On the Continued Value of Universal Dependencies in the Era of Large Language Models

Wenxi Li[1,2] and **Jingyu Peng**[3]
[1]School of the Chinese Nation Studies & School of Liberal Arts, Minzu University of China
[2]Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE
[3]City University of Hong Kong
Correspondence: liwenxi@pku.edu.cn

Abstract

The necessity of explicit linguistic representations has been increasingly questioned in the era of large language models (LLMs). In this work, we revisit this issue using Universal Dependencies (UD) as a case study, examining whether and in what ways this cross-lingual syntactic framework can still benefit contemporary LLMs. We focus on a cross-lingual adversarial paraphrase identification task that is designed to foreground the role of syntactic structure in semantic interpretation across languages. Within this setting, we systematically evaluate three strategies for integrating UD into LLMs: UD-Prompt, UD-Tuning, and UD-Attention. Our experiments show that, although the magnitude of gains depends on how UD-based structural priors interact with model behavior and cross-lingual variation, UD-augmented models consistently outperform their syntax-agnostic counterparts. Across strategies, we observe average accuracy improvements of 2.67%, 8.24%, and 2.53%, respectively. These findings demonstrate that linguistic knowledge remains informative for LLMs, offering practical value in cross-lingual settings where structural alignment is challenging.

1 Introduction

Large language models (LLMs) have achieved remarkable success across a wide range of natural language processing tasks. Trained on massive corpora using self-supervised objectives, modern LLMs are assumed to implicitly acquire syntactic and semantic regularities of languages, leading to a growing belief that explicit structured linguistic frameworks may no longer be necessary. Yet, it still remains unclear whether this assumption truly holds in practice, particularly in cross-lingual contexts where structural divergence across languages presents a unique challenge that may still benefit from explicit syntactic modeling.

This work revisits this question through a cross-lingual adversarial paraphrase identification (CAPI) task, employing Universal Dependencies (UD; Nivre et al., 2016, 2020) as a case study. As illustrated in Figure 1, CAPI presents a rigorous evaluation setting where surface-level cross-lingual semantic matching is insufficient. The German and Chinese counterparts are constructed to preserve strong word-level alignment with the English source while inducing divergent semantic interpretations. This design yields challenging adversarial examples for LLMs: models that rely on spurious lexical cues are likely to fail when syntactic alignment is perturbed. We argue that this makes the CAPI task a principled testbed for assessing whether UD, a language-neutral framework for cross-lingual structural modeling, continues to provide benefits to contemporary LLMs.

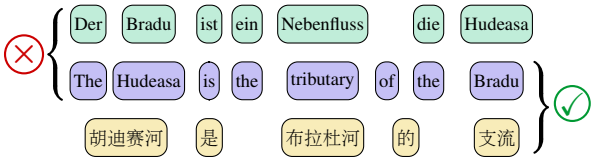


Figure 1: Sentence pair examples in the CAPI task. The check mark indicates semantic equivalence, while the cross indicates non-equivalence.

We propose three strategies for integrating UD into LLMs, enabling a systematic and fine-grained examination of how explicit syntactic structure influences model performance. These strategies span progressively stronger forms of intervention.

- **UD-Prompt**: augments model inputs with descriptions of UD relations, introducing syntactic information in a training-free manner.
- **UD-Tuning**: conducts supervised fine-tuning on UD-aware data, encouraging the model to internalize UD knowledge for inference.
- **UD-Attention**: enforces a syntactic inductive

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17650–17662

bias by modulating the self-attention mechanism with UD-derived structural constraints.

By evaluating this spectrum of strategies on the CAPI task, we aim to identify the conditions under which structural priors provided by UD yield performance gains in modern LLMs.

Our experimental results show that incorporating UD information consistently improves LLM performance across all the integration strategies: UD-Prompt yields an average accuracy gain of 2.67% across multiple models; UD-Tuning produces a particularly large improvement, 8.24%, in the instruction-tuning setting; and UD-Attention delivers a 2.53% accuracy increase across language pairs. Also, a strong correlation between UD-based structural similarities and downstream accuracy is observed, indicating that LLM performance is closely tied to the degree of syntactic similarity across languages. Taken together, we believe that these results demonstrate that UD remains informative and beneficial for LLMs, though the magnitude of its gains depends on a complex interaction between model capacity, learning paradigm, and cross-lingual variations.

2 Related Work

2.1 Linguistic Representations in Pretrained Language Model

Prior to the emergence of pretrained language models, explicit linguistic representations — such as syntactic trees, dependency graphs, and semantic roles — are widely integrated into neural architectures to improve generalization and interpretability. Early work incorporates constituency or dependency parses into recurrent and convolutional models using graph-based encoders or tree-structured networks, and demonstrates improvements on tasks including parsing, semantic role labeling, and natural language inference (Socher et al., 2013; Marcheggiani and Titov, 2017; Bastings et al., 2017). These studies show that structured linguistic inductive biases are particularly beneficial in low-resource settings and in languages with complex syntactic phenomena.

With the advent of pretrained transformer-based models, explicit linguistic supervision is largely replaced by end-to-end representation learning. A substantial body of probing research shows that such models implicitly encode a broad range of syntactic and semantic information, including part-of-speech tags, dependency relations, and hierarchical structure (Hewitt and Manning, 2019; Tenney et al., 2019; Chi et al., 2020; Jawahar et al., 2019). These findings contribute to the prevailing perception that explicit linguistic representations may be unnecessary for modern neural language models.

Despite these strong implicit representations, a growing line of work revisits the role of explicitly incorporating syntactic structures. This research demonstrates that leveraging linguistic information can still yield measurable gains on downstream tasks. For instance, syntax-aware attention mechanisms improve sentence-level semantic understanding (Bai et al., 2021; Ahmad et al., 2021; Liu et al., 2020) and machine translation (Zhang et al., 2019a; Deguchi et al., 2019; Egea Gómez et al., 2021). Similar benefits are observed for semantic role labeling (Strubell et al., 2018; Xia et al., 2019; Zhang et al., 2019b) and relation extraction (Jafari et al., 2021; Tian et al., 2022), suggesting that explicit syntactic cues can complement learned representations in pretrained models.

