Can LLMs Extract Frame-Semantic Arguments?

Jacob Devasier, Rishabh Mediratta, Chengkai Li
University of Texas at Arlington
cli@uta.edu

Abstract

Frame-semantic parsing is a critical task in natural language understanding, yet the ability of large language models (LLMs) to extract frame-semantic arguments remains underexplored. This paper presents a comprehensive evaluation of LLMs on frame-semantic argument identification, analyzing the impact of input representation formats, model architectures, and generalization to unseen and out-of-domain samples. Our experiments, spanning models from 0.5B to 78B parameters, reveal that JSON-based representations significantly enhance performance, and while larger models generally perform better, smaller models can achieve competitive results through fine-tuning. We also introduce a novel approach to frame identification leveraging predicted frame elements, achieving state-of-the-art performance on ambiguous targets. Despite strong generalization capabilities, our analysis finds that LLMs still struggle with out-of-domain data.

1 Introduction

Frame-semantic parsing (Gildea and Jurafsky, 2002) is a fundamental task in natural language understanding that involves identifying semantic frames (Baker et al., 1998) and their associated elements within a sentence. This process is typically divided into three sub-tasks: target identification (detecting words that evoke frames, e.g., began in Figure 1), frame identification (determining the specific frame evoked, i.e., Activity_start), and argument identification (extracting frame elements, i.e., Time, Agent, and Activity).

Traditional approaches to frame-semantic parsing have found success with supervised classification models (Chakma et al., 2024). However, the potential of large language models (LLMs) for this task remains largely unexplored. Recent work (Su et al., 2024) has applied in-context learning with LLMs but found their performance to be significantly weaker, raising concerns about their ability to accurately extract frame-semantic arguments.

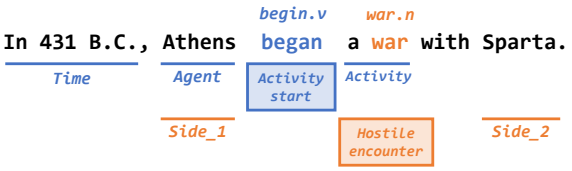

Figure 1: An example of frame-semantic annotations.

In this work, we conduct a comprehensive study on the effectiveness of LLMs for frame-semantic argument identification, evaluating key factors that may influence performance, including input representation formats, model architecture and scale, and generalization to unseen and out-of-domain samples. Our experiments span a diverse range of state-of-the-art LLMs, from 0.5B-parameter models to 78B-parameter models, including both open-source models (e.g., Qwen 2.5 (Qwen et al., 2025), Llama 3) and closed-source models (e.g., GPT-4o).

Recent work (Devasier et al., 2024a) has explored unifying target identification and frame identification by applying frame identification models to candidate targets. We also expand on this idea with a novel method for unifying frame identification and argument identification by leveraging predicted frame elements of candidate frames.

Our findings reveal several important insights. First, we confirm that LLMs struggle in zero-shot and few-shot settings, reinforcing prior concerns about their reliability for frame-semantic parsing. Second, we demonstrate that the choice of input representation significantly impacts model performance, with JSON-based formats showing superior results. Surprisingly, we found that while model scale generally correlates with better performance, smaller models like Qwen 2.5 (3B) outperform the much larger Llama 3.3 (70B). Finally, our proposed frame identification method using predicted frame elements achieves competitive performance, particularly for ambiguous targets (i.e., words that can

evoke multiple frames), where it surpasses previous state-of-the-art approaches by +1.2%.

To summarize, this work makes the following contributions.

- We conduct a systematic evaluation[1] of different input/output representations for frame-semantic parsing with generative LLMs.
- We produce comprehensive benchmarks of different LLM architectures at varying scales, resulting in a +3.9% F1 score improvement over the previous best argument identification model.
- We developed a novel frame identification approach leveraging predicted frame elements on candidate frames which achieves state-of-the-art performance on ambiguous targets.

2 Background and Related Works

Frame-semantic parsing is the automatic extraction of semantic frames and their elements. The task is often applied to FrameNet, a large corpus of frame-semantic annotations and definitions, and is typically separated into three subtasks: target identification, frame identification, and argument extraction (sometimes referred to as frame-semantic role labeling). Target identification is the process of identifying targets–instances of predefined lexical units–in a sentence. Lexical units are unique pairings of words and their meaning, indicated in FrameNet using the word's lemma and part-of-speech (e.g., *begin.v* and *war.n* in Figure 1) which are associated with a particular frame. Frame identification is the process of identifying the frames evoked in a sentence (e.g., Activity_start and Hostile_encounter in Figure 1), often done by classifying the previously extracted targets. Argument extraction is the process of extracting all frame elements of a particular frame evoked in a sentence (e.g., Time, Agent, and Activity for the Activity_start frame in Figure 1).

