Injecting Frame Semantics into Large Language Models via Prompt-Based Fine-Tuning

Shahid Iqbal Rai **Danilo Croce** **Roberto Basili**
Department of Enterprise Engineering
University of Rome Tor Vergata, 00133, Rome, Italy
rjshahidrai@gmail.com {croce,basili}@info.uniroma2.it

Abstract

Large Language Models (LLMs) have demonstrated remarkable generalization across diverse NLP tasks, yet they often produce outputs lacking semantic coherence due to insufficient grounding in structured linguistic knowledge. This paper proposes a novel method for injecting Frame Semantics into a pretrained LLaMA model using Low-Rank Adaptation (LoRA). Leveraging FrameNet (a rich resource of over 1,000 semantic frames) we construct a training corpus comprising structured triples of frame definitions, frame elements, and lexical units. Our method encodes these examples into the model via LoRA adapters and evaluates performance using zero-shot prompting for textual entailment and semantic role labeling (SRL) over FrameNet. Experimental results show that our adapted frame-aware LLM substantially outperforms the baseline across closed, open-ended, and multiple-choice prompts. Moreover, we observe significant improvements in SRL accuracy, demonstrating the efficacy of combining frame-semantic theory with parameter-efficient pretraining.

1 Introduction

Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023) and LLaMA (Dubey et al., 2024) have demonstrated impressive capabilities across a wide range of natural language processing (NLP) tasks. However, despite their generalization strength, these models often lack explicit grounding in linguistic theories, which can occasionally result in fluent outputs that overlook deeper semantic distinctions and, in some cases, lead to factual inconsistencies or semantic hallucinations (Ji et al., 2023). To address this gap, enriching LLMs with structured linguistic knowledge could certainly be beneficial, as improved interpretability may support more reliable and semantically coherent outputs.

One promising direction is the integration of Frame Semantics (Fillmore, 1976), a linguistic theory that connects word semantics to situational, i.e. conceptualized, information in terms of *frames*. Each frame consists of a situation (i.e. the frame) and prototypical participants, known as *Frame Elements* (FEs). It is triggered by specific *Lexical Units* (LUs) in the text. For example, the verb *provide* triggers a SUPPLY frame, with roles such as SUPPLIER, RECIPIENT, and THEME. Unlike purely distributional approaches, Frame Semantics imposes situational constraints on semantic role assignments, grounding language interpretation in real-world scenarios. FrameNet (Baker et al., 1998), a computational resource based on Frame Semantics, offers a comprehensive repository of over 1,000 frames and their annotated instances. In fact, equipping models with frame-level information can make a tangible difference for tasks like semantic role labeling (Das and Smith, 2010), question answering (Madabushi et al., 2024), and even commonsense reasoning (Botschen et al., 2018b; Wang et al., 2021b).

In this work, we present a parameter-efficient method to inject Frame Semantics into large language models (LLMs) through fine-tuning with Low-Rank Adaptation (LoRA) (Hu et al., 2022). Our central idea is to make an abstract linguistic theory usable by LLMs by textualizing its core concepts: we systematically convert FrameNet's structured knowledge (frame definitions, frame elements or FEs, and lexical units or LUs) into natural language examples in the form of question–answer pairs. For instance, we generate prompts that ask for the definition of a frame, the roles it involves, or the words that evoke it, thus producing an artificial dataset that "translates" theoretical content into a format suitable for instruction-based adaptation. In total, this process yields a dataset of 6,628 question–answer pairs covering 60 FrameNet frames. By fine-tuning LLaMA models on this textualized

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 31–47

dataset, we aim to encourage the model to internalize frame-semantic structures and relationships, enabling it to better reason about frames, roles, and their instantiations in text, even in the absence of explicit annotation.

A critical question, however, is whether LLMs fine-tuned on such examples merely *memorize* specific facts about the frames encountered during training, or whether they actually *generalize* frame-semantic knowledge to novel, previously unseen frames. To address this, we explicitly evaluate model performance on both **seen frames** (included in fine-tuning) and **unseen frames** (held out from training). This experimental design allows us to disentangle the model's ability to recall injected knowledge from its capacity to abstract and apply frame-semantic principles to new scenarios-an essential property for robust knowledge integration.

We assess the effectiveness of our approach in two ways. First, we probe the model's frame-semantic competence by evaluating its ability to answer structured questions about frames, elements, and lexical units-essentially measuring whether the injected knowledge is accessible via prompting. Second, and more crucially, we test whether this knowledge generalizes to downstream tasks for which the model has not seen explicit training examples. In particular, we consider semantic role labeling (SRL): given a sentence, can the model correctly identify and assign core frame elements? Notably, during fine-tuning, the model is never shown labeled sentences (only definitions and conceptual relations) so improvements on SRL reflect genuine semantic knowledge transfer. Our results show that the frame-aware LLM not only answers frame-related questions more accurately, but also outperforms the baseline on zero-shot SRL tasks, supporting the claim that structured linguistic knowledge can be effectively injected via prompt-based fine-tuning.

Our main contributions are as follows: 1) We propose a lightweight, LoRA-based method for injecting frame-semantic knowledge into LLMs using structured FrameNet annotations. 2) We design a diverse set of instructional prompting templates and linguistic variations to simulate realistic use cases for frame-role understanding. 3) We provide extensive evaluation on both **seen** and **unseen** frames for zero-shot knowledge probing and SRL inference, demonstrating enhanced interpretability and generalization.

In the rest of the paper, Section 2 reviews related work, Section 3 describes our methodology, Section 4 presents experiments and results, and Section 5 concludes with final remarks and future directions.

2 Background and Related Work

Frame Semantics, introduced by Fillmore (Fillmore, 1976), provides a principled approach to modeling linguistic meaning by organizing words into conceptual structures called *frames*. Each frame represents a prototypical scenario, described by a set of frame elements (FEs), and is evoked by specific lexical units (LUs). The FrameNet project (Baker et al., 1998) operationalizes this theory by cataloguing over 1,000 frames, their core and peripheral elements, and annotated instances.

While Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023) and LLaMA (Dubey et al., 2024) achieve remarkable performance across diverse NLP tasks, they are pre-trained on general web corpora and lack explicit integration of structured linguistic resources like FrameNet. As a result, LLMs may generate fluent yet semantically misaligned outputs when required to interpret or generate language in terms of frame-semantic roles.

Recent research has sought to bridge this gap by augmenting LLMs with frame-semantic knowledge. Fine-tuning LLMs on FrameNet data has been shown to enhance their ability to model semantic structures and improve interpretability (Cui and Swayamdipta, 2024a; Torrent et al., 2022). Several works have proposed injecting frame-level information into transformer architectures to support semantic role labeling (Das and Smith, 2010; Zhang et al., 2023), question answering (Madabushi, 2024), commonsense reasoning (Botschen et al., 2018a; Wang et al., 2021a), and even named entity recognition (Alexiev and Casamayor, 2016). Frame-based representations have also been leveraged for more robust and factually grounded summarization (Han et al., 2016; Guan et al., 2021).

Despite these advances, most prior work either leverages FrameNet solely as a source of annotations for supervised tasks or incorporates frame information as static features. In contrast, our approach aims to *internalize* frame-semantic knowledge by textualizing FrameNet diverse knowledge into instructional prompts for LLM adaptation. Furthermore, we explicitly assess the quality of injected knowledge by evaluating the model not just

on frames used for adaptation, but also on *unseen* frames held out from training, a perspective rarely addressed in prior studies.