2.2 Linguistic Knowledge in Large Language Model

The rise of LLMs raises renewed questions about the role of explicit linguistic structure at scale. Although LLMs implicitly capture substantial linguistic knowledge, recent studies suggest that this knowledge is not always applied reliably. In particular, work on model consistency shows that LLMs often produce divergent outputs when prompted with syntax-preserving or syntactically altered paraphrases, indicating that prompt syntax alone can significantly affect factual knowledge retrieval (Elazar et al., 2021; Linzbach et al., 2024).

Motivated by these findings, recent work has explored injecting linguistic knowledge into LLMs through prompting, where syntactic trees and semantic graphs are linearized and appended to the input to assess their impact on model behavior. For example, Swarup et al. (2025) show that incorporating constituency trees, dependency parses, and semantic role labels into prompts improves extraction robustness. Similarly, Jin et al. (2024) and Raut et al. (2025) investigate Abstract Meaning Representation (AMR; Banarescu et al., 2013), a graph-based semantic formalism, and find that LLMs can condition on such structured representations, although effectiveness varies substantially across tasks and prompt formulations. Building on this direction, Zhang et al. (2025) reformulate AMR structures as natural-language explanations,

leveraging the instruction-following capabilities of LLMs and yielding further performance gains.

Recent work has also explored incorporating linguistic knowledge into LLMs through parameter-efficient fine-tuning, motivated by the limitation that prompt-based methods are sensitive to prompt design and do not guarantee that structural information is consistently internalized. An important development in this direction is SR-LLM (Zhang et al., 2025), which constructs a hybrid instruction-tuning dataset mixing text-only and structure-augmented examples, enabling the model to jointly learn task-following behavior and structure–task associations during training.

2.3 Positioning of This Work

Inspired by prior research, we propose and systematically explore a range of strategies for integrating UD into LLMs. We unify and evaluate these strategies within a single experimental framework, providing a comprehensive comparison of UD integration methods. Moreover, to the best of our knowledge, this is the first work to examine the effectiveness of linguistic knowledge in a cross-lingual setting, in contrast to existing studies focusing on monolingual benchmarks.

3 Task: Cross-lingual Adversarial Paraphrase Identification

3.1 Definition

In the standard paraphrase identification, the task is to determine whether a pair of sentences in one language conveys the same meaning. In contrast, the CAPI task considered in this work is both *cross-lingual* and *adversarial* by design. Each instance consists of a *cross-lingual* sentence pair $(s^{\text{EN}}, s^{\text{Non-EN}})$, where s^{EN} is an English sentence and $s^{\text{Non-EN}}$ is its counterpart in a non-English language. Moreover, the task is *adversarial* since it includes sentence pairs that diverge in meaning despite exhibiting high lexical similarity.

3.2 Why UD Matters

For adversarial sentence pairs, semantic equivalence cannot be reliably inferred from lexical overlap alone. This challenge is further exacerbated in cross-lingual settings, where languages differ substantially in their surface realizations:

- **Word order** (e.g., SVO vs. SOV),
- **Argument realization** (e.g., case marking vs. positional encoding),
- **Functional constructions** (e.g., auxiliaries and light verbs).

As illustrated in Figure 2, UD abstracts away from such surface-level linguistic variations and instead encodes grammatical relations — such as NSUBJ, OBJ, and ADVCL — within a unified and cross-lingually consistent framework, enabling more reliable alignment of semantic content across languages. We therefore hypothesize that providing LLMs with access to UD-based analysis can improve their performance on the CAPI task.

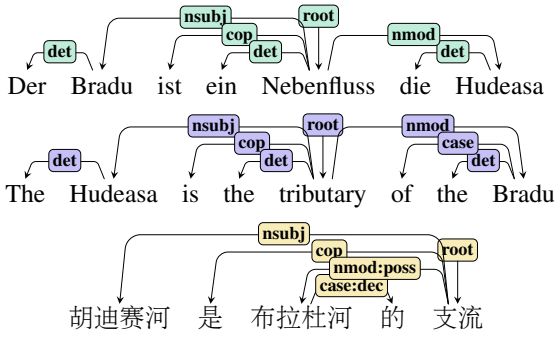


Figure 2: UD analyses for the sentence pair example.

3.3 Dataset

We use PAWS-X (Yang et al., 2019) as the source dataset. PAWS-X is a multilingual extension of PAWS and contains adversarial sentence pairs in seven languages: English (EN), French (FR), Spanish (ES), German (DE), Chinese (ZH), Japanese (JA), and Korean (KO). In the original PAWS-X benchmark, however, each instance consists of a sentence pair written in a *single* language.

In this work, we convert PAWS-X into a cross-lingual paraphrase identification dataset. This transformation is enabled by the fact that all non-English instances in PAWS-X are obtained by translating English sentences, resulting in parallel sentence structures across languages. Leveraging this property, we reorganize the data: for each English sentence pair $(s_1^{\text{EN}}, s_2^{\text{EN}})$, we match each sentence with its translation in a target language, i.e., $(s_1^{\text{Non-EN}}, s_2^{\text{Non-EN}})$. If the original English pair is labeled as a paraphrase, we then construct two cross-lingual paraphrase pairs: $(s_1^{\text{EN}}, s_1^{\text{Non-EN}})$ and $(s_2^{\text{EN}}, s_2^{\text{Non-EN}})$. Conversely, if the original pair is labeled as a non-paraphrase, we generate two cross-lingual non-paraphrase pairs: $(s_1^{\text{EN}}, s_2^{\text{Non-EN}})$ and

$(s_2^{\text{EN}}, s_1^{\text{Non-EN}})$. This reorganization preserves the original labels while ensuring that each constructed pair spans two languages. The final dataset contains 22176 instances, of which 42.8% are labeled as paraphrases and 57.2% as non-paraphrases.

