Better Benchmarking LLMs for Zero-Shot Dependency Parsing

Ana Ezquerro, Carlos Gómez-Rodríguez, David Vilares
Universidade da Coruña, CITIC
Departamento de Ciencias de la Computación y Tecnologías de la Información
Campus de Elviña s/n, 15071
A Coruña, Spain
`{ana.ezquerro, carlos.gomez, david.vilares}@udc.es`

Abstract

While LLMs excel in zero-shot tasks, their performance in linguistic challenges like syntactic parsing has been less scrutinized. This paper studies state-of-the-art open-weight LLMs on the task by comparing them to baselines that do not have access to the input sentence, including baselines that have not been used in this context such as random projective trees or optimal linear arrangements. The results show that most of the tested LLMs cannot outperform the best uninformed baselines, with only the newest and largest versions of LLaMA doing so for most languages, and still achieving rather low performance. Thus, accurate zero-shot syntactic parsing is not forthcoming with open LLMs.

1 Introduction

Autoregressive large language models (LLMs) and instruction-based variants (Jiang et al., 2023; OpenAI, 2024; Dubey et al., 2024) are known for their zero-shot and few-shot abilities (Radford et al., 2019). In practical terms, they can serve as versatile systems whose behavior is easily adapted through prompting. Beyond what we experience as everyday users, documented examples in the context of natural language processing (NLP) include question answering (Baek et al., 2023; Li et al., 2024), summarization (Wang et al., 2023), machine translation (Johnson et al., 2017; Wang et al., 2021; Zhang et al., 2023) and information retrieval (Zhuang et al., 2023; Adeyemi et al., 2024; Qin et al., 2024), among many other tasks.

Related, syntactic parsing has long explored few-shot learning approaches. Prior to the development of current LLMs, various methods were studied to perform zero-shot or few-shot parsing, and many of these approaches achieved competitive results. These methods focused on factors such as the quality and quantity of annotations (Meechan-Maddon and Nivre, 2019), cross-lingual learning (Xu and Koehn, 2021), multilingual pre-training (Tran and Bisazza, 2019), and treebank difficulty (Søgaard, 2020; Anderson et al., 2021). However, the effectiveness of zero-shot parsing with LLMs remains a topic of debate. Recent work has showed how state-of-the-art LLMs exhibit low performance in syntactic parsing (Bai et al., 2023; Lin et al., 2023), even when designing manual specific prompts (Li et al., 2023; Blevins et al., 2023). Nonetheless, these results have been deemed sufficient to categorize LLMs as potential zero-shot parsers. While some studies (Tian et al., 2024) suggest that multi-stage complex approaches can yield a competitive zero-shot performance, in this work we will focus on single prompt approaches – similar to what works for other NLP tasks – to evaluate to what extent LLMs can perform the task on their own without externally-provided planning. Studies covering these approaches leave a substantial gap in evaluating LLMs on low-resource setups and often omit comparison with uninformed baselines, which are essential for determining whether LLMs achieve accuracy levels meaningfully above chance.

Contribution We address the lack of comparison against uninformed baselines, and include some that have not been proposed before but offer a higher standard than traditional blind baselines, such as left- or right-branching trees. These baselines provide more robust benchmarks and offer a fairer evaluation of LLMs' potential as zero-shot parsers. We prioritize depth over breadth by evaluating a wide range of LLMs to identify any substantial differences across them - a contribution that, to our knowledge, has not been thoroughly explored in previous work.[1]

---

[1] Code available at `github.com/anaezquerro/naipar`

Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 121–135
March 3-4, 2025 ©2025 University of Tartu Library

2 Zero-shot dependency parsing

Next, we review the notation, benchmarks, uninformed baselines, and introduce the LLMs used. Dependency parsing is the task of obtaining the syntactic structure of a sentence as a set of labeled directed relations (*dependencies*) between words. In zero-shot parsing (whether relying on LLMs or other models), the core idea is to perform the dependency parsing task without using task-specific labeled data during either the pre-training or fine-tuning steps. This approach contrasts with the standard setup for training dependency parsers, where task-specific labeled data is integral to explicitly teaching models syntactic structures in a supervised learning framework. Instead, zero-shot parsing leverages the model's general pre-trained knowledge to infer syntactic relationships in unseen data. Before the emergence of large autoregressive generative models, pre-training largely avoided task-specific annotations, aligning closely with the zero-shot paradigm. However, given the extensive and diverse nature of the data these models are trained on, it is plausible that some exposure to annotated dependency parsing examples has occurred. This possibility will be examined further in subsequent sections.