Nearly all previous systems use classification methods for argument extraction (Chakma et al., 2024). These approaches are primarily dominated by BERT-like encoder models. Argument extraction is often structured as either a token or segment classification task (Su et al., 2024; Zheng et al., 2023, 2022; Bastianelli et al., 2020; Swayamdipta et al., 2017; Lin et al., 2021) or a span identification task (Ai and Tu, 2024; Devasier et al., 2024b; Zheng et al., 2023; Chen et al., 2021). Token/segment classification approaches classify each token or sequence of tokens as one of the frame elements, span identification approaches identify the beginning and end positions of each frame element. One previous study on argument extraction has used zero/few shot in-context learning with a simple prompt on Llama 2 (Su et al., 2024), but observed very poor performance. Another previous study on the very similar task of semantic role labeling has also found a large performance discrepancy when using LLMs (Cheng et al., 2024).

3 Methodology

3.1 Input Representation Design

Previous research has shown that large language models are sensitive to input formatting (Sclar et al., 2023) and that different representations can result in different model performance (Tam et al., 2024; Gao et al., 2024; Macedo et al., 2024). To study these effects on frame-semantics, we systematically evaluated multiple input-output representation formats to determine their impact on frame element extraction performance.

For all input formats, we wrap the target word or phrase in double asterisks, as shown in Table 1, to explicitly mark the token that evokes the frame. This marking helps focus the model's attention on the relevant part of the sentence when making frame element predictions, ensuring that the model identifies frame elements for the correct target.

We developed and tested four distinct representation formats. The Markdown format offers a simple, human-readable approach where frame elements are represented as a markdown list. Each list item contains a frame element name paired with its corresponding text span from the sentence. This format only includes frame elements that the model predicts are present in the input. The XML Tags format provides a structured approach that uses XML-style tags to wrap frame elements within the sentence text. The tag names correspond to frame element names, providing both semantic labeling and precise positional information without requiring additional processing.

We also developed two JSON-based formats. The JSON-Existing format uses frame element names as keys and their corresponding text spans from the sentence as values. Similar to the Markdown format, this only includes predicted frame elements. The JSON-Complete format provides an exhaustive representation different from previous representations that includes all possible frame el-

[1]All of our training and evaluation code is available at https://anonymous.4open.science/r/llm-fsp-831F

Representation	Input	Output
Markdown	Your **contribution** to Goodwill will mean more than you may know.	- Donor: Your - Recipient: to Goodwill
XML Tags		`<Donor>Your</Donor> contribution <Recipient>to Goodwill</Recipient> will mean more than you may know.`
JSON-Existing		`{“Donor”: “Your”, “Recipient”: “to Goodwill”}`
JSON-Complete		`{“Donor”: “Your”, “Recipient”: “to Goodwill”, “Theme”: “”, “Place”: “”, ...}`

Table 1: Representation formats for the given input and outputs.

ements as keys, with empty strings as values for elements not found in the sentence. This format was designed to test whether explicitly presenting all possible frame elements might improve model performance. Examples of each representation format are provided in Table 1, illustrating how they encode the same semantic information in different ways.

3.2 Model Selection and Implementation

To ensure a comprehensive evaluation across the current LLM landscape, we selected models varying in size, architecture, and accessibility. Our selection criteria focused on three key dimensions. In terms of model scale, we included models ranging from 0.5B to 78B parameters, categorizing them into small-scale (0-14B parameters) and large-scale (14B+ parameters) groups to analyze the impact of model size on performance. For architecture diversity, we selected top-performing models from the HuggingFace LLM leaderboard, with particular focus on Qwen 2.5 and Llama 3.2, which have shown strong performance on various tasks.

We included both open-source models (Qwen 2.5, Llama 3, and Phi-4) and closed-source systems (GPT-4o and GPT-4o-mini) to compare performance across different levels of model accessibility. For the open-source models, we implemented fine-tuning using LoRA (Hu et al., 2021). We used $r = 16$ for all models except Llama 3.3 and Qwen 2.5 (72B) where we used $r = 32$, according to best practices. This approach allowed us to optimize model performance while maintaining reasonable computational requirements.

3.3 Evaluation

Our evaluation framework was designed to comprehensively assess model performance across different scenarios and conditions. We began by testing each representation's effectiveness using controlled experiments with GPT-4o-mini. Model performance was evaluated using standard metrics including precision, recall, F1 score, and accuracy with exact match criteria.