4 Methods

We investigate three strategies for integrating UD into LLMs: UD-Prompt, UD-Tuning and UD-Attention. Together, they span a spectrum from lightweight prompt-based augmentation to parameter-efficient syntactic adaptation.

4.1 UD-Prompt: Incorporating Universal Dependencies through Prompting

UD-Prompt integrates UD information directly into the prompt of an LLM. We consider two alternative representations of UD within this framework (see more details in Appendix A.1).

The first variant appends a *serialized dependency representation* to the original sentence. Given an input sentence, its UD parse is converted into a linearized list of dependency arcs, which is then concatenated with the sentence as additional contextual information. The second variant adopts a *natural-language realization* of UD, where each dependency relation is converted into a declarative natural-language statement that verbalizes the underlying syntactic relation. Formally, we define a UD-to-natural-language transformation function:

$$f_{\text{UD-NL}} : G \rightarrow \mathcal{T},$$

where $G = (V, E)$ is a directed labeled dependency graph, with $V = \{w_1, \ldots, w_n\}$ denoting the set of tokens and $E \subseteq V \times \mathcal{R} \times V$ denoting the set of dependency arcs, and $\mathcal{R}$ the set of UD relation labels. The output $\mathcal{T} = \{t_1, \ldots, t_{|E|}\}$ is a set of natural-language sentences. For each dependency arc $(w_i, r, w_j) \in E$, the function generates a sentence t_k using a predefined, relation-specific template. The resulting descriptions $\mathcal{T}$ is then provided to the model alongside the original sentence.

$(w_i, \text{NSUBJ}, w_j) \mapsto$ "The nominal subject of w_j is w_i."

4.2 UD-Tuning: Leveraging Universal Dependencies in Fine-Tuning

UD-Tuning investigates the role of UD information as an explicit source of syntactic supervision in training-dependent, task-specific fine-tuning. UD is incorporated by augmenting the input with verbalized dependency relations, while a text-only configuration serves as a baseline. To fully characterize the impact of this supervision, we evaluate UD-Tuning under two contrasting paradigms: *classification-based fine-tuning* (discriminative) and *instruction-based fine-tuning* (generative).

In the classification-based setting, paraphrase identification is framed as a binary sequence classification problem: given a sentence pair, the model predicts whether the two sentences convey the same meaning. The LLM is then equipped with a task-specific classification head and fine-tuned on the paraphrase detection objective, optionally with UD analyses. To address class imbalance, we employ a weighted cross-entropy loss, where w_{y_i} denotes the class-specific weight associated with the gold label y_i for input x_i:

$$\mathcal{L}_{\text{cls}} = -\sum_i w_{y_i} \log p(y_i \mid x_i)$$

By contrast, instruction-based tuning casts paraphrase identification as a conditional text generation task. A causal language model is fine-tuned to produce a natural-language decision conditioned on a task instruction that includes the sentence pair, and optionally, UD information. The loss is applied only to the answer tokens, while prompt tokens are masked. Given an input sequence $\mathbf{z} = [\mathbf{p}, \mathbf{a}]$, where $\mathbf{p}$ is the instruction prompt and $\mathbf{a}$ is the answer span, the objective is:

$$\mathcal{L}_{\text{ins}} = -\sum_{t \in \mathbf{a}} \log p(z_t \mid z_{<t})$$

4.3 UD-Attention: Modulating Attention with UD-Guided Matrices

Building on prior work that integrates syntactic structure into the attention mechanisms of pre-trained language models (Bai et al., 2021; Ahmad et al., 2021), we propose UD-Attention, a parameter-efficient approach that injects UD structures directly into the computation of self-attention. Whereas UD-Tuning relies on the model to implicitly internalize structural patterns during fine-tuning, UD-Attention explicitly introduces a syntactic affinity matrix derived from UD.

Our construction of this matrix broadly follows the formulation of Li (2025). Consider a sentence pair $S^{\text{EN}} = \{w_1^{(1)}, \ldots, w_m^{(1)}\}$ and $S^{\text{Non-EN}} = \{w_1^{(2)}, \ldots, w_n^{(2)}\}$. Their UD parses are first converted into hypergraph representations. A UD-guided similarity matrix is then computed by iteratively comparing hyperedges $e_m^{(1)}$ and $e_n^{(2)}$ headed

Algorithm 1 Construct_UD_Matrix

Require: UD parses $P^{(1)}$, $P^{(2)}$
Ensure: UD-guided matrix M
Convert $P^{(1)}$ and $P^{(2)}$ into hypergraphs
Extract hypernodes and hyperedges
for each hyperedge $e_i^{(1)}$ headed by $w_i^{(1)}$ **do**
  for each hyperedge $e_j^{(2)}$ headed by $w_j^{(2)}$ **do**
    compute Sim_N (head-node similarity)
    compute Sim_E (edge-level similarity)
    $Sim_{ij} \leftarrow Sim_N, Sim_E, h_{ij}$
  end for
end for
return M

by tokens in the two sentences using a similarity function Sim (outlined in Algorithm 1). The resulting matrix $M \in \mathbb{R}^{m \times n}$ is defined as:

$$M_{mn} = \text{Sim}(e_m^{(1)}, e_n^{(2)})$$

UD-Attention then integrates the UD-derived matrix M into the self-attention mechanism. Let A denote the standard scaled dot-product attention score matrix, where Q and K are the query and key projections and d is the key dimensionality. We inject UD by modulating the attention scores with M in an element-wise manner, controlled by a learnable gating scalar γ. This design enables the model to dynamically regulate when and to what extent the UD signal influences attention, rather than imposing M as a fixed structural prior.

$$A = \frac{QK^\top}{\sqrt{d}}$$
$$A' = A \odot (1 + \gamma M)$$

5 Experiments

5.1 Experimental Setup

UD Parsing The newly-created dataset above, where seven typologically diverse languages that vary in word order, argument realization, and functional morphology are used in this experiment. Their UD representations are automatically obtained using a standard multilingual UD parser, the Stanza parser (Qi et al., 2020).