Notation Let W=$(w_1, ..., w_n)$ be a sentence, a dependency graph is defined as $G = (W, A)$, where W is the set of nodes and A the set of arcs. Each arc in A is a tuple (h, d), where $h \in [0, n]$ is the position of the head node, and $d \neq h \in [1, n]$ the position of the dependent node.[2] G is a tree T iff (i) is a *connected acyclic* graph, (ii) each word w_i has only one head, so $A = \{(h, d) : d = 1...n\}$, and (iii) there is only one arc of the form $(0, d)$, and w_d is designated as the *root* of the sentence. This work only studies trees.

2.1 Zero-shot (uninformed) baselines

Previous work has reported results on parsing using LLMs (Lin et al., 2023), classifying LLMs as *potentially* zero-shot parsers. We revisit this claim by proposing a comprehensive benchmark and comparing against uninformed baselines (i.e., baselines that generate an output tree without looking at the contents of the input, although sometimes with access to its length). Uninformed baselines are useful to determine whether the models are meaningfully processing the input or just generating outputs that could be obtained by chance or by using properties that are not specific to the input sentence (e.g. the common trend towards projectivity in human syntax). We use both conventional uninformed baselines that have been used previously in related contexts (e.g. Klein and Manning, 2004) and more sophisticated, though still uninformed, baselines that, to our knowledge, have not yet been applied for this purpose.

2.1.1 Conventional baselines

We now describe baselines that have been used as naive approaches to build simplistic yet valid trees.

Randomized root-based tree generation Our most basic baseline randomly selects a root node, denoted as d', and creates a dependency from d' to the rest of nodes. Formally, $\hat{A} = \{(0, d'), (d', d) : d = (1, ..., n) \neq d'\}$.

Right- and left-branching tree generation This method assigns each word as a dependent of the previous (next) word, with the first (last) word as the root. Right-branching trees are a classic baseline for unsupervised English dependency parsing (Klein and Manning, 2004), as English syntax is predominantly right-branching. The left-branching baseline is included as some languages are predominantly left-branching.

Uniformly random tree generation Another parsing baseline (Klein and Manning, 2004). We use the Aldous (1990) algorithm to guarantee generation of a uniformly random dependency tree.

Sampling from a reference treebank We build the tree distributions of different lengths from a reference treebank. For a sentence, we sampled a dependency tree from the distribution of length n. Note that this is the only one among our baselines that has access to a treebank, although it *is still uninformed* with respect to the input sentence.

2.1.2 Novel uninformed baselines

We refine random tree generation by taking into account observed properties of human language: the scarcity of crossing dependencies (Ferrer-i-Cancho et al., 2018) and dependency distance minimization, i.e., the tendency of syntactic structures to minimize the distance between syntactically related words (i.e. the length of dependencies) in order to reduce cognitive processing effort (Liu et al., 2017; Ferrer-i Cancho et al., 2022).

---

[2] Graphs have arcs labeled with syntactic functions, but we ignore them here as our evaluation is unlabeled.

Uniformly random projective tree generation The goal is to generate a projective tree (where dependencies do not cross) uniformly at random. As rejection sampling is too slow, we use Nijenhuis and Wilf (1978)'s algorithm to generate a random unlabeled rooted tree and then assign a random projective arrangement following (Futrell et al., 2015; Alemany-Puig and Ferrer-i Cancho, 2024).[3]

Uniformly random (projective) optimal-distance tree generation Again, we start from a uniformly random unlabeled rooted tree. In this case, we give it the linear arrangement that minimizes the sum of dependency distances, using Shiloach (1979)'s algorithm, as well as the minimum-distance *projective* arrangement, with the algorithm by Alemany-Puig et al. (2022).[3]

2.2 Zero-shot parsing with LLMs

Prompting setup Adopting a strategy similar to Lin et al. (2023), we query LLMs using simple prompts. The prompt includes an introductory sentence requesting output in CoNLL format, followed by a basic example from a reference treebank, where only the `ID`, `HEAD`, and `DEPREL` fields are populated. We selected a random sentence of length 4 to 7 to avoid longer sequences, maintaining a zero-shot setup. Although this may resemble a one-shot setup, the example is intentionally simple, serving only to reduce formatting errors rather than offering linguistic content. Figure 1 breaks down the specific prompt we used.

