.37)	73.33 (−13.34)	90.51 (−3.99)
+SPR1, allowing N/A	95.62 (−1.46)	92.44 (−2.31)	84.43 (−3.27)	77.78 (−8.89)	92.13 (−2.37)
All args per prompt w/ PB guidelines	96.84	93.91	81.15	86.67	93.26
+SPR1	95.62 (−1.22)	92.44 (−1.47)	79.51 (−1.64)	73.33 (−13.34)	91.37 (−1.89)

Table 10: Effect of eliciting SPRL along with SRL across prompt variants on SPR1. For the one-arg-per-prompt template, a gold star (*) indicates the use of gold annotated SPR properties, and a silver star (*) indicates the use of the model's own predicted SPR properties. All other settings elicit the model's own predictions.

(a) GPT-4o accuracy, evaluated on well-formatted responses only, SPR1

Prompt template	ARG0 n = 389	ARG1 n = 450	ARG2 n = 102	ARG3-5 n = 35	All core roles n = 976
All args per prompt	97.17	96.0	90.2	94.29	95.8
+instigate,chg-loc	97.69 (+0.52)	93.78 (−2.22)	87.25 (−2.95)	91.43 (−2.86)	94.57 (−1.23)
+instigate,chg-state	96.4 (−0.77)	92.67 (−3.33)	85.29 (−4.91)	94.29 (+0.0)	93.44 (−2.36)
+volition,chg-state	98.2 (+1.03)	94.67 (−1.33)	86.27 (−3.93)	91.43 (−2.86)	95.08 (−0.72)
+volition,chg-loc	96.92 (−0.25)	94.22 (−1.78)	90.2 (+0.0)	91.43 (−2.86)	94.77 (−1.03)
+SPR1	97.69 (+0.52)	95.56 (−0.44)	87.25 (−2.95)	94.29 (+0.0)	95.49 (−0.31)
+SPR1, allowing N/A	97.43 (+0.26)	94.89 (−1.11)	90.2 (+0.0)	97.14 (+2.85)	95.49 (−0.31)
All args per prompt w/ PB guidelines	97.69	95.11	89.22	94.29	95.49
+SPR1	97.94 (+0.25)	95.56 (+0.45)	88.24 (−0.98)	88.57 (−5.72)	95.49 (+0.0)

(b) Llama-3.2-3B accuracy, evaluated on well-formatted responses only and allowing fuzzy role matching, SPR1

Prompt template	ARG0 n = 332	ARG1 n = 368	ARG2 n = 87	ARG3-5 n = 36	All core roles n = 823
All args per prompt	90.36	67.12	51.72	8.33	72.3
+instigate,chg-loc	92.77 (+2.41)	57.61 (−9.51)	24.14 (−27.58)	8.33 (+0.0)	66.1 (−6.2)
+volition-chg-state	94.58 (+4.22)	56.79 (−10.33)	13.79 (−37.93)	2.78 (−5.55)	65.13 (−7.17)
+SPR1∩ SPR2	93.67 (+3.31)	57.34 (−9.78)	17.24 (−34.48)	13.89 (+5.56)	65.86 (−6.44)
+SPR1	95.18 (+4.82)	57.61 (−9.51)	18.39 (−33.33)	2.78 (−5.55)	66.22 (−6.08)

(c) GPT-4o accuracy, evaluated on well-formatted responses only, Ontonotes

Prompt template	ARG0 n = 13595	ARG1 n = 21742	ARG2 n = 7654	ARG3-5 n = 849	All core roles n = 43840
All args per prompt	95.55	94.98	91.1	84.92	94.28
+instigate,chg-loc	95.67 (+0.12)	91.65 (−3.33)	83.76 (−7.34)	85.98 (+1.06)	91.41 (−2.87)
+volition,chg-state	95.22 (−0.33)	90.41 (−4.57)	80.61 (−10.49)	86.45 (+1.53)	90.11 (−4.17)
+SPR1∩ SPR2	95.28 (−0.27)	92.94 (−2.04)	86.4 (−4.7)	86.93 (+2.01)	92.41 (−1.87)
+SPR1∩ SPR2, allowing N/A	95.3 (−0.25)	91.49 (−3.49)	82.74 (−8.36)	84.81 (−0.11)	91.02 (−3.26)

Table 11: SRL accuracy on subsets of data in which all responses are well-formatted, ensuring penalties are due to errors in meta-linguistic reasoning, rather than failures to correctly format output. All scores are for the *All-args* prompt variant.