Model We evaluate our methods on a set of representative LLMs, including Llama 3.1 (Grattafiori et al., 2024), GPT 3.5-Turbo, and GPT 5-Mini, which are used under the UD-Prompt setting (see Appendix A.2). For UD-Tuning and UD-Att, we adopt Llama 3.1 as the primary LLM backbone, while XLM-RoBERTa-Large (Conneau et al., 2020) serves as a pretrained language model (PLM) baseline for evaluating the effect of UD-Attention in a discriminative manner.

For each model, we construct both the text-only variant and its UD-aware counterpart using the methods described in §4. To ensure fair comparison, all fine-tuned models are trained with the same datasets, batch sizes, optimization schedules, and sampling strategies. Additional training details and hyper-parameters are provided in Appendix B.1.

5.2 Experimental Results

UD-Prompt Results Table 1 details the performance of the UD-Prompt strategy across all language pairs. The results reveal that the efficacy of prompting with explicit syntax is contingent on both the modality of the representation and the capacity of the underlying model.

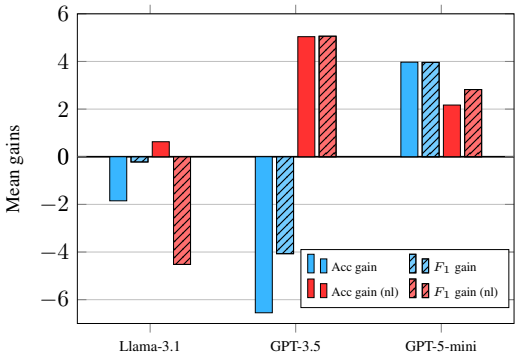


Figure 3: Impact of different UD-Prompt strategies on accuracy and F_1 performance across three LLMs, with both metrics averaged across languages.

As shown by Figure 3, the model performance is highly sensitive to the textual form of UD information. Injecting raw symbolic UD often proves detrimental, precipitating accuracy drops of over 6% for models such as GPT 3.5-turbo. In contrast, converting these dependencies into natural language consistently improves performance across architectures. This suggests that, although LLMs struggle with symbolic representations, they can effectively exploit verbalized syntactic cues aligned with their linguistic pre-training, which is in line with prior studies showing that formal graph structures tend to offer less benefit than textual descriptions (Jin et al., 2024; Zhang et al., 2025).

However, a critical exception, and a reversal of this trend, is observed with GPT 5-mini. Unlike its predecessors, this model achieves superior performance with the raw symbolic variant (∼4.0% gain), even outperforming its own natural language counterpart (∼2.2% gain). One possible explanation is

		EN-FR	EN-ES	EN-DE	EN-ZH	EN-JA	EN-KO
Llama3.1-Ins (8B)	Text-only	49.73/65.68	**51.35/66.67**	54.59/70.21	50.27/65.67	52.97/65.88	48.11/63.36
	UD-Prompt	48.65/65.45–	49.19/65.94↓	54.05/70.18–	49.19/65.94–	49.19/65.94↓	45.68/62.71↓
	UD-Prompt (nl)	**50.27/64.62**–	50.27/63.78–	**56.22/69.43**↑	**50.81/58.45**–	**54.59/56.70**↑	**48.65/57.40**–
GPT 3.5-turbo	Text-only	78.38/80.10	80.10/81.22	82.66/84.00	**75.14/78.10**	72.43/75.36	70.27/73.68
	UD-Prompt	71.35/75.12↓	74.59/76.50↓	78.32/80.58↓	64.05/71.52↓	67.03/73.01↓	64.32/71.30↓
	UD-Prompt (nl)	**85.95/87.00**↑	**86.49/87.31**↑	**85.41/87.08**↑	74.05/78.38–	**80.54/83.02**↑	**76.76/80.00**↑
GPT 5-mini	Text-only	83.51/84.56	84.32/84.82	82.66/83.92	79.46/81.46	79.46/80.90	77.57/78.99
	UD-Prompt	**87.03/87.37**↑	**88.11/88.24**↑	**87.57/89.10**↑	82.16/84.36↑	**82.70/84.47**↑	**83.24/84.88**↑
	UD-Prompt (nl)	84.86/86.41–	85.95/87.13↑	86.49/88.37↑	**82.16/84.65**↑	80.00/82.13–	80.54/82.86↑

Table 1: Accuracy/F_1 performance on the CAPI task under three UD-prompt configurations: text-only baseline, UD-Prompt with linearized UD relations, and UD-Prompt (nl) expressing UD information in natural-language form. Symbols denote statistical significance relative to the text-only one: ↑ improvement, ↓ degradation, − not significant.

that, while weaker models rely more heavily on language-mediated structural descriptions, highly capable models exhibit a greater ability to process symbolic structures directly.

We also conduct an additional experiment to further disentangle whether the gains of GPT 5-mini from natural language prompts arise from the UD structure or simply from increased textual descriptiveness. In this setting, the same verbalized sentences are retained, but all syntactic relations are randomly shuffled, thereby preserving lexical content while disrupting structural information. As Table 2 shown, while performance generally remains above the text-only baseline, it is typically inferior to the original UD-based natural language prompts. We contend that this finding suggests that the performance gains cannot be attributed solely to more verbose or descriptive input; rather, the syntactic organization encoded in the prompts plays a substantive role.