Postprocessing We account for possible corrupted outputs, such as column mismatches, missing nodes, or multiple roots. From the model's raw output, we applied two post-processing steps: first, filtering tabular lines and filling fields to match the CoNLL format with correct row and column counts for sentences of length n. Second, we resolved cycles, enforced a unique-root constraint, and replaced out-of-range arcs with root connections to ensure a single-rooted, connected tree. Figure 1 also shows an example of the input and output after the first post-processing step.

3 Experiments

We conduct an in-depth evaluation of LLMs as zero-shot dependency parsers by generating outputs in CoNLL format and comparing them to uninformed baselines. Unlike Lin et al. (2023), who evaluated only ChatGPT-3.5 due to limited system availability, our work expands the analysis to a broader set of models across a select few languages, albeit on a smaller subset of treebanks. This approach, while time-intensive due to the extensive input and output token requirements, offers a more comprehensive understanding of model performance across different LLMs.

Prompt example

```
In dependency parsing the CoNLL format for the sentence <The
trial begins again Nov 28 .> is:
1 The _ _ _ _ 2 det _ _
2 trial _ _ _ _ 3 nsubj _ _
3 begins _ _ _ _ 0 root _ _
4 again _ _ _ _ 3 advmod _ _
5 Nov. _ _ _ _ 3 obl:tmod _ _
6 28 _ _ _ _ 5 nummod _ _
7 . _ _ _ _ 3 punct _ _
Now return the CoNLL format for the sentence: <What if Google
Morphed Into GoogleOS ?>
```

(1) Well-formatted output

```
1 What _ _ _ _ 0 nsubj _ _
2 if _ _ _ _ 4 mark _ _ _
3 Google _ _ _ _ 4 nsubj _ _
4 Morphed _ _ _ _ 0 root _ _
5 into _ _ _ _ 6 case _ _
6 GoogleOS _ _ _ _ 8 nmod _ _
7 ? _ _ _ _ 4 punct _ _
```

Figure 1: Prompt and output after the first post-processing. See Figure 3 for step-by-step process.

Datasets We selected 4 treebanks from UD 2.14 (Zeman et al., 2024) to conduct experiments in different languages, specifically in English$_{\text{EWT}}$, French$_{\text{GSD}}$, German$_{\text{GSD}}$, and Hindi$_{\text{HDTB}}$.

Evaluation We use the unlabeled attachment score (UAS) and unlabeled exact match (UM) as our primary metrics. For the zero-shot dependency parsers, we report performance after the first post-processing step (ensuring that the CoNLL format file contains all columns) and the second (confirming that the tree is well-formed).

Models We selected several instruction-based models from the Gemma (Gemma Team et al., 2024a,b), LLaMA (Touvron et al., 2023; Dubey et al., 2024), and Mistral (Jiang et al., 2023, 2024) series. Appendix B (Table 3) breaks down the links to all models. All reported results were obtained limiting the inference to half precision.[3]

4 Analysis of results

Table 1 compares the performance of the tested models with uninformed baselines. We see that only the latest and largest versions of LLaMa (i.e.,

---

[3] We used the implementation of these algorithms in the LAL library (Alemany-Puig et al., 2021).

[3] Preliminary experiments indicated that reduced inference precision had minimal impact on performance.