In summary, while previous research has demonstrated the benefits of integrating frame-semantic supervision into neural models, there remains a need for approaches that support robust generalization and interpretability via explicit, structured knowledge injection. Our work aims to reduce this gap by proposing a scalable, prompt-based method for frame-semantic adaptation, and by providing a systematic evaluation on both in-domain and out-of-domain (seen/unseen) frames.

Our work also relates to recent efforts in discourse semantics that employ question answering as a tool for evaluating consistency and logical understanding. For example, (Miao et al., 2024) introduce a Socratic QA framework to test whether LLMs respond consistently to logically equivalent or entailed discourse questions, while (Rabinovich et al., 2023) propose QUDeval to measure semantic consistency across related QA pairs grounded in discourse theory. These studies highlight the importance of consistency in QA-based evaluation, which is complementary to our focus on injecting frame-semantic knowledge into LLMs.

3 Injecting Frame-Semantics into LLMs

Our knowledge injection pipeline, illustrated in Figure 1, is designed to make the structured content of FrameNet directly usable by large language models. The process begins with the extraction of frame-level information from FrameNet: for each frame, we collect its definition (a concise description of the scenario the frame represents), its core frame elements (the prototypical participants or roles involved), and the set of lexical units (words or multiword expressions that evoke that frame in context).

To give a concrete example, Table 1 shows the SUPPLY frame: its definition describes a scenario where a "SUPPLIER provides a THEME to a RECIPIENT." The core frame elements here are roles such as SUPPLIER, RECIPIENT, and THEME, each mapping to a participant in this scenario, for instance, "*China*" as the SUPPLIER, "*Iran*" as RECIPIENT and "*decontamination materials*" as the THEME in the sample sentence. The associated lexical units (LUs) are verbs and nouns like "*provide*", "*supply*", or "*equipment*", each capable of triggering the frame in different contexts.

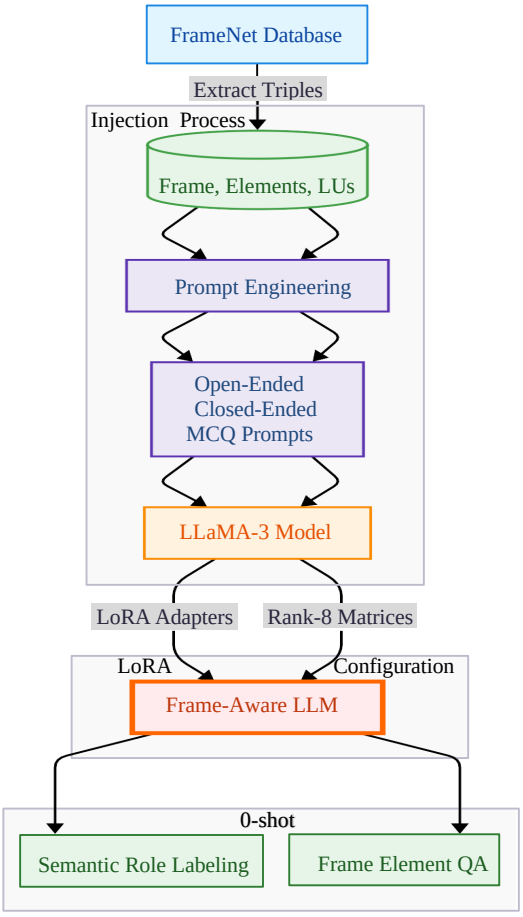

Figure 1: System architecture for frame-semantic knowledge injection into LLaMA-3. The pipeline extracts frame-element-lexical unit triples from FrameNet, converts them into multiple prompt formats (open-ended, closed-ended, MCQ), and fine-tunes the model using LoRA adapters. The resulting Frame-Aware LLM enables zero-shot semantic role labeling and frame element QA.

Frame: SUPPLY		
A SUPPLIER provides a THEME to a RECIPIENT		
FEs	SUPPLIER	Has **China** *provided* Iran with decontamination materials?
	RECIPIENT	Has China *provided* **Iran** with decontamination materials?
	THEME	Has China *provided* Iran **with decontamination materials**?
LUs	*afford.v, equip.v, equipment.n, fix up.v, fuel.v, furnish.v, issue.v, outfit.v, provide.v, provision.n, provision.v, supplier.n, supply.n, supply.v*	

Table 1: Illustration of the SUPPLY frame with annotated frame elements and associated lexical units.

Rather than using FrameNet only as a source for supervised labeling, we transform this structured knowledge into a set of natural language question–answer pairs. For each frame, we generate prompts that ask about its definition, the roles it

contains, or which words evoke it, simulating realistic queries a user or downstream application might pose. Prompts are generated in various formats, including open-ended, closed-ended, and multiple-choice ensuring broad coverage of the theory.

This synthetic QA dataset serves as supervision for fine-tuning a pretrained LLaMA-3 model via Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA is a parameter-efficient fine-tuning technique that augments a frozen pretrained model with small trainable low-rank matrices. During training, only these additional parameters are updated, greatly reducing memory and computational cost while preserving the general linguistic competence already encoded in the model. These properties make LoRA particularly suitable for injecting structured resources like FrameNet, where the large number of frame–role combinations would make full fine-tuning both expensive and prone to catastrophic forgetting, i.e., the overwriting of previously acquired knowledge. By constraining learning to a compact set of additional parameters, LoRA enables the integration of frame-semantic knowledge without erasing the model's broader abilities. In our experiments, we adopt both 3B and 8B LLaMA models, which balance computational feasibility with meaningful evaluation.

This allows the model to internalize the relationships between frame definitions, roles, and lexical units, without relying on explicit sentence-level annotation. After training, the resulting Frame-Aware LLM can be probed on zero-shot tasks such as semantic role labeling and frame-related question answering.

Instructional Template Construction. To effectively inject frame-semantic knowledge into large language models (LLMs), we design natural language templates (Zheng et al., 2023; Su et al., 2021; Wen et al., 2024) that translate structured FrameNet annotations-such as frame definitions, frame elements (FEs), and lexical units (LUs)-into instructive, contextualized prompts. In the *Question: "Can you list some frame elements in the X frame?*" with *Answer: "The frame elements of the X frame are: FE_1, FE_2, and FE_3.*" symbols, such as *X* or FE_1, are placeholders replaced with annotations from FrameNet. This approach builds on the principle that linguistic structure can be aligned with QA-based representations (He et al., 2015), supporting both training as well as augmenting interpretability.

We compose 11 task-specific templates, grouped as follows: six open-ended, four closed-ended, and one multiple-choice (MCQ) format. Each template addresses a distinct aspect of frame-semantics, ranging from recognizing frame elements to identifying lexical units and mapping roles to frames. This diversity enables the model to encounter a wide range of linguistic formulations, enhancing generalization (Ma et al., 2022; Cui and Swayamdipta, 2024b).