	Acc/F_1	vs. Text-only	vs. UD-Prompt (nl)
EN-FR	85.95/87.38	+2.44/+2.82	+1.09/+0.97
EN-ES	85.75/86.95	+1.43/+2.13	−0.20/−0.18
EN-DE	85.41/87.44	+2.75/+3.52	−1.08/−0.93
EN-ZH	81.08/83.72	+1.62/+2.26	−1.08/−0.93
EN-JA	78.38/81.13	−1.08/+0.23	−1.62/−1.00
EN-KO	79.46/82.08	+1.89/+3.09	−1.08/−0.78

Table 2: UD-Prompt results with shuffled syntactic relations using GPT-5-mini.

UD-Tuning Results The results of UD-Tuning, summarized in Table 3, show that its effectiveness is highly dependent on the underlying training paradigm. In the classification-based setting, incorporating UD information through supervised fine-tuning consistently degrades performance, suggesting that for models optimized for discriminative objectives, the forced injection of explicit syntactic representations behaves as noise, disrupting rather than reinforcing established decision boundaries. By contrast, the instruction-based setting displays the opposite trend: UD-Tuning yields a substantial average accuracy gain of 8.24%. We attribute this improvement to the generative nature of instruction tuning, which enables the model to interpret UD information as supportive contextual signals rather than competing features, thereby facilitating more effective integration.

UD-Attention Results Table 4 reports the performance of the UD-Attention strategy on the CAPI task. We observe consistent gains across both architectures: the PLM achieves an average improvement of 1.98% in accuracy and 4.41% in F_1, while the LLM obtains average gains of 2.53% in accuracy and 1.80% in F_1. These results demonstrate that explicitly biasing self-attention with dependency-based structural constraints in a classification setting is an effective means of enhancing cross-lingual transfer performance.

Beyond the overall performance gains, the two architectures display markedly different robustness patterns. Unlike LLMs — whose benefits are greatest for typologically distant languages — the PLM shows a clear close-language bias. Its largest improvements occur for closely related European pairs, yielding an average accuracy gain of 3.34% (up to 4.59% for EN–ES), while gains for distant languages are minimal, averaging only 0.63%. This contrast suggests that LLMs possess a stronger latent capacity for cross-lingual generalization, enabling them to exploit syntactic priors most effectively where linguistic alignment is weak.

		EN-FR	EN-ES	EN-DE	EN-ZH	EN-JA	EN-KO
CLASS-based	Text-only	**84.05/82.28**	**80.81/79.30**	**81.35/77.96**	**72.16/72.24**	**73.51/73.66**	**71.89/71.57**
	UD-Tuning	78.38/74.36↓	75.41/74.65↓	74.32/73.09↓	66.76/64.23↓	66.49/64.57↓	67.30/60.33↓
INSTR-based	Text-only	87.30/86.84	**88.11/87.71**	85.14/84.59	44.05/60.57	48.38/60.46	43.24/60.07
	UD-Tuning	**89.19/88.57**↑	87.84/87.39–	**85.68/84.99**–	**69.73/70.53**↑	**58.65/62.22**↑	**54.59/52.27**↑

Table 3: Accuracy/F_1 performance on the CAPI task of UD-Tuning. Results are reported for two LLM settings, classification-based and instruction-based, each compared against its text-only baseline. Symbols denote the statistical significance: ↑ improvement, ↓ degradation, − not significant.

		EN-FR	EN-ES	EN-DE	EN-ZH	EN-JA	EN-KO
PLM (Li, 2025)	Text-only	91.08/90.21	88.92/87.91	89.18/87.65	66.22/59.81	64.32/52.52	68.92/62.78
	UD-Attention	**95.41/94.64**↑	**93.51/92.64**↑	**90.27/89.02**↑	**66.76/61.20**–	**64.85/61.64**–	**69.73/68.18**–
LLM	Text-only	84.05/82.28	80.81/79.30	**81.35/77.96**	72.16/72.24	73.51/73.66	71.89/71.57
	UD-Attention	**84.32/83.33**–	**83.51/82.01**↑	80.54/77.54–	**76.76/75.14**↑	**76.23/74.71**↑	**77.57/75.08**↑

Table 4: Accuracy/F_1 performance of the UD-Attention on the CAPI task for both PLM and LLM architectures. Symbols denote statistical significance relative to the text-only one: ↑ improvement, ↓ degradation, − not significant.

6 Discussion

While UD integration is broadly beneficial, its impact is not uniform. We conduct a finer-grained analysis of when UD structures meaningfully support CAPI in LLMs and how these gains arise.

6.1 Cross-Lingual and Typological Analysis

To examine whether the effect of UD integration varies across language families, we group them into Indo-European (FR, ES, DE) and Asian (ZH, JA, KO) subsets and compute their respective performance gains. The independent two-sample t-test is performed to assess cross-family differences.

		EUR	ASI	p Value
UD-Promt	Llama 3	0.36	0.90	0.5821
	GPT 3.5	5.57	4.50	0.7599
	GPT 5	4.07	3.87	0.8537
UD-Tuning	CLASS	-6.03	-5.67	0.7014
	INSTR	0.72	15.77	0.0916
UD-Attention	LLM	0.72	4.33	0.0574

Table 5: Mean gains for Indo-European (EUR) vs. Asian (ASI) languages, with p-value comparisons

The results (Table 5) reveal a clear family-specific effect for UD-Tuning and UD-Attention, which deliver significantly larger gains for Asian than for Indo-European languages (p=0.0916 and p=0.0574), while UD-Prompt shows no cross-group difference. This pattern indicates that UD acts primarily as a compensatory structural prior: it is most beneficial where cross-lingual syntactic alignment is weak, and explicit structural supervision can stabilize representations. The effect is also stronger when UD is integrated as a targeted structural constraint rather than as optional guidance, which helps explain the more uniform gains observed in UD-Prompt.

