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 72B 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 arguments 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, e.g., ACTIVITY_START), and *argument identification* (extracting frame elements, e.g., 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 works (Su et al., 2024; Cui and Swayamdipta, 2024) have applied in-context learning with LLMs but found their performance to be significantly weaker than previous non-LLM methods.

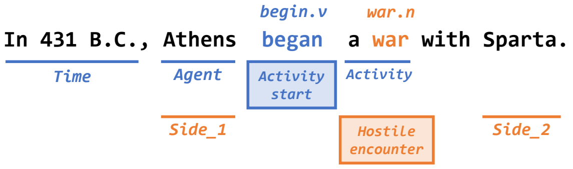

Figure 1: An example of frame-semantic annotations.

In this work, we conduct a comprehensive study[1] on the effectiveness of LLMs for argument identification, evaluating key factors that may influence performance, including input representation formats, model architecture and scale, and their generalizability to unseen and out-of-domain samples. Our experiments span a diverse range of state-of-the-art LLMs, from 0.5B to 72B-parameter models, including both open-source models (Qwen 2.5 (Qwen et al., 2025), Llama 3 (Meta, 2024), Phi-4 (Abdin et al., 2024), and Deepseek V3 (DeepSeek-AI, 2025)) and closed-source models (GPT-4o/4o-mini (OpenAI, 2024)).

Recent work (Devasier et al., 2024a) has explored unifying target identification and frame identification by applying a frame identification model 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 experiments on FrameNet 1.7 (Ruppenhofer et al., 2016) reveal several important insights. First, we confirm that LLMs struggle in in-context learning 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 gener-

[*] Corresponding author

[1] Our training and evaluation code is available at `https://github.com/idirlab/llm-frame-semantic-parser`.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 30609–30622
November 4-9, 2025 ©2025 Association for Computational Linguistics

ally correlates with better performance, smaller models such as Qwen 2.5 (3B) outperform the much larger Llama 3.3 (70B). We also applied our findings to the Chinese FrameNet (Li et al., 2024) dataset to evaluate the cross-lingual performance of LLMs on argument identification. This resulted in an improvement of +9% over the baseline from CCL25 [2] by using a fine-tuned Qwen2.5-7B. Finally, our proposed frame identification method using predicted frame elements achieves strong performance, particularly for ambiguous targets (words which can evoke multiple frames), where it surpasses the previous state-of-the-art by +1.2%.

We summarize our contributions as follows:

- We conducted a systematic evaluation of different frame element representations for argument identification with generative LLMs in English.
- We produced 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 (Ruppenhofer et al., 2016), a large corpus of frame-semantic annotations and definitions, and is typically separated into three subtasks: *target identification*, *frame identification*, and *argument identification* (also 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), often done by classifying the previously extracted targets. Argument identification 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).

Nearly all previous systems use classification methods for argument identification (Chakma et al., 2024). These approaches are primarily dominated by BERT-like (Devlin et al., 2019) encoder models. Argument identification 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, whereas span identification approaches identify the beginning and end positions of each frame element. Two previous studies on argument identification have used in-context learning with a simple prompt on Llama 2 (Su et al., 2024) and GPT-4 (Cui and Swayamdipta, 2024), but observed very poor performance. Cui and Swayamdipta (2024) also explored the use of LLMs for augmenting FrameNet by generating new sentences. Another study on the similar task of semantic role labeling also found a performance reduction when using LLMs (Cheng et al., 2024).

3 Methodology

3.1 Frame Element 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 argument identification 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. Table 1 provides examples of each representation format. 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

[2] https://tianchi.aliyun.com/competition/entrance/532338

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-Exist		`{“Donor”: “Your”, “Recipient”: “to Goodwill”}`
JSON-All		`{“Donor”: “Your”, “Recipient”: “to Goodwill”, “Theme”: “”, “Place”: “”, ...}`

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

sentence. The XML Tags format uses XML-style tags to wrap frame elements within the sentence text. The tag names correspond to frame element names, and provide argument labels and positional information without ambiguity even if tokens identical to the frame elements appear elsewhere in the same sentence.