Prompt Types:

- **Open-ended Prompts** (6 templates): Encourage free-form, descriptive responses and probe the model's ability to verbalize frame knowledge in its own words. These cover frame definitions (e.g., *What is the definition of the X frame?*), frame elements (e.g., *Can you list some frame elements in the X frame?*), frame element definitions, and lexical units.
  Example: Question: Can you identify a few frame elements or roles in the 'SUPPLY' *frame?*
  *Answer: The frame elements "*SUPPLIER, THEME *and* RECIPIENT*" are associated with the* SUPPLY *frame.*

- **Closed-ended Prompts** (4 templates): Binary (yes/no) or direct verification questions to check specific facts about frames, roles, or lexical units.
  Example: Question: Are the roles 'RECIPIENT' *and* 'THEME' *part of the frame elements of the* 'SUPPLY' *frame?*
  Answer: Yes

- **Multiple-choice Prompts** (1 template): The model selects the correct answer among several options, diagnosing confusion or gaps in understanding.
  Example: Question: Which role is part of the frame elements in the 'SUPPLY' *frame?*
  A) SUPPLIER *B)* RECIPIENT *C)* LOCATION *D)* THEME
  Answer: D) THEME

Linguistic Variations. Template diversity alone is not sufficient to guarantee robustness: a model could simply memorize fixed associations between question forms and answers. To promote generalization, for each template and frame, we systematically construct five alternative phrasings of the question and fifteen variants of the answer. For

instance, for the "list frame elements" template, questions might include: *Which are the roles in the X frame?*, *What are some frame elements defined for X?*, *Who are the core entities in the X frame?*, etc. Answers likewise vary (e.g., *The X frame includes FE_1, FE_2, and FE_3*; *FE_1, FE_2, and FE_3 define the X frame*; and so on).

For each training instance, one question and one answer variant are chosen at random and the pair is used as a supervised example. This strategy, inspired by Dong et al. (Dong et al., 2017), exposes the model to a wide spectrum of natural language formulations and minimizes spurious correlations, crucial for supporting transfer to unseen frames (see Section 4). In this way, the LLM cannot "cheat" by matching surface forms; it must internalize the underlying frame-semantic associations. Taken together, our prompt engineering pipeline, spanning diverse task templates and systematic linguistic variation, supports both the depth and breadth of frame-semantic knowledge acquisition. This methodology improves robustness, interpretability, and aligns the knowledge injection process more closely with the real-world variability of language. To further ensure quality, we manually inspected around 100 generated SRL examples, confirming that the questions and answers were consistent with the intended frame-semantic annotations (Mihaylov et al., 2018).

4 Experimental Evaluation

In this section, we evaluate the effectiveness of our frame-semantic knowledge injection approach for large language models (LLMs). Our experimental objectives are twofold: (1) determine whether the injected knowledge substantially enhances the model's ability to reason about frames, frame elements, and lexical units; and (2) assess whether this acquired semantic knowledge generalizes effectively to practical downstream tasks, most notably, semantic role labeling (SRL)-even without explicit SRL supervision during training.

4.1 Experimental Setup

Our experiments utilize FrameNet version 1.7[1] (Baker et al., 1998), a comprehensive lexical database cataloging over 1,000 semantic frames, their associated core and peripheral frame elements, and lexical units.

[1] https://framenet.icsi.berkeley.edu/frameIndex

Frame Selection and Dataset Composition. For our initial evaluation, we constructed a representative subset of 60 frames from FrameNet, designed to maximize semantic diversity and ensure robust hierarchical coverage (see Appendix A for full criteria and the frame list). The selection process began with a set of core "seed" frames (such as ABANDONMENT, BRINGING, ASSISTANCE, MOTION, and COMMUNICATION) which were chosen to span different domains and frame complexities. From these seeds, we expanded the set by systematically including frames that are hierarchically related, either *inheriting from* or *being inherited by* others within the FrameNet taxonomy. This relational expansion yielded a set of frames that are both semantically coherent and structurally interconnected, capturing the full breadth of frame–element–lexical unit configurations observed in FrameNet. As a result, the final subset covers 175 unique frame elements and 730 lexical units, with frames selected to reflect a broad range of structures (from simple to highly articulated) and to ensure that all major types of frame–element relations and domains are represented. This principled construction ensures the resulting dataset is both challenging and realistic for frame-semantic evaluation.

Instance Sampling Strategy. Naturally, the sampling process differs slightly depending on the type of task. For now, we disregard the additional layer of linguistic variation and focus on the core instance generation procedure. For every open-ended task, the approach is straightforward: for example, when eliciting the definition of a frame or asking which frame corresponds to a given definition, a single core instance is generated per frame (before further expansion via linguistic paraphrasing). However, for tasks that involve frame elements (restricted here to core frame elements) or lexical units, the number of instances per frame directly depends on the number of relevant elements or units present in that frame. In other words, frames with more core frame elements or lexical units will yield proportionally more question–answer pairs for those tasks. Further details on the sampling strategies adopted for frame elements and lexical units are provided in Appendix B. Closed-ended tasks require both positive and negative examples to prevent the model from defaulting to trivial responses (e.g., always answering *no*). Positive samples are created by pairing correct annotations (e.g.,

frame definitions, frame elements) with their respective frames. Negative samples, however, must be selected carefully to avoid class imbalance: using all incorrect annotations would overwhelm the dataset with negatives. To address this, we fix the ratio to $p = 3$ positives and $q = 6$ negatives per frame–task pair. Negatives are drawn from unrelated frames and filtered to avoid duplication. This results in a balanced and informative training signal: $p - 2$ positive and $q - 2$ negative samples are assigned to training, with the rest evenly split across validation and test. Full sampling procedures for specific tasks are detailed in Appendix C. Multiple-choice tasks were constructed with a fixed number of $k = 5$ samples per frame, allocated as $k - 2 = 3$ to the training set, and one each to the validation and test sets. Each MCQ instance presented a single correct answer along with a set of distractors sampled from alternative frames, ensuring that all options were unique and plausible. To increase the challenge and diagnostic value, some prompts included an additional distractor option such as "None of these", following practices proposed in prior work (Yatskar et al., 2016). The training split intentionally contained both positive and fully negative MCQs (i.e., questions with only incorrect options), while the validation and test sets each included one positive and one negative sample per frame to support balanced evaluation.

Synthetic Dataset Construction. From the targeted frames, we systematically generated a total of 6,628 synthetic question–answer pairs, employing linguistically diverse prompt templates (described in Section 3 and exemplified in Appendix D). Following generation, we allocated 3,642 samples to the training set, 1,493 to the development (validation) set, and 1,493 to the test set. Except this split, with same approach a separate set of 1,052 question–answer pairs was generated using 10 unseen frames to evaluate the model's generalization at unseen frame-semantic knowledge. Moreover, no question or answer surface form is ever repeated across different splits, preventing the model from memorizing fixed linguistic patterns. Each of the 11 tasks was instantiated using multiple paraphrased templates for both questions and answers. Specifically, for each task, three distinct question formulations were assigned to the training set, one to the validation set, and one to the test set. Answer templates followed a similar logic: out of a total of 15 available variants per task, 10 were designated as eligible for training (from which 3 were randomly sampled for each frame), 2 were allocated for validation (with one randomly selected), and 3 were reserved for testing (with one randomly selected). This controlled partitioning ensures strict paraphrastic separation across splits, preventing the model from relying on surface-form memorization and encouraging genuine generalization. Concrete examples and the complete set of question–answer paraphrases for a representative frame-based task are provided in Appendix D. In particular, a detailed summary of the prompt types, task formulation strategies, and sampling counts is reported in Table 6.

Evaluating Generalization. To rigorously assess the model's ability to generalize beyond memorization, we adopted a frame-level splitting strategy rather than random sampling: 50 frames were designated as *seen* (utilized for training and validation), while 10 frames were held out as *unseen* and reserved exclusively for zero-shot evaluation. The unseen frames, such as RELEASING, MANIPULATION or CONTROL, were selected to ensure semantic and structural diversity against phenomena not observed during training. Further details on the selection process, as well as the full frame list and distribution across tasks, are provided in Appendix A.