Following Li (2025), we also assess whether UD-based similarity can predict LLM performance. We approximate cross-lingual structural similarity by converting the UD-derived matrices into scalar scores and averaging them for each language, then correlate these scores with the peak accuracy achieved under each integration strategy. For each paradigm, we compute Pearson correlation coefficients and fit a linear regression model. The results reveal a consistent positive relationship between UD similarity and accuracy across all settings, with strong correlations for UD-Tuning ($r = 0.916$, $p = 0.0104$), UD-Attention ($r = 0.965$, $p = 0.0018$), and UD-Prompt ($r = 0.863$, $p = 0.0267$). The corresponding regression models ($R^2 = 0.746$–0.931), shown in Figure 4, indicate that UD-based structural similarity explains a substantial share of the observed performance variance, supporting its value as a predictive metric.

6.2 Cost-Benefit Analysis across Paradigms

Beyond absolute gains, we evaluate the efficiency of UD integration by relating performance improvements to computational overhead[1]. For each paradigm, we compute baseline-relative FLOPs

[1]UD-Prompt is excluded, as it is a training-free prompting method rather than a parameter-modifying intervention.

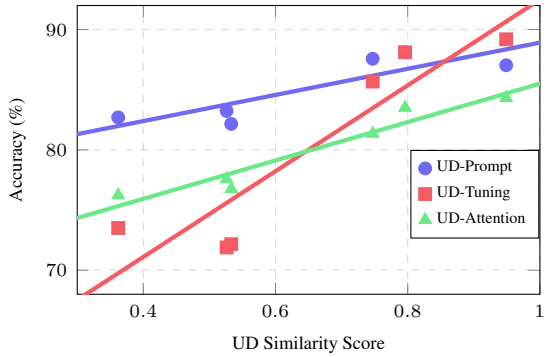
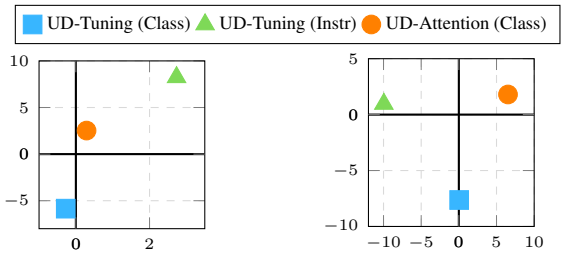


Figure 4: Linear regression analysis of UD similarity score and model accuracy across integration strategies.

and memory increases using two efficiency ratios, the data for which are provided in Appendix B.2, and visualize the cost–benefit trade-offs.

$$\eta_{\text{Acc}} = \frac{\text{Acc}^{\text{UD}} - \text{Acc}^{\text{base}}}{\text{FLOPs}^{\text{UD}} - \text{FLOPs}^{\text{base}}}$$

$$\eta_{\text{F}_1} = \frac{\text{F}_1^{\text{UD}} - \text{F}_1^{\text{base}}}{\text{Mem}^{\text{UD}} - \text{Mem}^{\text{base}}}$$



(a) Acc vs. Compute Cost (b) F_1 vs. Memory Overhead

Figure 5: Cost–Benefit efficiency curves

As shown in Figure 5, in the classification setting UD-Attention is more effective than UD-Tuning: the latter slightly reduces FLOPs but incurs large accuracy drops, whereas UD-Attention delivers consistent gains despite higher cost. In contrast, the instruction setting provides the best cost–benefit trade-off, with UD-Tuning yielding the largest average improvement (+8.24 pp) while introducing minimal or even negative memory overhead.

However, what is noteworthy is that the instruction-based regime exhibits substantially greater cross-language dispersion in both the baseline and UD-Tuning configurations (Figure 6). By contrast, performance in the classification-based setting is markedly more stable, showing low variance across languages and modest gains after UD integration, indicating stronger robustness.

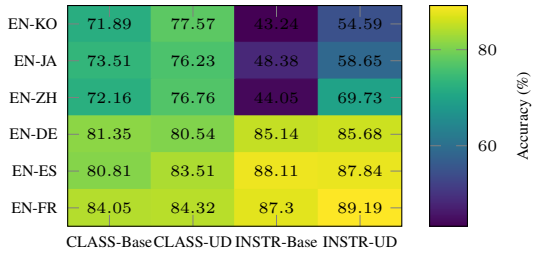


Figure 6: Robustness heatmap for baseline vs. UD-augmented models across different settings

We argue that these findings point to complementary use cases: although more memory-intensive, UD-Attention is highly effective in the classification setting, particularly for typologically distant languages, whereas instruction-based UD-Tuning delivers strong gains with minimal memory overhead and is well suited to closely related languages.

6.3 Representation Form of UD

Our results also indicate that the *form* in which UD information is represented plays a decisive role in its effectiveness. In the UD-Prompt, natural-language verbalizations generally outperform raw serialized structures, whereas in the classification-based regime, structural matrices in UD-Attention, rather than textual descriptions in UD-Tuning, more effectively support syntactic reasoning.

To further examine this representational effect, we conduct two ablations within UD-Attention. First, we ablate the hypergraph and iteratively compute arc-wise similarity, $(w_i, r, w_j) \in E$ across two languages, constructing an alternative similarity matrix. Second, we augment the M with positional information via a function $p(w_n, w_m)$: when w_n and w_m lie on the same side of their heads, we set $p(w_n, w_m) = \delta$, and otherwise $p(w_n, w_m) = 1$, updating the corresponding matrix entries.

LAN	−Hypergraph	+Word order
EN-FR	83.78/81.82 [↓0.54/↓1.51]	84.32/82.74 [−0.00/↓0.59]
EN-ES	83.51/83.29 [−0.00/↑1.28]	82.16/81.36 [↓1.35/↓0.65]
EN-DE	77.57/74.30 [↓2.97/↓3.24]	79.46/76.25 [↓1.08/↓1.20]
EN-ZH	75.68/72.73 [↓1.08/↓2.41]	75.14/73.26 [↓1.62/↓1.88]
EN-JA	74.32/73.39 [↓1.91/↓1.32]	73.78/73.42 [↓2.45/↓1.29]
EN-KO	74.59/72.83 [↓2.98/↓2.25]	75.41/72.84 [↓2.16/↓2.24]

Table 6: Results for UD-Attention under two ablations: removing the hypergraph(−Hypergraph) and augmenting with positional information (+Word order).