To understand data requirements and efficiency, we analyzed performance with varying amounts of training data to understand data efficiency and saturation points. We also conducted extensive testing of model performance on unseen frames, unseen frame elements, and out-of-domain samples. Finally, we analyzed the distribution of argument extraction performance for each frame to gain a granular understanding. This evaluation framework enables us to systematically evaluate LLMs' capabilities in frame-semantic parsing while providing insights into the impact of different design choices and implementation strategies.

4 Experiments

In this section, we thoroughly evaluate the performance of LLMs on frame-semantic parsing through several experiments designed to address three primary research questions: RQ1) How does the representation of FEs impact performance? RQ2) How does model architecture and scale impact performance? RQ3) Are LLMs better on out-of-domain/unseen samples than previous non-LLM methods?

Format	P	R	F1	Acc
XML-tag	0.318	0.368	0.342	0.206
JSON-all	0.356	**0.577**	0.440	0.282
Markdown	0.376	0.554	0.448	0.289
JSON-exist	**0.416**	0.543	**0.471**	**0.308**

Table 2: Few-shot performance metrics for different FE representations using GPT-4o-mini with 0 temperature.

4.1 Dataset

We utilize the FrameNet 1.7 (Ruppenhofer et al., 2016) dataset for our primary experiments. FrameNet provides detailed definitions of semantic frames and their elements, including partially-annotated exemplar sentences for each frame and a corpus of fully-annotated sentences (referred to as "full-text annotations"). We use the full-text annotations for model training due to their complete coverage of frames and frame elements.

Following established conventions from Swayamdipta et al. (2017) and Das and Smith (2011), we use standard train/test splits of non-overlapping documents. The training split consists of 3,353 sentences which evoke 19,391 frames with 34,219 frame elements, while the test split contains 1,247 sentences evoking 6,714 frames and 11,302 frame elements. For out-of-domain evaluation, we use the YAGS dataset, which contains 2,093 test sentences evoking 364 frames with 4,162 frame elements.

4.2 Frame Element Representations (RQ1)

To answer RQ1, we evaluate various FE representation approaches using in-context learning with GPT-4o-mini. Our experiments (Table 2) reveal that JSON-Existing achieves superior performance with significant margins in precision (+4%), F1 score (+3.4%), and accuracy (+1.9%). While JSON-Complete showed higher recall (+4.4%), we attribute this to the simplified cognitive load of outputting all possible frame elements rather than selecting relevant ones. This comprehensive approach leads to more complete but less precise predictions. Notably, XML tag representation performed significantly worse, showing a 12.9 percentage point reduction in F1 score compared to JSON-exist, likely due to the difficulty of the representation. This also suggests that FrameNet's annotations may not be included in common pretraining data as they are originally in XML format.

We found that using JSON-Existing results in the highest precision, F1 score, and accuracy, by significant margins (+4%, +3.4%, and +1.9%, respectively). Interestingly, JSON-Complete had a higher recall (+4.4%). We believe this is due to a reduced cognitive load given by instructing the LLM to output each frame element instead of just the ones which exist in the sentence. This likely reducing the chance of missing particular frame elements, resulting in higher recall. We also found that XML tags performed the worst by a large margin, likely due to the added positional requirements introduced. This indicates that FrameNet is likely not included in the pretraining corpus of LLMs as its native annotations are in XML format.

4.3 Generating LLM Instructions

To validate our instruction creation process, we conducted a comparative study using instructions generated by GPT-4o. The automated approach included all frame-specific information and examples to allow flexibility in prompt generation. Despite being similar to our manual instructions (ROUGE-1/L score: 0.59/0.36), the automated instructions resulted in significantly lower performance (F1 score: 0.225 vs. 0.471). We found that this was primarily due to the LLM predicting frame elements that do not exist, leading us to proceed with manually-crafted instructions for subsequent experiments.

Listing 1: Sample prompt used for zero-shot evaluation.

```
### Task:
You are given a sentence and a frame with its
     associated frame elements and sometimes
     examples. Your task is to label the frame
     elements in the sentence using JSON. Keys
     should only be one of the defined frame
     elements. Do not make up your own frame
     elements, and do not remove or change the input
      in any way. Identify the frame elements based
     on the highlighted target word.

### Frame Information:
Frame Name: Awareness
Frame Definition: A Cognizer has a piece of Content
     in their model of the world. ... [omitted for
     brevity] ...
Examples:
  - Your boss is aware of your commitment. -> {"
       Cognizer": "Your boss", ...}
  ... [omitted] ...

Frame Elements:
Cognizer (Core): The Cognizer is the person whose
     awareness of phenomena is at question.
  - Your boss is **aware** of your commitment. -> {"
       Cognizer": "Your boss"}
  ... [omitted] ...

Explanation (Extra-Thematic): The reason why or how
     it came to be that the Cognizer has awareness
     of the Topic or Content.

### Notes:
- Return the tagged sentence in a ```json ``` code
     block.
- Texts must not overlap.
```