		English$_{\text{EWT}}$			French$_{\text{GSD}}$			German$_{\text{GSD}}$			Hindi$_{\text{HDTB}}$		
		UAS	**UM**	**%w**	**UAS**	**UM**	**%w**	**UAS**	**UM**	**%w**	**UAS**	**UM**	**%w**
	A	20.74	13.91	100.00	5.99	0.24	100.00	9.79	2.05	100.00	5.92	0.06	100.00
	R	23.30	12.13	100.00	10.67	0.00	100.00	11.59	1.23	100.00	**25.34**	0.00	100.00
	L	34.41	9.39	100.00	29.78	0.00	100.00	29.59	1.43	100.00	21.34	0.00	100.00
	RD	20.10	10.78	100.00	5.93	0.00	100.00	9.64	1.13	100.00	5.67	0.00	100.00
	RD*	21.45	12.42	100.00	6.06	0.00	100.00	9.38	1.74	100.00	5.59	0.00	100.00
	LI	28.09	11.75	100.00	16.81	0.00	100.00	19.24	1.64	100.00	19.99	0.06	100.00
	LI*	26.99	10.98	100.00	17.08	0.00	100.00	20.53	1.84	100.00	20.06	0.00	100.00
	S	31.14	15.55	100.00	19.43	**0.96**	100.00	20.02	2.46	100.00	17.92	**0.30**	100.00
	`v1-2b`	15.80 $_{5.23}$	5.54 $_{7.22}$	7.17	6.51 $_{0.52}$	0.00 $_{0.00}$	0.48	9.66 $_{0.91}$	1.02 $_{0.92}$	2.15	11.96 $_{-1.03}$	0.00 $_{0.00}$	6.92
G	`v1-7b`	21.26 $_{5.17}$	6.93 $_{6.88}$	24.84	14.93 $_{-0.66}$	0.00 $_{0.00}$	7.69	16.61 $_{0.99}$	0.82 $_{1.13}$	10.75	9.78 $_{-0.08}$	0.00 $_{0.00}$	8.43
	`v2-9b`	20.20 $_{3.35}$	6.79 $_{6.50}$	15.17	13.32 $_{-1.34}$	0.00 $_{0.72}$	4.81	14.75 $_{-0.68}$	0.92 $_{1.13}$	5.32	12.18 $_{-0.77}$	0.00 $_{0.06}$	3.27
	`v2-7b`	12.98 $_{10.38}$	3.18 $_{11.12}$	24.22	18.20 $_{-1.45}$	0.00 $_{0.24}$	1.92	18.70 $_{0.19}$	0.10 $_{2.05}$	4.09	10.64 $_{-1.74}$	0.00 $_{0.00}$	5.82
	`v2-13b`	18.95 $_{4.01}$	5.83 $_{8.96}$	14.59	13.64 $_{-0.81}$	0.00 $_{0.00}$	22.12	19.40 $_{-1.78}$	0.00 $_{0.00}$	32.89	14.77 $_{-1.77}$	0.00 $_{0.06}$	18.71
	`v2-70b`	13.78 $_{11.38}$	4.00 $_{10.98}$	46.89	19.05 $_{-1.42}$	0.24 $_{0.00}$	13.70	25.88 $_{-2.41}$	1.43 $_{0.20}$	23.34	15.27 $_{-2.33}$	0.00 $_{0.00}$	7.47
	`v3-8b`	18.34 $_{6.56}$	7.41 $_{8.09}$	49.01	14.80 $_{-2.84}$	0.00 $_{0.00}$	4.81	29.51 $_{-2.26}$	1.74 $_{0.10}$	22.42	17.03 $_{-1.54}$	0.00 $_{0.06}$	0.48
∞	`v3-70b`	38.30 $_{0.98}$	**16.37** $_{1.44}$	58.69	29.20 $_{-0.61}$	0.96 $_{0.48}$	33.41	33.91 $_{-1.16}$	3.17 $_{0.41}$	28.25	14.24 $_{-1.54}$	0.00 $_{0.00}$	26.14
	`v3.1-8b`	28.83 $_{-1.31}$	11.75 $_{2.07}$	34.38	24.61 $_{-6.35}$	0.72 $_{0.00}$	5.29	26.93 $_{-4.67}$	1.64 $_{0.20}$	12.90	18.86 $_{-2.11}$	0.00 $_{0.12}$	2.43
	`v3.1-70b`	**39.69** $_{1.46}$	15.65 $_{2.41}$	**65.86**	**34.62** $_{-1.30}$	0.96 $_{0.24}$	**42.79**	**36.75** $_{-1.14}$	**3.48** $_{0.20}$	**46.57**	14.37 $_{-0.69}$	0.00 $_{0.00}$	26.14
	`v3.2-1b`	15.07 $_{7.35}$	4.53 $_{8.33}$	16.75	8.05 $_{-0.65}$	0.00 $_{0.24}$	11.06	12.61 $_{-0.9}$	0.41 $_{1.64}$	10.03	7.20 $_{-1.9}$	0.00 $_{0.0}$	8.37
	`v3.2-3b`	18.51 $_{3.98}$	6.64 $_{6.55}$	18.68	10.22 $_{-1.1}$	0.24 $_{0.0}$	12.74	17.69 $_{0.34}$	1.13 $_{1.13}$	10.44	13.84 $_{0.03}$	0.00 $_{0.0}$	14.90
	`v1-7b`	18.59 $_{3.83}$	6.55 $_{6.55}$	16.85	10.32 $_{-0.54}$	0.00 $_{0.00}$	5.29	16.37 $_{-1.31}$	1.23 $_{0.61}$	5.94	10.63 $_{-0.44}$	0.00 $_{0.00}$	0.12
	`v2-7b`	23.02 $_{2.10}$	6.93 $_{6.07}$	15.12	18.73 $_{-3.28}$	0.24 $_{0.00}$	3.85	20.49 $_{-2.90}$	1.13 $_{0.61}$	5.02	13.80 $_{-1.11}$	0.00 $_{0.06}$	0.36
	`v3-7b`	25.04 $_{2.49}$	7.66 $_{5.39}$	28.17	27.36 $_{-5.52}$	0.24 $_{0.48}$	11.06	28.34 $_{-4.98}$	1.13 $_{0.51}$	19.55	19.41 $_{-3.15}$	0.00 $_{0.00}$	0.71
■	`x1-7b`	15.46 $_{3.21}$	2.63 $_{4.17}$	26.22	13.00 $_{-1.06}$	0.24 $_{0.00}$	3.37	16.67 $_{0.00}$	0.00 $_{0.00}$	25.00	13.68 $_{-0.67}$	0.06 $_{0.00}$	1.25
	`x1-22b`	32.91 $_{0.85}$	13.72 $_{3.32}$	57.74	23.75 $_{0.13}$	0.68 $_{0.24}$	36.73	22.48 $_{-0.19}$	2.76 $_{0.92}$	38.44	19.37 $_{-0.67}$	0.09 $_{0.26}$	**37.84**
	`nemo`	20.96 $_{3.89}$	7.56 $_{7.03}$	15.74	15.85 $_{-1.49}$	0.00 $_{0.00}$	3.61	14.10 $_{-0.57}$	1.23 $_{0.72}$	4.09	9.59 $_{-0.06}$	0.00 $_{0.00}$	0.48
	`large`	28.71 $_{0.81}$	10.01 $_{4.77}$	18.25	15.21 $_{-0.58}$	0.83 $_{0.42}$	5.00	17.66 $_{0.58}$	1.74 $_{0.92}$	7.88	14.46 $_{-1.4}$	0.00 $_{0.00}$	26.14