Fine-tuning Configuration. We fine-tuned pretrained LLaMA models-LLaMA 3.2 3B[2] and LLaMA 3.1 8B[3], using Low-Rank Adaptation (LoRA) (Hu et al., 2022), building on the architecture described in (Touvron et al., 2023). LoRA introduces trainable low-rank matrices into the model's attention and feedforward layers, allowing for parameter-efficient adaptation with minimal computational overhead. In our experiments, we used a rank of 16, a scaling factor of $\alpha = 16$, and no dropout. Fine-tuning was carried out using instruction-style prompts consistent with the supervised instruction tuning paradigm (Ouyang et al., 2022). Each input was structured in a standardized format: `### Input: <QUESTION> - ### Response: <ANSWER>`. We employed the Unsloth framework (Daniel Han and team, 2023) to enable efficient fine-tuning with 4-bit quantized weights. Models were trained for 7 epochs using

[2] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

the AdamW optimizer (8-bit), with a learning rate of 2×10^{-4}, a batch size of 4 per device, and 8 gradient accumulation steps. To optimize memory usage, we activated gradient checkpointing and used FP16 or BF16 precision depending on hardware capabilities. Model selection was based on validation loss, evaluated every 50 steps. All fine-tuned LoRA models [4][5] and associated tokenizers have been released on the Hugging Face Hub (Wolf et al., 2020) to facilitate reproducibility.

4.2 Results and discussion

Evaluation of Injected Frame-Semantic Knowledge. To evaluate the effectiveness of our frame-aware supervision strategy, we group the 11 frame-related tasks into three broad categories based on the type of prompt: *Closed-ended*, *Open-ended*, and *Multiple-choice questions (MCQs)*. Each category reflects a different cognitive demand: Closed-ended tasks involve binary decisions (e.g., verifying if a role belongs to a frame), Open-ended tasks require free-form generative responses (e.g., defining a frame or a role), and MCQs present a set of options from which the model must select the correct answer. We further assess performance under two generalization regimes. The first includes the 50 *seen* frames used during training and validation ("In-domain"). The second consists of 10 *unseen* frames ("Out-of-domain"), explicitly held out for zero-shot evaluation to test the model's ability to generalize beyond the training distribution. Results for both the baseline LLaMA models (3.1 8B and 3.2 3B) and their fine-tuned variants are summarized in Table 2. Each prompt category is evaluated using a metric suited to its output type. For both Closed-ended and Multiple-choice (MCQ) tasks, we report the F1 score, which balances precision and recall, effectively capturing the model's ability to make accurate binary and categorical predictions. Although MCQs involve a selection among distractors, their scoring is treated as a binary classification of correctness, hence the use of F1. For Open-ended tasks, which require the model to produce free-form natural language responses (e.g., definitions, descriptions of frame elements), we adopt a semantic similarity metric. Specifically, we compute the cosine similarity between the predicted and reference answers using Sentence-BERT embeddings[6]. This approach, standard in semantic textual similarity evaluation, allows us to assess whether the model captures the intended meaning even when surface forms differ.

Fine-tuned models consistently outperform their pretrained counterparts across all prompt types and model sizes. For instance, the LLaMA 3.1 8B model shows a substantial improvement in F1 score on Closed prompts, rising from 0.55 to 0.93, and in cosine similarity on Open-ended tasks, from 0.64 to 0.87. The gains extend to Multiple-choice questions as well, with F1 increasing from 0.27 to 0.66. The smaller LLaMA 3.2 3B model exhibits similar trends, confirming the robustness of the approach. These results demonstrate that injecting structured frame-semantic supervision significantly enhances the model's ability to understand and reason over semantic roles, definitions, and frame-element associations. Despite not being exposed to ten frames during training, fine-tuned models retain strong performance on these held-out examples. For instance, the 8B model drops only slightly from 0.87 to 0.73 in cosine similarity on Open-ended prompts, and from 0.93 to 0.87 in F1 on Closed ones. This small degradation indicates that the model generalizes well beyond memorization, applying abstract frame-semantic reasoning to novel lexical and conceptual configurations. The original LLaMA models perform consistently worse in all settings. On unseen frames, the baseline 3.2 3B model achieves only 0.33 F1 on MCQs and 0.63 cosine on Open-ended tasks. In contrast, the fine-tuned models maintain substantially higher scores. These discrepancies highlight the necessity of targeted, frame-aware training signals: without them, the model struggles to interpret even well-formed prompts about roles, definitions, or lexical associations. In sum, our results provide strong evidence that explicit semantic supervision-grounded in FrameNet and enhanced by task- and template-level diversity-substantially improves the model's ability to understand and manipulate frame-semantic knowledge. Notably, the generalization observed on zero-shot frames suggests that the learned representations are not only effective but also transferable, paving the way for broader deployment in downstream tasks such as frame disambiguation, SRL, and knowledge-based QA.

[4] https://huggingface.co/shahidrai/llama_3.1_8b/tree/main

[5] https://huggingface.co/shahidrai/llama_3.2_3b_finetuned/tree/main

[6] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

A detailed breakdown of performance across individual tasks is provided in Appendix E.

Model	Pr. Type	Metric	Zero-shot		Fine-tuned	
			In	Out	In	Out
LLaMA	Closed	F1	0.55	0.52	**0.93**	**0.87**
3.1	Open-ended	Cos	0.64	0.63	**0.87**	**0.73**
8B	MCQs	F1	0.27	0.34	**0.66**	**0.63**
LLaMA	Closed	F1	0.49	0.50	**0.91**	**0.86**
3.2	Open-ended	Cos	0.63	0.63	**0.83**	**0.73**
3B	MCQs	F1	0.25	0.33	**0.52**	**0.50**

Table 2: Performance comparison across models and prompt types, grouped by evaluation setting. "In" refers to seen frames; "Out" refers to unseen frames.

Semantic Role Labeling (SRL) Evaluation. To evaluate whether frame-semantic knowledge acquired through our supervision strategy transfers to a practical downstream task, we design a controlled Semantic Role Labeling (SRL) experiment. In this setting, the model is prompted to identify the lexical unit evoking a given frame and extract the associated frame elements expressed in a sentence. We employ a zero-shot prompting strategy inspired by instructional paradigms (Devasier et al., 2025), using structured, natural language instructions (detailed in Appendix 2) rather than fine-tuning on SRL-annotated data. We evaluate on the Open-Sesame dataset[7] (Swayamdipta et al., 2017), which is based on FrameNet and originally released in CoNLL format (Carreras and Màrquez, 2005). The evaluation set contains 371 sentences covering 38 of the 50 frames used during training, totaling 468 annotated instances. While the task setup is not intended to compete with dedicated SRL systems, it provides a diagnostic test bed to verify whether the model can apply definitional and structural knowledge to recognize semantic roles in naturalistic text.

Table 3 breaks down model performance across three increasingly strict evaluation criteria for frame element identification. The first row (Roles Only) considers predictions correct if the role label matches, regardless of span alignment. The second criterion (Roles + Span (25%)) adds a minimum 25% token-level overlap requirement between the predicted and gold spans. The final setting (Roles + Span (75%)) requires a much tighter alignment, with at least 75% span overlap. The fine-tuned model significantly outperforms the baseline in all settings, achieving a fourfold improvement in role-only detection (0.60 vs. 0.14) and similarly large gains in span-aware scoring (e.g., 0.41 vs. 0.10 at 25% threshold). Even under the strictest criterion (75% overlap), it reaches 0.25 F1, far surpassing the baseline's 0.07.