The impacts of these modifications on LLM performance are summarized in Table 6. Specifically,

the result reveals that omitting the hypergraph structure leads to a consistent degradation in both accuracy and F_1 score across all languages. This confirms that the hypergraph-based representation is essential for LLM performance, as it captures higher-order structural dependencies that simpler arc-wise similarities fail to recover. Conversely, word-order augmentation yields marginal or even detrimental effects, which suggests that once structural relations are abstracted via the UD framework, linear sequencing becomes largely superfluous; in fact, it may introduce positional noise that undermines cross-lingual structural alignment.

7 Conclusion

Focusing on UD, this study examines whether incorporating a language-neutral syntactic framework into LLMs remains beneficial for improving their cross-lingual generalization and robustness.

Our contributions are threefold. First, we introduce CAPI, a task designed to foreground the dependence of semantic equivalence on cross-lingual syntax. To our knowledge, this is the first controlled task specifically developed for testing the effectiveness of structural priors in cross-lingual settings. Second, we formalize a suite of UD integration methods, ranging from training-free prompting to parameter-efficient fine-tuning. By systematically comparing them, we offer a clearer roadmap of how structured linguistic knowledge can be incorporated into LLMs to improve task performance. Finally, we conduct a fine-grained analysis that reveals systematic trade-offs in these approaches. We show that the efficacy of UD is not monolithic; rather, it is a context-dependent design choice contingent upon model capacity, learning regimes, representational forms, and typological variation.

Overall, our findings advocate for a nuanced perspective on utilizing linguistic knowledge. Rather than rendering formal linguistics obsolete, it is beneficial to leverage linguistic insights into LLMs in targeted and complementary ways.

Limitations

Though demonstrating promising effectiveness of UD-based structural priors for enhancing LLMs in cross-lingual settings, our evaluation remains primarily anchored on English-centric language pairs. Extending the analysis to a broader set of typologically diverse pairings would offer a more complete view of how these priors generalize across different transfer conditions. Likewise, applying the proposed approaches to additional cross-lingual reasoning tasks would further broaden the scope and applicability of our results.

Additionally, due to the multilingual scope of the dataset, it is currently infeasible to perform fine-grained error analysis tracing erroneous dependency relations and their impact on model predictions across all languages, as we lack native-level expertise in each.

References

Wasi Ahmad, Haoran Li, Kai-Wei Chang, and Yashar Mehdad. 2021. Syntax-augmented multilingual BERT for cross-lingual transfer. 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 4538–4554, Online. Association for Computational Linguistics.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–

8451, Online. Association for Computational Linguistics.

Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. 2019. Dependency-based self-attention for transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 239–246, Varna, Bulgaria. INCOMA Ltd.

Santiago Egea Gómez, Euan McGill, and Horacio Saggion. 2021. Syntax-aware transformers for neural machine translation: The case of text to sign gloss translation. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 18–27, Online (Virtual Mode). INCOMA Ltd.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Aaron Grattafiori, Abhimanyu Dubey, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohammad Mahdi Jafari, Somayyeh Behmanesh, Alireza Talebpour, and Ali Nadian Ghomsheh. 2021. Improving pre-trained language model for relation extraction using syntactic information in Persian. In *Proceedings of the Second International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021) co-located with ICNLSP 2021*, pages 38–44, Trento, Italy. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Zhijing Jin, Yuen Chen, Fernando Gonzalez Adauto, Jiarui Liu, Jiayi Zhang, Julian Michael, Bernhard Schölkopf, and Mona Diab. 2024. Analyzing the role of semantic representations in the era of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3781–3798, Mexico City, Mexico. Association for Computational Linguistics.

Wenxi Li. 2025. Evaluating the effectiveness of linguistic knowledge in pretrained language models: A case study of universal dependencies. *Preprint*, arXiv:2506.04887.

Stephan Linzbach, Dimitar Dimitrov, Laura Kallmeyer, Kilian Evang, Hajira Jabeen, and Stefan Dietze. 2024. Dissecting paraphrases: The impact of prompt syntax and supplementary information on knowledge retrieval from pretrained language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3645–3655, Mexico City, Mexico. Association for Computational Linguistics.

Tao Liu, Xin Wang, Chengguo Lv, Ranran Zhen, and Guohong Fu. 2020. Sentence matching with syntax- and semantics-aware BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3302–3312, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Ankush Raut, Xiaofeng Zhu, and Maria Leonor Pacheco. 2025. Can LLMs interpret and leverage structured linguistic representations? a case study with AMRs. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 173–185, Vienna, Austria. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Anushka Swarup, Avanti Bhandarkar, Ronald Wilson, Tianyu Pan, and Damon Woodard. 2025. From syntax to semantics: Evaluating the impact of linguistic structures on LLM-based information extraction. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 36–48, Vienna, Austria. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving relation extraction through syntax-induced pre-training with dependency masking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1875–1886, Dublin, Ireland. Association for Computational Linguistics.

Qingrong Xia, Zhenghua Li, and Min Zhang. 2019. A syntax-aware multi-task learning framework for Chinese semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5382–5392, Hong Kong, China. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Jiahuan Zhang, Tianheng Wang, Ziyi Huang, Yulong Wu, Hanqing Wu, DongbaiChen DongbaiChen, Linfeng Song, Yue Zhang, Guozheng Rao, and Kaicheng Yu. 2025. SR-LLM: Rethinking the structured representation in large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3443–3462, Vienna, Austria. Association for Computational Linguistics.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019a. Syntax-enhanced neural machine translation with syntax-aware word representations. 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 1151–1161, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Zhang, Rui Wang, and Luo Si. 2019b. Syntax-enhanced self-attention-based semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 616–626, Hong Kong, China. Association for Computational Linguistics.