Model	P	R	F1	Acc
GPT-4o-mini	0.416	0.543	0.471	0.308
Deepseek V3	0.466	0.665	0.548	0.377
GPT-4o	0.550	0.642	0.592	0.420
KID	0.741	0.773	0.756	-
AGED	0.757	0.776	0.767	-
Ai and Tu (2024)	0.764	0.777	0.771	-
KAF-SPA	**0.819**	**0.807**	**0.813**	-

Table 3: In-context learning performance comparison.

Listing 2: Sample input for fine-tuning.

```
{
    "role": "system",
    "content": "### Task:
    You are given a sentence and a frame with its
         associated frame elements and sometimes
         examples. Your task is to label the frame
         elements in the sentence using JSON. Keys
         should only be one of the defined frame
         elements. Do not make up your own frame
         elements, and do not remove or change the
         input in any way. Identify the frame
         elements based on the highlighted target
         word.

    ### Notes:
    - Return the tagged sentence in a ```json ```
         code block.
    - Texts must not overlap."
},
{
    "role": "user",
    "content": "### Frame Information
    Frame Name: Law
    Frame Definition: A Law regulates activities or
         states of affairs within a Jurisdiction,
         dictating ... [omitted for brevity] ...

    Frame Elements:
    Law (Core): This FE identifies the rule designed
          to guide ... [omitted]
    ... [omitted]

    ### Input:
    Since the early 1990s , China has improved its
         export controls , including the
         promulgation of **regulations** on nuclear
         and nuclear dual - use exports and has
         pledged to halt exports of nuclear
         technology to un - safeguarded facilities."
},
{
    "role": "assistant",
    "content": "### Output:
    ```json{'Law': 'regulations', 'Forbidden': 'on
         nuclear and nuclear dual - use exports
         '}```"
}
```

4.4 Model Selection and Evaluation (RQ2)

We evaluated the in-context learning performance with GPT-4o, GPT-4o-mini, and Deepseek V3 using the prompt in Listing 1. The results of these models are shown in Table 3. These experiments included several exemplar sentences defined in each frame. Since these in-context learning methods use exemplar data, we include previous works which have used exemplar sentences. Additionally, we benchmark these LLM-based approaches against state-of-the-art systems, including KID (Zheng et al., 2022), AGED (Zheng et al., 2023), and Ai and Tu (2024).

Model	P	R	F1	Acc
Qwen 2.5-0.5B	0.716	0.682	0.699	0.537
Llama 3.2-3B	0.717	0.691	0.704	0.543
Llama 3.2-8B	0.736	0.711	0.724	0.567
Qwen 2.5-1.5B	0.748	0.719	0.733	0.579
Qwen 2.5-3B	0.765	0.740	0.752	0.603
Qwen 2.5-7B	0.769	0.754	0.762	0.615
GPT-4o-mini	0.774	0.762	0.768	0.624
Qwen 2.5-14B	0.782	0.772	0.777	0.635
Phi-4 (14B)	0.793	0.777	0.785	0.646
Llama 3.3-70B	0.748	0.738	0.743	0.591
Qwen 2.5-32B	0.792	0.787	0.789	0.652
Qwen 2.5-72B	**0.798**	**0.790**	**0.794**	**0.658**
Lin et al. (2021)	-	-	0.721	-
AGED	0.750	0.752	0.751	-
KAF-SPA	0.760	0.743	0.751	-
Ai and Tu (2024)	0.756	0.753	0.755	-

Table 4: Performance of different models fine-tuned using JSON-exist. Models are ordered by F1 score and separated into size buckets 0-14B and 14B+.

For fine-tuning, we experimented with Llama 3.2 (3B, 8B), Llama 3.3 (70B), Qwen 2.5 (0.5B-72B), Phi-4 (14B), and GPT-4o-mini[2], as detailed in Table 4. These models were fine-tuned exclusively on the full-text annotations without exemplar sentences. Consequently, we exclude methods that rely on exemplars for fine-tuning. Our fine-tuning prompt is shown in Listing 2.

To assess the impact of instruction tuning, we compared the base and instruction-tuned variants of Qwen 2.5-7B. The instruction-tuned version performed significantly worse (0.703 vs. 0.768 F1 score), leading us to prioritize base models where available in subsequent experiments.