Consider the sentence: *"Has China provided Iran with decontamination materials?"* In this representative example involving the SUPPLY frame, both the base LLaMA model and our fine-tuned version correctly identify *China* as the SUPPLIER. However, the base model incorrectly labels *Iran* as a LOCATION and fails to detect any additional role. In contrast, our fine-tuned model correctly assigns the RECIPIENT role to *Iran* and identifies *materials* as the THEME. Although the predicted span misses part of the full constituent ("with decontamination materials"), it successfully captures the semantic head, which is often sufficient for downstream tasks. This pattern is consistent with our overall results: the fine-tuned model reliably recovers nearly all core roles, in line with the aggregate metrics, but span completeness can occasionally be imprecise.

While our setup is simplified and intentionally scoped, it is noteworthy that the fine-tuned model achieves competitive (if not superior) performance compared to recent LLM-based SRL systems. For example, Cheng et al. (2024) report F1 scores of 0.40 and 0.38 using ChatGPT in a 3-shot setting on *CoNLL-2005 WSJ* and *CoNLL-2012 WSJ*, respectively, and just 0.22 F1 in a zero-shot setting on *CoNLL-2005 WSJ*. In contrast, our model reaches 0.41 F1 in zero-shot SRL, despite being trained on a smaller and more focused dataset comprising only 50 FrameNet frames. This discrepancy can be explained in part by the underlying resource differences: while Cheng et al. (2024) evaluate over PropBank-style predicates, our approach concentrates on a curated subset of FrameNet frames. This narrower scope likely contributes to the higher accuracy, as it allows the model to internalize more structured and semantically grounded knowledge.

Evaluation Criterion	Zero-shot	Fine-tuned
Roles Only	0.14	**0.60**
Roles + Span (25% ov.)	0.10	**0.41**
Roles + Span (75% ov.)	0.07	**0.25**

Table 3: SRL Performance on Frame Element Prediction (F1 Score)

These results confirm that frame-semantic supervision improves both structural role identification and token-level grounding of semantic roles,

[7] https://github.com/swabhs/open-sesame

demonstrating generalization from injected knowledge to real-world SRL inputs. These results, although obtained in a controlled setting and limited to a selected subset of frames, mark a promising first step: their consistency indicates that the approach is robust and generalizable. The natural next step is to scale the fine-tuning procedure to the full FrameNet inventory, a conceptually straightforward extension that merely requires a longer training cycle.

A brief analysis highlights both strengths and weaknesses of the fine-tuned model. At the positive end, qualitative analysis shows clear improvements over the baseline. For instance, in the SUPPLY frame the fine-tuned model correctly recovers all gold-standard roles and spans in: "*Has* [*China*]$_{\text{SUPPLIER}}$ *provided* [*Iran*]$_{\text{RECIPIENT}}$ [*with decontamination materials*]$_{\text{THEME}}$*?*". By contrast, the baseline mislabels the THEME span, predicting [*materials*]$_{\text{MATERIAL}}$ and missing the full constituent. This suggests that frame semantics help the model align roles and spans more faithfully to gold annotations, correcting systematic errors made by the baseline. At the same time, errors remain. In the EXCHANGE frame, sentence "*The Mycenaeans were an acquisitive race who came to conquer, not to trade*", the fine-tuned model hallucinates THEME and RECIPIENT alongside the correct roles EXCHANGER 1 and EXCHANGER 2, inflating false positives. Similarly, in the OBJECTIVE INFLUENCE frame, sentence "*Many Jamaicans head to the States for further education, and the American economic influence on areas such as business investment and planning is growing*", it adds a spurious AREA role where none was annotated. Another common error is predicting roles without spans, which negatively impacts F1. By contrast, in such difficult cases the baseline typically fails to recover any meaningful roles at all.

5 Conclusion and Observations

In this work, we introduced an efficient and principled methodology for injecting structured frame-semantic knowledge into large language models via LoRA-based fine-tuning. By transforming FrameNet resources into instructional prompts, we enabled the model to internalize rich semantic abstractions grounded in linguistic theory. Our experiments demonstrate substantial gains in both frame and role recognition tasks, as well as in zero-shot semantic role labeling (SRL). Importantly, the model exhibits strong generalization to unseen frames, highlighting its ability to abstract beyond surface-level associations and apply learned structures in novel contexts.

These findings suggest that explicitly aligning LLMs with Frame Semantics can meaningfully enhance their semantic behavior, without sacrificing general language capabilities. This opens promising avenues for future research, including scaling to broader frame inventories, by also exploring more refined prompting strategies. Moreover, we will study the overall impact of the proposed adaptation framework on LLM interpretability and reliability in other downstream tasks, like QA and dialogue. Future work could also explore frame-to-frame relations (e.g., inheritance links), which are highly relevant for reasoning tasks such as NLI where entailment often depends on recognizing hierarchical or causal connections between events. In addition, future evaluations should stratify FrameNet QA data to examine which question types (e.g., frame definitions, frame elements, lexical units) drive the observed improvements, and extend the study across multiple large language models to assess the generalizability of frame-semantic knowledge injection beyond a single architecture.

Acknowledgments

We acknowledge financial support from the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

Limitations

This study focuses on a constrained subset of FrameNet frames and tasks, reflecting an intentionally scoped investigation. While our model shows substantial gains in frame-semantic reasoning, several limitations remain. First, it occasionally predicts spurious frame elements, especially in low-resource frames, reducing precision. Second, it often fails to produce accurate spans for correctly identified roles, limiting its effectiveness in span-level SRL. Extending coverage to the full FrameNet inventory and evaluating across additional tasks (e.g., QA, dialogue) are key directions for future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,

Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Vladimir Alexiev and Gerard Casamayor. 2016. Fn goes nif: integrating framenet in the nlp interchange format. In *Proceedings of the LDL 5th Workshop on Linked Data in Linguistics: Managing, Building and Using Linked Language Resources*, pages 1–10.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. Technical Report 98-009, International Computer Science Institute.

Teresa Botschen, Daniil Sorokin, and Iryna Gurevych. 2018a. Frame-and entity-based knowledge for common-sense argumentative reasoning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 90–96.

Theresa Botschen and 1 others. 2018b. Learning to reason with framenet. In *Proceedings of EMNLP*.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164.

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. *arXiv preprint arXiv:2405.06410*.

Xinyue Cui and Swabha Swayamdipta. 2024a. Annotating framenet via structure-conditioned language generation. *arXiv preprint arXiv:2406.04834*.

Xinyue Cui and Swabha Swayamdipta. 2024b. 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. Association for Computational Linguistics.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Dipanjan Das and Noah A. Smith. 2010. A probabilistic frame-semantic parser. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956.

Jacob Devasier, Rishabh Mediratta, and Chengkai Li. 2025. Can llms extract frame-semantic arguments? *arXiv preprint arXiv:2502.12516*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 875–886.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2003. Background to Framenet. *International Journal of Lexicography*, 16(3):235–250.

Charles J Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hu Zhang. 2021. Frame semantics guided network for abstractive sentence summarization. *Knowledge-Based Systems*, 221:106973.

Xu Han, Tao Lv, Zhirui Hu, Xinyan Wang, and Cong Wang. 2016. Text summarization using framenet-based semantic graph model. *Scientific Programming*, 2016(1):5130603.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774. Association for Computational Linguistics.

Harish Madabushi and 1 others. 2024. Frame-based embeddings for coherent question answering. In *Proceedings of EACL*.

Harish Tayyar Madabushi. 2024. Fs-rag: A frame semantics based approach for improved factual accuracy in large language models. *arXiv preprint arXiv:2406.16167*.