We also developed two JSON-based formats. The JSON-Exist format uses frame element names as keys and their corresponding text spans from the sentence as values. JSON-Exist, Markdown, and XML Tags all only predict frame elements which are present in the input sentence. The JSON-All format provides an exhaustive representation different from previous representations that includes all possible frame elements 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.

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

For in-context learning experiments, we sample all exemplar sentences defined within each frame's XML file, including both frame and frame element-specific exemplar sentences. On average, this resulted in 4.57 frame exemplars and 4.95 frame element exemplars.

3.3 Evaluation

We began by testing each representation's effectiveness using controlled experiments with GPT-4o-mini. Model performance was evaluated using precision, recall, F1 score, and accuracy, all of which required exact matches on frames and frame elements and their arguments. To understand data requirements and efficiency, we analyzed performance with varying amounts of training data. 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.

4 Experiments

This section evaluates the performance of LLMs on argument identification through several experiments designed to address three primary research questions: RQ1) How does the representation of frame elements (FEs) impact performance? RQ2)

Format	P	R	F1	Acc
GPT-4o mini in-context learning				
XML Tags	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**
Qwen 2.5-7B in-context learning				
Markdown	0.330	0.253	0.287	0.167
XML Tags	0.302	0.288	0.295	0.173
JSON-All	0.302	**0.499**	0.376	0.232
JSON-Exist	**0.339**	0.490	**0.401**	**0.251**
Qwen 2.5-72B in-context learning				
Markdown	0.330	0.400	0.362	0.221
XML Tags	0.383	0.487	0.429	0.273
JSON-All	0.377	**0.635**	0.473	0.310
JSON-Exist	**0.418**	0.616	**0.498**	**0.332**

Table 2: In-context learning performance using different frame element representations.

How does model architecture and scale impact performance? RQ3) Are LLMs better on out-of-domain/unseen samples than non-LLM methods?

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 only use the full-text annotations—a set of documents where each sentence is fully annotated for all of the evoked frames and frame elements—for model training due to their complete coverage of frame elements.

We use standard train/test splits from conventions established in Swayamdipta et al. (2017) and Das and Smith (2011). 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 (Hartmann et al., 2017), which contains 2,093 test sentences evoking 364 frames with 4,162 frame elements. YAGS differs from FrameNet 1.7 largely due to its focus on informal and user-generated content as opposed to FrameNet's high-quality texts.

4.2 Frame Element Representations (RQ1)

To answer RQ1, we evaluate different frame element (FE) representation approaches using in-context learning with GPT-4o-mini. For each frame, we include up to five annotated training examples in the prompt, resulting in an average of 4.6 examples per frame. We do not enforce structured output or assume the LLM will produce well-formed responses. Predictions that cannot be parsed in their corresponding format (JSON, XML, or Markdown) or contain misspelled or invalid frame elements are considered incorrect predictions. However, in practice, these errors were rare and had minimal impact on overall performance.

All prompts for the evaluated models were developed without first testing them on any specific model to avoid potentially biasing performance. We also explored LLM-generated prompts in Appendix B.1, but found worse performance than our manually created prompts.

Our experiments (Table 2) reveal JSON-Exist achieves the best performance across model size and family. While JSON-All consistently showed higher recall, we attribute this to the simplified cognitive load of outputting all possible frame elements rather than selecting relevant ones. The Qwen 2.5 family models tend to perform poorly on the Markdown format, while the GPT-4o mini model achieved performance on-par with JSON-All. This indicates that GPT-4o mini likely places larger importance on markdown-style pre-training data than the Qwen 2.5 family models. XML Tags consistently performed poorly, likely due to the added difficulty of needing to positionally represent the tags within the sentence. This also suggests that FrameNet's full-text annotations—originally in XML format—were likely not included in the pretraining data of the LLMs, as their inclusion would have likely led to better performance in our experiments.