Our results show that Qwen 2.5 consistently outperforms Llama 3 across all model sizes. Most fine-tuned LLMs surpass previous state-of-the-art approaches, with Qwen 2.5 (3B) notably outperforming the much larger Llama 3.3 (70B). Among smaller-scale models, Phi-4 achieved the best performance, while at the larger scale, Qwen 2.5 (72B) outperformed all competitors, including the smaller models. Notably, these two LLMs surpassed the previous best-performing system Ai and Tu (2024) by +3.0% and +3.9% F1 score, respectively.

[2]Due to high training and inference costs, we did not fine-tune GPT-4o.
```

Format	P	R	F1	Acc
5 Most-FE	0.605	0.699	0.649	0.480
5 Diverse	0.648	0.708	0.677	0.511
5 Random	0.717	0.675	0.696	0.533
Full Dataset	**0.774**	**0.762**	**0.768**	**0.624**

Table 5: Performance comparison of GPT-4o-mini fine-tuned on different partitions of the dataset.

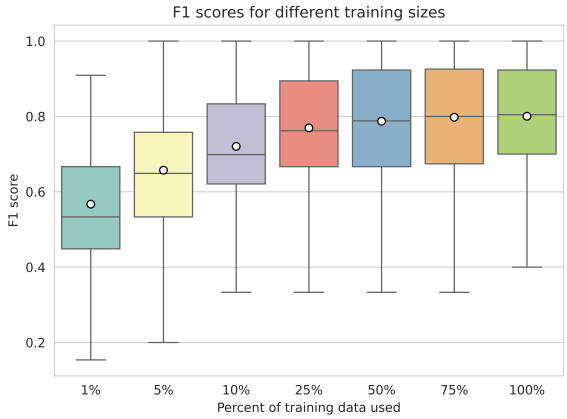

Figure 2: Per-frame argument identification performance distribution for different training dataset sizes.

4.5 Dataset Analysis

Fine-tune Data Subsampling To reduce the costs associated with the high token count of the full training dataset, we first investigated whether strategic subsampling could reduce training overhead and cost while maintaining performance. We evaluated three distinct approaches: selecting up to five samples with the highest number of Frame Elements (5 Most-FE), randomly selecting up to five samples (5 Random), and selecting up to five samples that maximize Frame Element diversity (5 Diverse). Each of these approaches utilize approximately 15% of the original training dataset.

The results of this experiment, presented in Table 5, revealed an interesting trade-off. While the diversity-focused and FE-rich sampling strategies achieved higher recall, they resulted in lower F1 scores and precision compared to random sampling. This suggests that these targeted approaches enhanced the model's ability to identify a broader range of Frame Elements, but at the expense of precision on commonly occurring FEs. Because each of these approaches still fell significantly short of the full dataset's performance, we continue subsequent experiments with the entire dataset.

Training %	P	R	F1	Acc
1%	0.551	0.471	0.508	0.340
5%	0.652	0.590	0.619	0.448
10%	0.728	0.652	0.688	0.524
25%	0.767	0.726	0.746	0.595
50%	0.778	0.753	0.766	0.620
75%	0.781	0.776	0.779	0.638
100%	0.793	0.777	0.785	0.646

Table 6: Detailed per-frame argument identification performance distribution at different levels of data saturation.

Format	P	R	F1	Acc
All	0.793	0.777	0.785	0.646
Unseen Frame	0.725	0.691	0.708	0.548
Unseen FEs	0.560	0.477	0.515	0.347

Table 7: Performance using a fine-tuned Phi-4 model on unseen samples.

Data Saturation Analysis We also examined the relationship between training data volume and LLM performance through systematic experimentation with different dataset sizes during fine-tuning. We conducted this analysis using Phi-4, selected for its combination of strong performance and smaller model size. Each smaller subset is fully contained within larger ones to ensure consistency.

The results of this analysis are presented visually in Figure 2 and in detail in Table 6. The performance trajectory shows distinct phases: a period of steady improvement from 1% to 25% of the dataset, followed by a transition to more modest gains beyond the 50% mark. While the rate of average performance improvement diminishes after utilizing 50% of the data, we observed two notable effects when using the complete dataset: a reduction in the inter-quartile range and enhanced performance on previously challenging frames. This suggests that additional training data continues to contribute to model robustness, even after average performance metrics begin to plateau.

4.6 Unseen and Out-of-domain Data (RQ3)

Unseen Sample Evaluation We evaluate the ability of LLMs to identify frame elements on unseen and out-of-domain data in Table 7. We separate unseen data into two categories, Unseen (Frame) and Unseen (FEs). These categories correspond to test samples whose frames and frame elements are not seen in the training set, respectively. For this

Model	P	R	F1	Acc
GPT-4o	0.363	0.415	0.387	0.240
Phi-4	0.567	0.503	0.533	0.363
SEMAFOR	-	-	**0.570**	-

Table 8: Performance comparison on out-of-domain samples using the YAGS test set.

experiment we use a fine-tuned Phi-4 LLM for the same reason stated above.

Our analysis reveals a notable performance disparity between the two categories of unseen data. On unseen frames, where the entire frame is unseen in the training set, we observe a reduction in performance of -7.7% F1 score compared to the overall performance. This relatively modest degradation suggests that the model has developed a robust general understanding of frame semantics that transfers reasonably well to new frames.

However, on unseen frame elements (FEs), we observe a substantially larger performance drop of -27.0% F1 score. This significant degradation indicates a fundamental challenge in generalizing to entirely new frame elements. The disparity between these two scenarios provides valuable insights into the model's learning dynamics: the model appears to develop strong transferable knowledge about common frame elements that appear across multiple frames, enabling it to maintain reasonable performance even when encountering new frames that use familiar elements.