A Experimental Details on the Training-free Method

This section illustrates the details in training-free integration of UD information into LLM inference, i.e., UD-Prompt.

A.1 Prompt Design

Given an input sentence and its automatically derived UD structures, the model is prompted to reason over structural information without updating model parameters. We list three prompting configurations here: (i) Text-Only, where the model receives only the raw sentence pair (see Figure 7); (ii) UD-Prompt, where UD are provided as linearized list of dependency arcs (see Figure 8); and (iii) UD-Prompt (NL), where the same structural content is rewritten into natural-language sentences (see Figure 9).

A.2 Implementation

The experiments are conducted using the different LLMs via their own standardized API interface respectively. To ensure the reliability of the semantic decision-making process, we used the following configuration: temperature is set to 1.0 to allow for sufficient reasoning variance while maintaining task focus and max token number is 8192.

B Experimental Details on the Training-dependent Method

This section details the implementation of training-dependent strategies for incorporating UD syntactic priors into LLMs, specifically through UD-

Text-only_Prompt_for_CAPI

[system_prompt]

You are given two sentences.
Your task is to determine whether the two sentences are paraphrases,
i.e., whether they convey the same meaning.
Only output 1 if they are paraphrases, or 0 otherwise.
Do not output any explanation.

[user_prompt]

sentence_1: {sentence1},
sentence_2: {sentences2},
Answer (1 = paraphrase, 0 = not paraphrase):

Figure 7: Text-only prompt example

UD_Prompt_for_CAPI

[system_prompt]

You are given two sentences and their Universal Dependency (UD) representations.
Your task is to determine whether the two sentences are paraphrases,
i.e., whether they convey the same meaning.
You may refer to the UD representations for syntactic comparison.
Only output 1 if they are paraphrases, or 0 otherwise.
Do not output any explanation.

[user_prompt]

sentence_1: {sentence1},
UD representation of sentence_1: {ud1},
sentence_2: {sentences2},
UD representation of sentence_2: {ud2},
Answer (1 = paraphrase, 0 = not paraphrase):

Figure 8: UD-Prompt example

UD_Prompt_for_CAPI (nl)

[system_prompt]

You are given two sentences and their syntactic relations expressed using Universal Dependencies in natural language.
Your task is to determine whether the two sentences are paraphrases,
i.e., whether they convey the same meaning.
You may refer to the UD representations for syntactic comparison.
Only output 1 if they are paraphrases, or 0 otherwise.
Do not output any explanation.

[user_prompt]

sentence_1: {sentence1},
Syntactic relations of sentence_1 (Universal Dependencies): {verbalized_ud1},
sentence_2: {sentences2},
Syntactic relations of sentence_2 (Universal Dependencies): {verbalized_ud2},
Answer (1 = paraphrase, 0 = not paraphrase):

Figure 9: UD-Prompt (nl) example

Tuning and UD-Attention. Both approaches adopt parameter-efficient fine-tuning (PEFT) to align model representations with explicit structural information derived from Universal Dependencies. The two strategies differ primarily in their task-specific initialization: for classification-oriented models, we employ the task_type = TaskType.SEQ_CLS configuration, whereas for instruction-following or autoregressive generation settings, we adopt the task_type = TaskType.CAUSAL_LM configuration. This design enables a consistent integration of syntactic priors across heterogeneous model architectures and supervision regimes.

B.1 Hyperparameter Configuration

To ensure the robustness of our fine-tuned models, we conduct an automated hyperparameter search using the Weights & Biases Sweep API. Employing a Bayesian optimization strategy that iteratively samples the search space to maximize the macro-F_1 score on the development set, we execute 10 sweep trials per configuration and select the best-performing checkpoint for final evaluation.

The search space is designed to jointly optimize the learning regime and the Low-Rank Adaptation (LoRA) parameters. The hyperparameter bounds and distributions utilized across both LLM archi-

tectures are detailed below:

- **Learning Rate:** We explore a uniform distribution between 1×10^{-5} and 2×10^{-4}.
- **Input Constraints:** The maximum sequence length is set to 512 tokens for both baseline and UD-augmented models.
- **Batch Size and Memory Management:** The Instruction-based regime uses a per-device batch size of 2 with 4 gradient accumulation steps to simulate a larger global batch size while maintaining a low memory footprint. The Classification-based regime utilizes a native batch size of 8.
- **Model Selection and Reproducibility:** Models are fine-tuned for 10 epochs, with early stopping based on validation performance. For reproducibility, we use a fixed random_state of 42 for the 80/10/10 train/val/test splits across all language pairs.
- **LoRA Configuration:** We tune the rank (r) across $\{4, 8, 16, 32\}$ and the scaling factor (α) across $\{16, 32, 64\}$. To mitigate overfitting, the LoRA dropout rate is sampled from a uniform distribution $[0, 0.1]$, while the bias parameter is fixed to `None`.

B.2 Computational Resources

All fine-tuning experiments and evaluations were conducted on a cluster of 8 NVIDIA vGPUs, each equipped with 48GB of VRAM. To provide a comprehensive overview of the computational overhead associated with our proposed methods, we report the floating-point operations (FLOPs) and peak memory usage for the baseline and UD-integrated variants in Table 7.

Method	FLOPs (Base)	FLOPs (UD)	Mem (Base)	Mem (UD)
UD-Tuning (C)	1.07×10^{17}	7.95×10^{16}	28.46	28.50
UD-Tuning (I)	6.76×10^{16}	3.41×10^{17}	28.50	18.54
UD-Attention (C)	1.07×10^{17}	1.36×10^{17}	28.46	34.98

Table 7: Computational efficiency: Total FLOPs and peak GPU memory (GB). (C) denotes CLASS, (I) denotes INSTR.