4.3 Model Selection and Evaluation (RQ2)

We evaluated the in-context learning performance of a diverse set of LLMs 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, with each model receiving the exact same inputs. Because of this, we include previous works which have also used exemplar sentences, including strong baselines KID (Zheng et al., 2022), AGED (Zheng

Model	P	R	F1	Acc
In-context learning (no fine-tuning)				
Llama 3.1-8B	0.199	0.331	0.249	0.142
Qwen 2.5-7B	0.338	0.392	0.401	0.251
Phi-4	0.390	0.534	0.451	0.291
GPT-4o-mini	0.416	0.543	0.471	0.308
Qwen 2.5-72B	0.418	0.616	0.498	0.332
Deepseek V3	0.466	0.665	0.548	0.377
GPT-4o	0.550	0.642	0.592	0.420
Supervised baselines				
Lin et al. (2021)	–	–	0.721	–
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 of language models compared to baselines. Models that use exemplar sentences in their training data are marked with *.

et al., 2023), and the state-of-the-art, Ai and Tu (2024). KID utilizes two graphs to represent frame semantic structures along with a GCN (graph convolutional network) (Kipf and Welling, 2017) to encode the inputs; AGED jointly encodes a given sentence and a frame and its frame elements and uses a pretrained language model to map frame element tokens to input spans; Ai and Tu (2024) expands on the work of AGED by jointly modeling the interactions of each frame element together instead of separately.

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,[3] as detailed in Table 4. These models were fine-tuned exclusively on the full-text annotations without exemplar sentences. To maintain fairness, we exclude methods that fine-tune using exemplars in Table 4; however, these results are directly comparable with Table 3.

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.

Our results showed 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.

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 3.1-8B	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 models fine-tuned using the JSON-Exist format. Models are grouped by size (0–14B and 14B+), and sorted within each group by F1 score. The performance of previous systems is shown at the bottom.

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 of GPT-4o-mini fine-tuned on different dataset subsets. "5 Most-FE" uses the five samples of each frame with the most FE annotations, "5 Diverse" selects five samples which maximize FE diversity, and "5 Random" samples five at random. Full dataset performance is shown for comparison.

4.4 Dataset Analysis

Fine-tune Data Subsampling To reduce the costs associated with the high token count of the full training dataset, we investigated whether strategic subsampling could reduce training overhead and cost while maintaining performance. We evaluated three distinct approaches: selecting from each frame up to five samples with the highest number of frame elements (5 Most-FE), randomly selecting up to five samples from each frame (5 Random),

[3] Due to high training costs, we did not fine-tune GPT-4o.

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: Performance of a fine-tuned Phi-4 model on argument identification across increasing training data.

and selecting up to five samples from each frame that maximize the number of distinct frame elements (5 Diverse). Each of these approaches utilize approximately 15% of the original training dataset.

We performed this experiment by fine-tuning Phi-4. 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 FEs, 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.

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. Each smaller subset is fully contained within larger ones to ensure consistency. We conducted this analysis using Phi-4, selected for its combination of strong performance and smaller model size.

The results of this analysis are presented in Table 6 and Figure 2 (in Appendix). The results show a period of steady improvement from 1% to 25% of the dataset, followed by 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 improved performance on frames the model previously struggled with. This indicates that additional training data continues to contribute to model robustness, even after average performance metrics begin to plateau.

Model	P	R	F1
GPT-4o	0.523	0.503	0.512
CCL25 Baseline*	0.557	0.566	0.562
Qwen 2.5-7B	0.668	0.637	0.652
Qwen 2.5-7B (EN+CN)	**0.681**	**0.652**	**0.666**

Table 7: Results on argument identification using gold frames in Chinese FrameNet 2.1 development dataset. *Our reproduced results of the CCL25 baseline.