The stark performance difference with unseen FEs can be attributed to a few factors. First, unseen FEs are often highly specific to particular frames and may represent more nuanced or specialized semantic roles. Second, these elements typically have fewer analogous examples in the training data, limiting the model's ability to learn generalizable patterns. Third, the contextual cues for identifying these specialized FEs may be more subtle or require domain-specific knowledge that the model hasn't adequately acquired during training.

Out-of-domain Evaluation We also evaluate the performance of Phi-4 on out-of-domain samples using the YAGS dataset (Table 8). We include an in-context learning GPT-4o implementation as a baseline along with SEMAFOR (Das et al., 2014). SEMAFOR is one of the first frame-semantic parsing systems, and the only other previous work which was evaluated on the YAGS dataset; however, it is often outperformed by modern approaches. We found that both LLM implementations performed quite poorly, with GPT-4o achieving an F1 score of 0.387 and Phi-4 achieving 0.533. Both of these are lower than SEMAFOR's performance. This indicates an area where LLMs struggle significantly more than previous approaches.

IFEval	BBH	GPQA	MUSR	MMLU-PRO
-0.624	0.519	0.021	**0.835**	0.586

Table 9: Performance partial correlations computed for each benchmark.

We also perform an assessment of the errors of these models to understand their cause. One observation we made is that there are many FEs in the YAGS dataset which are not defined in FrameNet. Another observation is that the sentences in YAGS tend to use poor grammar and often use slang. Additionally, we found that Phi-4's predictions were often more aligned with our human judgments than the original annotations, hinting at a possibility of data quality issues (further discussed in Appendix B). These factors, along with the new topics of discussion in the sentences are likely what leads to the poor performance on YAGS.

4.7 Benchmark Correlation Analysis

Finally, we aim to understand what makes particular LLMs better than others on frame-semantic parsing. To do this, we analyze the correlation between our performance metrics and several common benchmarks for each LLM, where available. For this experiment, we focus on the IFEVal (Zhou et al., 2023), BBH (Suzgun et al., 2022), GPQA (Rein et al., 2023), MUSR (Sprague et al., 2024), and MMLU (Hendrycks et al., 2021) benchmarks. We compute partial correlations between each benchmark and the F1 score on argument identification, accounting for model size as a confounding variable. Table 9 shows the correlation for each benchmark.

Our results indicate that MUSR exhibits the strongest positive correlation with frame-semantic parsing performance. Given that MUSR is designed to assess multistep reasoning, this suggests that models excelling in structured reasoning tasks also tend to perform well in frame-semantic parsing. Similarly, BBH and MMLU-PRO show strong positive correlations, aligning with their emphasis on complex reasoning and broad knowledge across multiple disciplines.

Interestingly, we observe a negative correlation

Model	All	Amb
Phi-4	0.375	0.262
Phi-4$_{cand}$ w/o LF	0.882	**0.862**
Phi-4$_{cand}$ w/ LF	0.894	**0.862**
KAF-SPA	0.912	0.776
KGFI	0.924	0.844
CoFFTEA	**0.926**	0.850

Table 10: Results on frame identification using frame element predictions.

with IFEval, which evaluates instruction-following capabilities using verifiable constraints. This suggests a potential trade-off between strict adherence to instructions and general problem-solving ability. This aligns with our earlier findings (Section 4.4) that instruction-tuned models underperform their base versions on frame-semantic parsing. One possible explanation is that instruction-tuned models prioritize following explicit directives over deep semantic understanding.

4.8 Frame Identification

Previous work (Devasier et al., 2024a) explored the possibility of filtering candidate targets produced by matching potential lexical units using a frame identification model. To build upon this idea towards a single-step frame-semantic parsing method, we explore the potential of frame elements being used to perform frame identification. In this approach, no ground-truth frame inputs are given. This also removes the bias from the model assuming the input always has at least one frame element.

We compared this method with state-of-the-art approaches not using exemplar sentences, including KGFI (Su et al., 2021), CoFFTEA (An et al., 2023), and KAF-SPA (Zhang et al., 2023). We used Phi-4 for this experiment for the same reason as previous experiments. We found that directly using the model performed poorly, likely due to bias in the model's training using ground-truth frames, i.e., each input contains the given frame. To address this, we fine-tuned Phi-4 using candidates (Phi-4$_{cand}$) from the training set produced by Devasier et al. (2024a) and achieved very strong performance.