Yisong Miao, Ellie Pavlick, Tom Kwiatkowski, Luke Zettlemoyer, and Pradeep Dasigi. 2024. Discursive socratic questioning: Evaluating the faithfulness of language models' understanding of discourse relations. In *Proceedings of the 62nd Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 6134–6153, Bangkok, Thailand. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ella Rabinovich, Samuel Ackerman, Mo Yu, Kun Xu, Sewon Min, and Dan Roth. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the 3rd Workshop on Generation, Evaluation, and Metrics (GEM) at EMNLP 2023*, pages 119–130, Singapore. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 5230–5240. Association for Computational Linguistics.

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

Tiago Timponi Torrent, Ely Edison da Silva Matos, Frederico Belcavello, Marcelo Viridiano, Maucha Andrade Gamonal, Alexandre Diniz da Costa, and Mateus Coutinho Marim. 2022. Representing context in framenet: A multidimensional, multimodal approach. *Frontiers in Psychology*, 13:838441.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2714–2730.

Chenhao Wang, Yubo Chen, Zhipeng Xue, Yang Zhou, and Jun Zhao. 2021a. Cognet: Bridging linguistic knowledge, world knowledge and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 18, pages 16114–16116.

Y Wang and 1 others. 2021b. Knowledge-aware commonsense reasoning with framenet. *Unknown*.

Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. 2024. Perception of knowledge boundary for large language models through semi-open-ended question answering. *arXiv preprint arXiv:2405.14383*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. https://huggingface.co/docs/transformers.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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, 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*, pages 14029–14037. AAAI Press.

A Task-Frame Sample Distribution

In FrameNet, semantic relationships between frames are organized through hierarchical links, primarily defined as *inherits from* and *inherited by*. To maximize semantic diversity and cover a broad range of frame phenomena, we selected 60 FrameNet frames according to the following principles:

- Each frame maintains at least one hierarchical relationship with another (either *inherits from* or *inherited by*), ensuring structural coverage within the FrameNet taxonomy.
- Selection prioritizes frames with diverse numbers of frame elements and lexical units, and spanning multiple FrameNet domains.
- Ten frames were held out as *unseen* for zero-shot evaluation: RELEASING, MANIPULATION, CONTROL, KIDNAPPING, and COMMUNICATION MANNER (see complete list below in Table 8).

The complete set of frames used for prompt generation and model training is listed in Table 9, where each row represents a FrameNet frame and each column corresponds to a template-based task. The columns of Table 9 are ordered and numbered as follows, and match the templates described in Section 3:

1. **Open-ended:** *What is the definition of the X frame?*
2. **Open-ended:** *Which frame is defined by "def(X)"?*
3. **Closed-ended:** *Is "def(X)" regarded as the definition of Frame X?*
4. **Open-ended:** *Can you list some frame elements in the X frame?*
5. **Open-ended:** *Which is the frame involving frame elements such as FE_1 and FE_2?*
6. **Multiple-choice:** *Which one of the following roles belongs to the set of frame elements of the X frame?*
7. **Closed-ended:** *Are roles such as Y and Z part of the frame elements of Frame X?*
8. **Open-ended:** *How is frame element FEi defined in X frame?*
9. **Closed-ended:** *Does the definition of frame element "def(FE_i)" accurately express FE_i in the X frame?*
10. **Open-ended:** *Could you list some lexical units associated with the X frame?*
11. **Closed-ended:** *Can LU_i, as a POS_i, be considered as the lexical unit of the X frame?*

Each cell in the table reports the number of question–answer pairs generated for the corresponding frame–task combination for details on sampling). The final column reports the total number of samples generated for each frame.

Task Type Details:

- **Open-ended Prompts (Tasks 1, 2, 4, 5, 8, 10):** The model provides free-form or descriptive responses, testing the ability to paraphrase and verbalize frame knowledge.
- **Closed-ended Prompts (Tasks 3, 7, 9, 11):** Require binary (yes/no) or direct verification, probing recognition of frame facts or rejections.
- **Multiple-choice Prompts (Task 6):** The model selects the correct answer among candidates, revealing confusion or gaps.

Unseen Frames for Zero-shot Evaluation: To assess the model's performance after fine-tuning, we apply the same sample generation methodology described in Section 3 to a separate set of unseen frames. By using a uniform prompt structure and evaluation format, we isolate the effect of unseen knowledge on model behavior in a controlled setting. Ten frames Table 8 were excluded from training and validation and used only for zero-shot testing:

- **SUICIDE ATTACK** (inherits from ATTACK)
- **MOTION NOISE** (inherits from MOTION)
- **COMMERCE PAY** (inherits from GIVING)
- **RECEIVING** (inherits from GETTING)
- **CORPORAL PUNISHMENT** (inherits from REWARDS AND PUNISHMENTS)
- **COMMUNICATION MANNER** (inherits from COMMUNICATION)

- **KIDNAPPING** (inherits from COMMITTING CRIME)
- **CONTROL** (inherits from OBJECTIVE INFLUENCE)
- **MANIPULATION** (inherits from INTENTIONALLY ACT)
- **RELEASING** (inherits from INTENTIONALLY AFFECT)

These choices ensure challenging and diverse coverage for generalization evaluation.

B Sampling Strategy for Open-ended Tasks

As described in Appendix A, open-ended tasks are sampled according to the specific structure of each prompt.

For tasks that require only a single reference to the frame-such as asking for the definition of a frame (*"What is the definition of the X frame?"*) or for the frame corresponding to a given definition (*"Which frame is defined by "def(X)"?"*)-the sampling process is straightforward. For each frame, we generate a single instance per prompt type, later augmented through linguistic variation in the main dataset.

In contrast, for open-ended tasks that involve frame elements or lexical units, sampling is more nuanced due to the multiplicity of possible elements within each frame. For example, in prompts like *"Can you list some frame elements in the X frame?"* and *"Which is the frame involving frame elements such as FE_1 and FE_2?"*, the core frame elements are randomly sampled or grouped, ensuring that not all elements always appear in the same order or configuration.

A particularly important case is the prompt *"How is frame element FE_i defined in the X frame?"*. Here, the number of generated samples is determined by the number of core frame elements associated with each frame. To ensure coverage while controlling dataset size, we generate $p = 3$ variations for each core frame element, allocating $p - 2$ samples to the training set, and one each to the validation and test sets.

For prompts targeting lexical units, such as *"Could you list some lexical units associated with the X frame?"*, we employ the iterative prompting technique from (Wang et al., 2022). The number of samples generated depends on the number of available lexical units, grouped by part of speech (POS; e.g., verbs, nouns, adjectives) within each frame. Specifically, let n_{POS} be the number of lexical units for a given POS. We partition the list of lexical units into sub lists, each containing at most five items. The number of samples s_{POS} for each POS is thus computed as

$$s_{\text{POS}} = \left\lceil \frac{n_{\text{POS}}}{5} \right\rceil$$

where $\lceil \cdot \rceil$ denotes the ceiling function.

For instance, if the frame ARRIVING includes $n_{\text{verb}} = 15$ verb lexical units, then

$$s_{\text{verb}} = \left\lceil \frac{15}{5} \right\rceil = 3$$

resulting in three samples for verbs. If there are $n_{\text{noun}} = 8$ noun lexical units, then

$$s_{\text{noun}} = \left\lceil \frac{8}{5} \right\rceil = 2$$

so two noun samples are produced.

For evaluation, the complete set of lexical units per part of speech is exhaustively covered, with three sample sets (each reflecting a different linguistic variation) distributed across training, validation, and test splits. This guarantees that all lexical units are sampled without repetition or overlap between sets (Fillmore et al., 2003).