4.5 Multilingual Applicability: Chinese

To investigate the multilingual applicability of our approach, we evaluated LLM performance on Chinese FrameNet 2.1 (Li et al., 2024). This dataset presents unique challenges compared to its English counterpart, with significantly more complex frame structures containing an average of 43 frame elements per frame. Our experiments compared three systems: (1) a BERT-based baseline provided for the CCL25-Eval Task, [4] (2) GPT-4o with in-context learning using the same prompting strategy as our English experiments, and (3) LoRA fine-tuned Qwen 2.5-7B on Chinese FrameNet training data using our original English prompts. All models were evaluated on the Chinese FrameNet development set, as the official evaluation set lacked ground-truth frame annotations. The results of this experiment are shown in Table 7.

While our fine-tuned Qwen 2.5-7B significantly outperformed both baselines, the performance gains were less pronounced than those observed in our English experiments. This performance gap likely stems from the increased structural complexity of Chinese frames and potential cross-lingual challenges when applying prompting strategies developed for English to Chinese data.

We also experimented with fine-tuning Qwen 2.5-7B on both English and Chinese to see if the additional training samples can improve the learned task representations of the LLM. This approach, labeled as (EN+CN) in Table 7, showed slight performance improvements on the Chinese FrameNet; however, we did not find similar performance improvements on the English FrameNet test set, which resulted in an F1 score of 0.741 (–2.1% vs. only English).

[4] https://github.com/SXUNLP/The-3nd-Chinese-Frame-Semantic-Parsing

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 8: Performance of a fine-tuned Phi-4 model on unseen subsets. "Unseen Frame" indicates frames absent from the training set; "Unseen FEs" refers to individual frame elements not encountered during training.

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

4.6.1 Unseen Sample Evaluation

We evaluate the ability of LLMs to identify frame elements on unseen data in Table 8. We separate unseen data into two categories, Unseen Frame and Unseen FEs. These categories correspond to test samples whose frames and frame-specific frame elements are not seen in the training set, respectively. For this experiment we use the previously fine-tuned Phi-4 model.

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 performance across the entire test set. This 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, 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 using common frame element names which appear across multiple frames, enabling it to maintain reasonable performance even when encountering new frames and their frame 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.

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 9: Performance on out-of-domain argument identification using the YAGS test set. SEMAFOR results are reported from Hartmann et al. (2017).

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

Table 10: Partial correlations of the argument identification performance with five benchmarks.

4.6.2 Out-of-domain Evaluation

We also evaluate the performance of Phi-4 on out-of-domain samples using the YAGS dataset (Hartmann et al., 2017). These results are presented in Table 9. 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. Surprisingly, SEMAFOR outperformed both of these.

We performed a qualitative assessment of the errors of these models to understand their cause. We observed that many FEs in the YAGS dataset are not defined in FrameNet, which may explain much of the performance drop. Additionally, 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 in YAGS (further discussed in Appendix B.3). While its performance is also poor, we believe SEMAFOR performs better on this task because it is designed to identify probable spans of arguments before attaching role labels to each span.

Model	All	Ambiguous
Phi-4	0.375	0.262
Phi-4$_{cand}$ without LF	0.882	**0.862**
Phi-4$_{cand}$ with LF	0.894	**0.862**
KAF-SPA	0.912	0.776
KGFI	0.924	0.844
CoFFTEA	**0.926**	0.850

Table 11: Frame identification accuracy using FE predictions. Results are shown for all targets and ambiguous targets. LF refers to lexicon filtering.

4.7 Benchmark Correlation Analysis

Finally, we aim to understand what makes particular LLMs better than others on argument identification. To do this, we analyze the correlation between frame-semantic parsing and several common benchmarks for each LLM. 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-PRO (Wang et al., 2024) benchmarks. We compute partial correlations (Table 10) between each benchmark and the F1 score on argument identification, with model size as a confounding variable.