Table 1: Performance on the test sets. The baselines are: all-to-root (A), left (L) and right (R) branching, random generation (RD), optimal linear arrangement (LI) and sampling (S). The symbol (*) indicates if projectivity is fixed as a constraint. **%w** is the ratio of outputs that did not require post-processing. We also report results with Gemma (G), LLaMA (∞), and Mistral models(■), with versions (`v`, `x`) and parameter counts. Subscripts indicate performance boost from the second post-processing step.

the 70B versions of Llama 3 and 3.1) consistently outperform all the baselines in most languages in terms of UAS and UM, and only do so barely (e.g., with the best result on English being about 5.5 points above the left-branching baseline without postprocessing, and close to 7 points with post-processing). The rest of the models clearly fall behind, showing that they are not doing any meaningful parsing at all. In the case of Hindi, no model at all reaches the best baselines. Among our baselines, traditional left (or right in the case of Hindi) branching baselines are the most competitive in terms of UAS,[4] although baselines based on optimal linear arrangement come close, and the sampling baseline is better in terms of English UM. In Appendix B (Tables 4 to 7) we also include tables showing the individual scores of each model based on the PoS tag of the head in each treebank.

Figure 2 complements Table 1 by illustrating the performance of a representative subset of models in terms of dependency displacements (i.e., performance taking into account the difference between the position of the dependent and its head) for the English$_{\text{EWT}}$. treebank. We observe that LLaMa v3.1 70B consistently performs better than the sampling and optimal linear arrangement baselines, not only on short dependencies but also on longer rightward dependencies. However, for the rest of the models, the differences with respect to uninformed baselines become subtler. Similar figures for the other evaluated treebanks can be found in the Appendix (Figures 4, 5, and 6).

Overall, the results show that open-weight LLMs are far from being potential zero-shot dependency parsers, contrary to claims about ChatGPT (Lin et al., 2023). Considerable scaling or other improvements would be required for this situation to change.

5 Limitations

Memorization Memorization refers to the LLM's ability to recall specific patterns, struc-

---

[4] Superiority of the left-branching baseline on English can be surprising, as right-branching has often been deemed better on English unsupervised parsing (Klein and Manning, 2004; Li et al., 2020); but these papers do not use UD.