C Sampling Strategy for Closed-ended Tasks

Closed-ended tasks require a balanced set of positive and negative examples to support meaningful learning and avoid degenerate behaviors (e.g., always predicting *no*). For tasks such as *"Does the definition of frame element def(FE_i) accurately express FE_i in the X frame?"*, we generate $k = p + q = 9$ samples per frame element FE_i, where $p = 3$ are positive and $q = 6$ are negative.

Negative examples are created by pairing the target frame with distractor definitions, elements, or lexical units sampled from unrelated frames (e.g., using definitions from ATTACK when evaluating GESTURE). This ensures diversity while avoiding overwhelming the model with negatives. To preserve task balance, $p-2$ positive and $q-2$ negative samples are included in the training set, with the remaining examples evenly split between validation and test.

For tasks involving lexical unit verification, such as *"Can LU_i, as a POS_i, be considered a lexical*

unit of the X frame?", samples are generated for each sublist of five lexical units. Given a part of speech POS_i, the number of such sublists is computed as:

$$s_{\text{POS}_i} = \left\lceil \frac{n_i}{5} \right\rceil$$

where n_i is the number of lexical units with POS equal to POS_i in the given frame. For example, in the frame ARRIVING, if there are $n_{\text{verb}} = 15$ verb lexical units:

$$s_{\text{verb}} = \left\lceil \frac{15}{5} \right\rceil = 3$$

then the total number of samples for that POS is:

$$s_{\text{total}} = s_{\text{verb}} \times k = 3 \times 9 = 27$$

Similarly, for $n_{\text{noun}} = 8$, we compute:

$$s_{\text{noun}} = \left\lceil \frac{8}{5} \right\rceil = 2 \quad \Rightarrow \quad s_{\text{noun, total}} = 2 \times 9 = 18$$

This sampling strategy ensures a consistent balance of examples across frames and tasks, while maintaining semantic relevance and avoiding annotation redundancy.

D Linguistic Variations

A key aspect of our data construction process is the use of diverse linguistic templates for both questions and answers. Each template contains placeholders—such as X for the frame name or $\text{def}(X)$ for the frame definition—that are instantiated using FrameNet annotations during prompt generation. This approach promotes generalization, prevents the model from memorizing fixed surface forms, and closely mirrors the variability found in real-world user queries.

Tables 4 and 5 provide concrete examples of linguistic variation for a representative open-ended task: frame definition. For each data split (training, validation, test), we sample distinct phrasings, ensuring that the same question or answer formulation is never shared across different splits. This careful partitioning avoids data leakage and tests the model's ability to generalize across different linguistic realizations.

Training
Q1: What is the definition of the X frame?
Q2: Can you define the X frame?
Q3: How is the X frame defined?
Validation
Q4: Could you provide the definition of the X frame?
Test
Q5: Please can you provide the definition of the X frame?

Table 4: Examples of question template variations for the frame definition task, grouped by data split. Each formulation is unique to a split to ensure maximal linguistic diversity and strict separation between training, validation, and test sets.

Training
Ans1: We can define the X frame as "def(X)."
Ans2: The X frame can be defined as "def(X)."
Ans3: The definition of the X frame is "def(X)."
Ans4: If we define the X frame, it would be "def(X)."
Ans5: We can define the X frame as "def(X)."
Ans6: The X frame can be defined as follows: "def(X)."
Ans7: We can outline the definition of the X frame as "def(X)."
Ans8: The X frame is defined as "def(X)."
Ans9: The X frame can be described as "def(X)."
Ans10: The definition of the X frame is "def(X)."
Validation
Ans11: The X frame can be summarized as "def(X)."
Ans12: If we describe the X frame, it will be "def(X)."
Ans13: A possible definition of the X frame is "def(X)."
Test
Ans14: The definition of the X frame can be "def(X)."
Ans15: We could define the X frame as "def(X)."

Table 5: Examples of answer template variations for the frame definition task, grouped by data split. As with the questions, each answer formulation is assigned to a single split, ensuring the model cannot rely on surface-level memorization.

E Task-level Evaluation Breakdown

To better understand where our frame-aware supervision strategy yields the most impact, we provide a task-level breakdown of performance. Table 7 reports scores on each of the 11 prompt-based tasks, comparing the original LLaMA model (zero-shot) with its fine-tuned counterpart. Task types are categorized as Open-ended, Closed-ended, or Multiple-choice (MCQs), and evaluated using appropriate metrics: cosine similarity for generative outputs, and F1 score for classification tasks.

The results consistently confirm the effectiveness of semantic supervision: for each task, the fine-tuned model outperforms the baseline. Particularly notable gains are observed in closed tasks requiring precise frame-role or definition-role verification (e.g., T3, T7, T9, T11), and in open-ended

ID	Prompt type	What varies?	Instances per frame	Why that number of instances?
T1	Open-ended	Wording of question about frame definition	5	One definition, asked in 5 paraphrased forms
T2	Open-ended	Wording of question about frame name from definition	5	One definition, reversed as 5 distinct questions
T3	Closed-ended	Match/mismatch of frame definitions	9 (3+6)	3 correct, 6 distractors from unrelated frames
T4	Open-ended	Surface forms of request for FE list	5	FE list is fixed, asked in 5 paraphrased forms
T5	Open-ended	Subset of core FEs and question formulation	5	2 FEs randomly sampled; question paraphrased 5 times
T6	Multiple-choice	Set of distractors for correct FE	5	Each with 1 correct + 3–4 distractors; mix of correct/incorrect MCQs
T7	Closed-ended	Pairings of roles with frames	9 (3+6)	3 true role sets, 6 sampled from unrelated frames
T8	Open-ended	Question formulation per core FE	$3 \times \lvert\text{core FE}\rvert$	3 paraphrases per core FE definition
T9	Closed-ended	FE–definition pairs	$9 \times \lvert\text{core FE}\rvert$	3 correct, 6 incorrect per FE
T10	Open-ended	POS-based sublists of LUs	$\sum_p \lceil \frac{n_p}{5} \rceil$	One question per LU-POS bucket; no overlap across splits
T11	Closed-ended	LU–POS verification questions	$9 \times \lceil \frac{n_p}{5} \rceil$	3 positives, 6 negatives per LU-POS bucket

Table 6: Summary of task-specific generation strategies and sample counts per frame. Task IDs correspond to the columns of Table 9.

ID	Task	Type	Metrics	Os	FT
T1	What is the definition of the X frame?	Open-ended	Cos	0.70	0.96
T2	Which frame is defined by def(X)?	Open-ended	Cos	0.50	0.84
T3	Is "def(X)" regarded as the definition of Frame X?	Closed	F1	0.48	0.70
T4	Can you list some frame elements in the X frame?	Open-ended	Cos	0.58	0.88
T5	Which is the frame involving frame elements such as FE_1 and FE_2?	Open-ended	Cos	0.47	0.83
T6	Which one of the following roles belongs to the set of frame elements of the X frame?	MCQs	F1	0.27	0.66
T7	Are roles such as Y and Z part of the frame elements of Frame X?	Closed	F1	0.61	0.85
T8	How is frame element FE_i defined in X frame?	Open-ended	Cos	0.68	0.87
T9	Does the definition of frame element "def(FE_i)" accurately express FE_i in the X frame?	Closed	F1	0.51	0.99
T10	Could you list some lexical units associated with the X frame?	Open-ended	Cos	0.75	0.85
T11	Can LU_i, as a POS_i, be considered as the lexical unit of the X frame?	Closed	F1	0.60	0.98

Table 7: Task-wise evaluation results across the 11 prompt templates. Metrics are cosine similarity for open-ended prompts and F1 score for closed-ended and MCQ formats.

prompts involving structured natural language responses (e.g., T1, T5, T8).