Our results indicate that MUSR has 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. Interestingly, we observe a negative correlation with IFEval, which evaluates an LLM's instruction-following capabilities. This suggests a potential trade-off between adherence to instructions and general problem-solving ability. This aligns with our earlier findings (Section 4.3) that instruction-tuned models underperform their base versions on frame-semantic parsing.

4.8 Frame Identification

Devasier et al. (2024a) previously explored unifying the target and frame identification steps by filtering candidate targets using a frame identification model. To build upon this idea towards a single-step frame-semantic parsing method, we explore using predicted frame elements to perform frame identification. We approach this by making the critical assumption that if the model predicts frame elements for a given frame, then that frame must be evoked in the sentence. We use the same instructions as before, but we apply this to each candidate frame for a particular target instead of just the ground truth frame. We select candidate frames using lexicon filtering (Hartmann et al., 2017).

In Table 11 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). KGFI uses GCN-based and BERT-based encoders to improve frame representations, COFFTEA uses a two-stage training process to improve learned representations from a dual-encoder pretrained language model (PLM), and KAF-SPA encodes frame information extracted from an end-to-end memory network into an encoder-decoder PLM. We used the previously fine-tuned Phi-4 model for this experiment for the same reason as previous experiments. We found that directly using the model performed poorly due to the model being biased during training to assume the ground-truth frame is always given. To address this, we fine-tuned Phi-4 using candidate frames (Phi-4$_{cand}$) from the training set. For incorrect candidate frames, we train the model to predict an empty JSON object.

Sometimes frame elements are predicted for multiple candidate frames. To solve this, we randomly select one of the frames to be used as the prediction. We explored other methods of frame disambiguation, such as selecting the one with the most frame elements, only selecting the first frame, or utilizing a second LLM step as a tie-breaker; however, none of these were effective. This frame identification method shows 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. Applying lexicon filtering on unambiguous targets, as is common among previous approaches, further increases overall accuracy 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). Different model families, namely Llama 3 and Qwen 2.5, show significantly different performance, with Qwen consistently outperforming with the same number of parameters. We believe the consistent performance improvements likely stem from the increased (18T vs 15T tokens) and more diversified pretraining and structured data-focused post-training in Qwen 2.5.

Our investigation into input representations demonstrates that LLMs are sensitive to specific input and output formats, with JSON formats achieving superior performance for argument identification. Our correlation analysis between argument identification performance and common LLM benchmarks reveals that models excelling in multi-step reasoning (as measured by MUSR) tend to perform better at argument identification, while instruction-following capabilities (measured by IFEval) show a negative correlation. Our cross-lingual analysis on Chinese FrameNet 2.1 shows that these performance improvements hold in cross-lingual settings. Furthermore, combining English and Chinese training data improved performance on Chinese argument identification by +1.4% F1.

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, 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.

Finally, our novel approach to frame identification using predicted frame elements of candidate frames shows promising results, 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 in further improving frame identification models and in unifying frame-semantic parsing.

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 such as GPT-4o and GPT-4. These models may achieve stronger results than those demonstrated in our current analysis.

Our analysis did not include any newer thinking/reasoning-focused models, e.g., Qwen3 (Yang et al., 2025). These models appear to be very capable and would likely have significantly improved performance. Future work is needed to analyze their capabilities on frame-semantic parsing.

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 experiment on Chinese FrameNet was limited to fine-tuning a single model using our findings from the English dataset. As a result, our findings may not hold for all input representations and models. Furthermore, our prompts were cross-lingual and were not written in Chinese, so the effects of fully Chinese instructions are not studied. Previous work (Huang et al., 2023; Dey et al., 2024) on the performance difference between English and native language prompting is not fully conclusive, but performance is often shown to be similar or better when prompted in English.