Sometimes frame elements are predicted for multiple candidate frames. When this happens, we randomly select one of the frames to be used as the prediction. Other options were explored, such as selecting the one with the most frame elements, only selecting the first frame, or utilizing GPT-4o as a tie-breaker, but none of these were effective.

This method showed strong performance, particularly on ambiguous targets–targets with more than one possible frame– where it achieved an accuracy of 0.862, higher than any previous approach. If we apply lexicon filtering (Su et al., 2021) on unambiguous targets, as is common among previous approaches, the overall accuracy is further increased to 89.4%.

5 Conclusion

This work presents a comprehensive evaluation of large language models for frame-semantic parsing, with a particular focus on argument identification. Our systematic analysis reveals several important insights about the capabilities and limitations of LLMs in this domain. While LLMs demonstrate poor performance in zero-shot and few-shot settings, fine-tuned models achieve state-of-the-art results, with Qwen 2.5 (72B) surpassing previous approaches by a significant margin (+3.9% F1 score).

Our investigation into input representations demonstrates that LLMs are sensitive to specific formats, with JSON-based formats achieving superior performance compared to alternatives. Our correlation analysis between frame-semantic parsing performance and common LLM benchmarks reveals that models excelling in multistep reasoning (as measured by MUSR) tend to perform better at argument identification, while instruction-following capabilities (measured by IFEval) show a negative correlation.

However, our results also highlight significant challenges. The substantial performance degradation on unseen frame elements (-27.0% F1 score) and out-of-domain data indicates that current LLM approaches, despite their improvements over previous methods, still struggle with generalization. This limitation suggests that frame-semantic knowledge may not be sufficiently encoded, and that additional strategies may be needed to enhance model robustness across diverse contexts.

Our novel approach to frame identification using predicted frame elements shows promise, particularly for ambiguous targets, where it achieves state-of-the-art performance. This suggests that integrating frame element predictions into the frame identification process could be a valuable direction for future research.

Limitations

Several methodological constraints impacted the scope and comprehensiveness of our analysis. Due to the substantial computational costs associated with fine-tuning large language models, we were unable to explore fine-tuning on certain high-performing models like GPT-4o and GPT-4. These models may have achieved even stronger results than those demonstrated in our current analysis.

Our experimental design relied on sequential parameter optimization to manage computational requirements. While this approach was practical, it introduces the possibility that certain combinations of parameters could yield unexpected results. For instance, XML representations might potentially outperform JSON embeddings when paired with 14B parameter models or applied to frame identification tasks. However, exploring these combinations was beyond the computational resources available for this study.

The scope of our research was limited to the English FrameNet dataset. As a result, our findings may not generalize to other languages or semantic frameworks. Cross-lingual validation would be necessary to establish the broader applicability of our approaches.

In the context of frame identification for ambiguous targets, our current method of handling multiple frames with predicted frame elements requires refinement. The randomized prediction approach can lead to inconsistent outputs. Additionally, our implementation used a fixed random seed of 0 for reproducibility, but we did not explore the potential impact of different random seeds on accuracy metrics.

Finally, our benchmark correlation analysis considered only model size as a confounding variable. This simplified approach may not account for other significant factors that could influence the relationship between benchmark performance and frame-semantic parsing capabilities. A more comprehensive analysis of confounding variables would provide deeper insights into these relationships.

References

Chaoyi Ai and Kewei Tu. 2024. Frame semantic role labeling using arbitrary-order conditional random fields. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17638–17646.

Kaikai An, Ce Zheng, Bofei Gao, Haozhe Zhao, and Baobao Chang. 2023. Coarse-to-fine dual encoders are better frame identification learners. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13455–13466, Singapore. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Emanuele Bastianelli, Andrea Vanzo, and Oliver Lemon. 2020. Encoding syntactic constituency paths for frame-semantic parsing with graph convolutional networks. *ArXiv*, abs/2011.13210.