F Instruction-style Prompt Used for SRL Evaluation

We evaluate our model's semantic role labeling (SRL) capabilities using a controlled instruction-style prompt, shown in Figure 2. The prompt requires the model to extract both the lexical unit evoking the frame and the associated frame elements, returning a structured `JSON` object. The input consists of a sentence and its corresponding frame label.

Although the prompt includes a single illustrative example, this is not intended as one-shot learning: the example solely clarifies the expected output format and does not correspond to the frame used in the actual input. A true one-shot setting would require frame-specific exemplars for each evaluation case, which are not provided. Thus, the evaluation remains fully zero-shot with respect to frame-specific role assignments.

```
"""
You are an expert in Frame Semantics and Semantic Role Labeling. Your task is to identify the **
     lexical unit** evoking a given frame  and extract the corresponding **frame elements** with
     their roles from a given sentence.

### Instructions:
1. Identify the **lexical unit** that evokes the given frame.
2. Extract **frame elements** present in the sentence and map them to their respective roles
3. Format your response strictly as a JSON object following the structure provided.
4. Do not include any additional explanations-return only the JSON.
5. Use only **frame elements** you know.

### Example:
#### Given Frame: LOCATION
#### Input Sentence:
Hall, who recently returned from a trip to Iraq....
#### Expected Output:
{
    "input_sentence": "Hall, who recently returned from....",
    "annotations": [
        {
            "frame": "LOCATION",
            "lexical_unit": "trip",
            "frame_elements": {
                "PLACE": "Iraq",
                "TRAVELER": "Hall"
            }
        }
    ]
}
Now, process the following input and return a JSON object:

#### Given Frame:
#### Input Sentence:
#### Your Output:
"""
```

Figure 2: Instructional prompt used for SRL evaluation.

Frame Name	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	**Total**
RELEASING	5	5	9	5	5	5	9	9	27	2	18	99
MANIPULATION	5	5	9	5	5	5	9	9	27	2	18	99
CONTROL	5	5	9	5	5	5	9	21	63	2	18	147
KIDNAPPING	5	5	9	5	5	5	9	6	18	3	27	97
COMMUNICATION MANNER	5	5	9	5	5	5	9	9	27	2	18	99
CORPORAL PUNISHMENT	5	5	9	5	5	5	9	9	27	2	18	99
SUICIDE ATTACK	5	5	9	5	5	5	9	6	18	1	9	77
MOTION NOISE	5	5	9	5	5	5	9	15	45	1	9	113
COMMERCE PAY	5	5	9	5	5	5	9	15	45	2	18	123
RECEIVING	5	5	9	5	5	5	9	9	27	2	18	99
Total	50	50	90	50	50	50	90	108	324	19	171	1052

Table 8: Summary of the sample counts produced for each task across the unseen frames.

Frame Name	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	**Total**
ABANDONMENT	5	5	9	5	5	5	9	6	18	12	27	106
ABUSING	5	5	9	5	5	5	9	6	18	12	27	106
APPOINTING	5	5	9	5	5	5	9	15	45	9	27	139
ARREST	5	5	9	5	5	5	9	12	36	9	27	127
ARRIVING	5	5	9	5	5	5	9	6	18	11	45	123
ASSEMBLE	5	5	9	5	5	5	9	12	36	4	9	104
ASSISTANCE	5	5	9	5	5	5	9	12	36	13	36	140
ATTACK	5	5	9	5	5	5	9	6	18	18	81	166
ATTEMPT MEANS	5	5	9	5	5	5	9	9	27	4	9	92
BRINGING	5	5	9	5	5	5	9	21	63	17	72	216
COME TOGETHER	5	5	9	5	5	5	9	12	36	6	27	124
COMMITTING CRIME	5	5	9	5	5	5	9	6	18	8	18	93
COMMUNICATION	5	5	9	5	5	5	9	12	36	10	36	137
COMMUNICATION RESPONSE	5	5	9	5	5	5	9	15	45	10	36	149
CONTACTING	5	5	9	5	5	5	9	15	45	12	54	169
DEPARTING	5	5	9	5	5	5	9	6	18	9	27	103
ENFORCING	5	5	9	5	5	5	9	9	27	8	18	105
ESCAPING	5	5	9	5	5	5	9	6	18	9	27	103
EVENT	5	5	9	5	5	5	9	9	27	8	18	105
EVENTIVE AFFECTING	5	5	9	5	5	5	9	6	18	5	18	90
EXAMINATION	5	5	9	5	5	5	9	15	45	8	18	129
EXCHANGE	5	5	9	5	5	5	9	6	18	8	18	93
EXECUTE PLAN	5	5	9	5	5	5	9	9	27	8	18	105
EXECUTION	5	5	9	5	5	5	9	6	18	9	27	103
FUNDING	5	5	9	5	5	5	9	9	27	4	9	92
GESTURE	5	5	9	5	5	5	9	15	45	9	27	139
GETTING	5	5	9	5	5	5	9	6	18	10	36	113
GIVING	5	5	9	5	5	5	9	9	27	12	54	145
INTENTIONALLY ACT	5	5	9	5	5	5	9	6	18	10	36	113
INTENTIONALLY AFFECT	5	5	9	5	5	5	9	9	27	4	9	92
KILLING	5	5	9	5	5	5	9	15	45	24	135	262
MOTION	5	5	9	5	5	5	9	21	63	8	45	180
OBJECTIVE INFLUENCE	5	5	9	5	5	5	9	21	63	8	18	153
PIRACY	5	5	9	5	5	5	9	9	27	12	27	118
RAPE	5	5	9	5	5	5	9	9	27	12	27	118
REPLACING	5	5	9	5	5	5	9	9	27	9	27	115
RESIDENCE	5	5	9	5	5	5	9	9	27	15	54	148
RESPONSE	5	5	9	5	5	5	9	12	36	8	18	117
REWARDS AND PUNISHMENTS	5	5	9	5	5	5	9	12	36	12	27	130
SELF MOTION	5	5	9	5	5	5	9	18	54	45	297	457
SMUGGLING	5	5	9	5	5	5	9	15	45	12	27	142
SUMMARIZING	5	5	9	5	5	5	9	9	27	8	18	105
SUPPLY	5	5	9	5	5	5	9	12	36	9	27	127
SUPPORTING	5	5	9	5	5	5	9	6	18	4	9	80
TAKING	5	5	9	5	5	5	9	9	27	8	18	105
TEMPORARY STAY	5	5	9	5	5	5	9	12	36	9	27	127
THEFT	5	5	9	5	5	5	9	12	36	20	99	210
USING	5	5	9	5	5	5	9	12	36	13	36	140
VEHICLE LANDING	5	5	9	5	5	5	9	6	18	4	9	80
VISITING	5	5	9	5	5	5	9	6	18	8	18	93
Total	250	250	450	250	250	250	450	525	1,575	524	1,854	6,628

Table 9: Overview of the number of samples generated for each task across frames, with tasks represented as columns and frames as rows. The table also includes the total number of sample pairs for each frame. Each cell reflects the actual number of QA pairs generated, which may vary according to (a) the number of frame elements or lexical units per frame, (b) the mix of positive and negative samples, and (c) the paraphrasing strategy adopted for data splitting. These design choices are fully detailed in Appendix B (for open-ended tasks) and Appendix C (for closed-ended tasks)