Our current method of handling multiple frames for frame identification with predicted frame elements requires refinement. The randomized prediction approach will 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. Future work should explore better methods for frame disambiguation.

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

Ethics Statement

This study explores the use of large language models (LLMs) for frame semantic parsing using the FrameNet dataset, a publicly available, expert-annotated linguistic resource intended for research. All experiments are conducted within the dataset's intended use and contain no private or sensitive information. The study involves no human subjects or personal data and is therefore exempt from formal IRB approval. Nonetheless, we adhere to responsible research practices and will release code and results to support transparency and reproducibility. The underlying language models used may reflect biases from their pretraining data.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

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.

Xinyue Cui and Swabha Swayamdipta. 2024. Annotating FrameNet via structure-conditioned language generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 681–692, Bangkok, Thailand. Association for Computational Linguistics.

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.

DeepSeek-AI. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Krishno Dey, Prerona Tarannum, Md. Arid Hasan, Imran Razzak, and Usman Naseem. 2024. Better to ask in english: Evaluation of large language models on english, low-resource and cross-lingual settings. *Preprint*, arXiv:2410.13153.

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.

Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 2017. Out-of-domain FrameNet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 471–482, Valencia, Spain. Association for Computational Linguistics.

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.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ru Li, Yunxiao Zhao, Zhiqiang Wang, Xuefeng Su, Shaoru Guo, Yong Guan, Xiaoqi Han, and Hongyan Zhao. 2024. A comprehensive overview of cfn from a commonsense perspective. *Machine Intelligence Research*, 21(2):239–256.

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.

Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

OpenAI. 2024. Hello gpt-4o.

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.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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 evaluation on a system with 1× Nvidia RTX 4070 and another system with 1× Nvidia A100 40GB. For medium-sized models (14-32B parameters), we only experimented with the system with 1× Nvidia A100 40GB. For large models (70B+ parameters), we used a third system with 1× Nvidia H100 80GB. Some portions of our code were developed with the assistance of GitHub Copilot.

The use agreement for FrameNet does not allow sharing of the original data. Access to the FrameNet data can be requested on `https://framenet.icsi.berkeley.edu/`.

Each of our experiments is done with a single run. To minimize variation, each model is trained and inferenced on the same initial random seed with the data shuffled using the same random seed as well. We attempted to check the variation in predictions using Phi-4, but found zero difference between the two separate inference runs on the test dataset. For in-context learning methods, we also set temperature to 0 to minimize variation in predictions.

A.1 LLM Prompts

Listing 1: Sample prompt used for in-context learning.

```
### 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.
```

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'}```"
}
~~~

B Additional Experiments

B.1 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 our manually-crafted instructions for subsequent experiments.

B.2 Data Saturation Analysis

We also include a visual presentation (Figure 2) of the improvements in argument identification with increasing portions of training data, as described in Section 4.4.

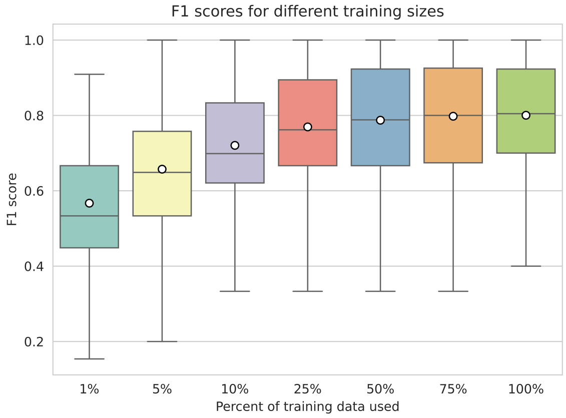

Figure 2: Per-frame argument identification performance distribution for different training dataset sizes.

B.3 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 two of these examples in Table 12.

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 12: Examples of disagreements in our annotations compared to YAGS which may contribute to low performance. The first row evokes the NEEDING frame and the second evokes the BECOMING frame.