Kunal Chakma, Sima Datta, Anupam Jamatia, and Dwijen Rudrapal. 2024. Semantic role labelling: A systematic review of approaches, challenges, and trends for english and indian languages.

Xudong Chen, Ce Zheng, and Baobao Chang. 2021. Joint multi-decoder framework with hierarchical pointer network for frame semantic parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2570–2578, Online. Association for Computational Linguistics.

Ning Cheng, Zhaohui Yan, Ziming Wang, Zhijie Li, Jiaming Yu, Zilong Zheng, Kewei Tu, Jinan Xu, and Wenjuan Han. 2024. Potential and limitations of llms in capturing structured semantics: A case study on srl. In *Advanced Intelligent Computing Technology and Applications*, pages 50–61, Singapore. Springer Nature Singapore.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Dipanjan Das and Noah A. Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444, Portland, Oregon, USA. Association for Computational Linguistics.

Jacob Devasier, Yogesh Gurjar, and Chengkai Li. 2024a. Robust frame-semantic models with lexical unit trees and negative samples. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6930–6941, Bangkok, Thailand. Association for Computational Linguistics.

Jacob Devasier, Rishabh Mediratta, Phuong Anh Le, David Huang, and Chengkai Li. 2024b. ClaimLens: Automated, explainable fact-checking on voting claims using frame-semantics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 311–319, Miami, Florida, USA. Association for Computational Linguistics.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2024. Text-to-sql empowered by large language models: A benchmark evaluation. *Proc. VLDB Endow.*, 17(5):1132–1145.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

ZhiChao Lin, Yueheng Sun, and Meishan Zhang. 2021. A graph-based neural model for end-to-end frame semantic parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3864–3874, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marcos Macedo, Yuan Tian, Filipe Cogo, and Bram Adams. 2024. Exploring the impact of the output format on the evaluation of large language models for code translation. In *Proceedings of the 2024 IEEE/ACM First International Conference on AI Foundation Models and Software Engineering*, FORGE '24, page 57–68, New York, NY, USA. Association for Computing Machinery.

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. *Preprint*, arXiv:2412.15115.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. FrameNet II: Extended theory and practice. *International Computer Science Institute, Berkeley, California.*

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *ArXiv*, abs/2310.11324.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *Preprint*, arXiv:2310.16049.

Xuefeng Su, Ru Li, Xiaoli Li, Jeff Z. Pan, Hu Zhang, Qinghua Chai, and Xiaoqi Han. 2021. A knowledge-guided framework for frame identification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5230–5240, Online. Association for Computational Linguistics.

Xuefeng Su, Ru Li, Xiaoli Li, and Zhichao Yan. 2024. A unified framework for frame-semantic parsing based on marker attention. *Data Intelligence*, N/A(N/A):N/A. Open Access.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *CoRR*, abs/1706.09528.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.

Rui Zhang, Yajing Sun, Jingyuan Yang, and Wei Peng. 2023. Knowledge-augmented frame semantic parsing with hybrid prompt-tuning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Ce Zheng, Xudong Chen, Runxin Xu, and Baobao Chang. 2022. A double-graph based framework for frame semantic parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4998–5011, Seattle, United States. Association for Computational Linguistics.

Ce Zheng, Yiming Wang, and Baobao Chang. 2023. Query your model with definitions in framenet: an effective method for frame semantic role labeling. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Reproducibility

To fine-tune the models in this work, we used three different systems. For the small models, (0.5-7B parameters) we experimented with training and evaluating on a system with 1x Nvidia RTX 4070 and another system with 1x Nvidia A100 40GB. For medium-sized models (14-32B parameters), we only experimented with the system with 1x Nvidia A100 40GB. For large models (70B+ parameters), we used a third system with 1x Nvidia H100 80GB.

Github Copilot was used in the creation of some of our code.

B YAGS Quality Assessment

We found several labels which we disagree with among a small random sample of Phi-4's predictions compared with the original annotations. We show these examples in Table 11.

Sentence	YAGS Annotation	Our Annotation
i feel that the pagan and wican be a lose people in **need** of a savior .	{'Dependent': 'at the pagan and wican be a lose people', 'Requirement': 'at the pagan and wican be a lose people in need of a savior'}	{'Cognizer': 'the pagan and wican', 'Requirement': 'of a savior'}
how do u **get** rid of or cover up razor burn ?	{'Entity': 'u'}	{'Entity': 'u', 'Final_quality': 'rid of or cover up razor burn'}

Table 11: Examples of disagreements in our annotations compared to YAGS which may contribute to low performance.