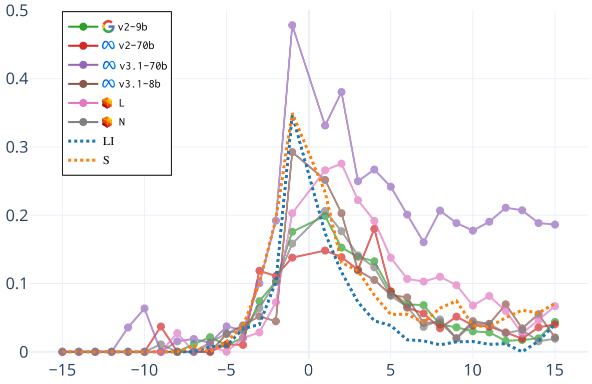

Figure 2: F-score across displacements in the English$_{\text{EWT}}$ test set.

		UAS	UM	%w
	A	$6.46_{-14.27}$	$1.69_{-12.22}$	100
	R	$24.31_{1.01}$	$0.70_{-11.43}$	100
	L	$20.15_{-14.25}$	$2.32_{-7.07}$	100
	RD	$5.51_{-14.59}$	$0.91_{-11.51}$	100
	RD*	$7.32_{-14.13}$	$1.51_{-10.91}$	100
	LI	$16.39_{-11.70}$	$1.77_{-9.98}$	100
	LI*	$19.54_{-7.45}$	$1.40_{-9.58}$	100
	S	$22.24_{-8.89}$	$2.55_{-13.00}$	100
∞	`v3.1-70b`	$28.83_{-12.32}$	$0.00_{-18.06}$	$38.72_{-27.14}$
	`v3.2-3b`	$8.14_{-14.35}$	$0.00_{-13.19}$	$11.04_{-7.64}$
◆	`x1-22b`	$20.37_{-13.39}$	$1.00_{-16.04}$	$58.92_{1.18}$
	`large`	$25.37_{-4.15}$	$3.00_{-11.78}$	$13.24_{-5.01}$

Table 2: Performance on silver annotations. Subscripts denote the performance drop from Table 1.

tures, or dependencies encountered during pre-training, rather than generalizing to unseen cases (Hartmann et al., 2023). This poses a risk of generalization issues or the regurgitation of chunks of text, which could affect our evaluation but is difficult to quantify (Sainz et al., 2023). Although this is beyond the main scope of our work, we have attempted to briefly analyze this phenomenon. To do so, we crawled a few hundred recent news articles from the New York Times Archive API.[5] The aim was to collect new text, guaranteeing that no annotations for it were available online when the models were trained. We then produced silver annotations by using a trained graph-based model (Dozat et al., 2017) – a state-of-the-art dependency parser – to parse these articles. In Table 2, we present the results of a few representative models against these silver annotations. The results are consistent with those for UD in Table 1: while UAS scores are lower across the board, this happens both for LLMs and baselines, and likely stems from NYT sentences being longer on average. In relative terms, the same trends as in UD stand, with only LLaMa 3.1-70B clearly outperforming all baselines, so we do not detect evidence of our main results being overestimated due to data contamination.

Prompting The prompting approach used in this study followed a straightforward design. We acknowledge that there may be room for improving parsing performance through more advanced prompt engineering techniques. Our goal was methodological, establishing a set of uninformed baselines rather than optimizing prompt configurations. In this context, approaches such as in-context learning (Brown et al., 2020; Chen et al., 2021), chain-of-thought prompting (Wei et al., 2022), and self-consistency (Wang et al., 2022) have shown promise in improving performance by fostering more structured reasoning.

Language selection Our selection of languages was limited to a small set, three of which belong to the Indo-European family. This choice was driven by two key factors. First, although we had the exclusive access to a few 24GB RTX 3090, these were insufficient for running larger models effectively. We also had access to CESGA, the supercomputing center of Galicia; but it was limited to queuing systems, making it difficult to estimate running times and prioritize experiments given the large number of models involved. Additionally, although many models claim to be multilingual, their performance tends to be skewed toward a subset of widely spoken languages. We therefore selected languages that have the most support across models to ensure consistent evaluations.[6]

6 Conclusion

We revisited the potential of autoregressive LLMs as zero-shot dependency parsers. Taking a more conservative approach than previous studies, we compared several LLMs with simple baselines to establish minimal performance benchmarks. Our results show that most LLMs performed on par with uninformed baselines, indicating comparable performance to toy approaches that operate without any access to the input sentence.

---

[5] `developer.nytimes.com/docs/archive-product/1`

[6] Note that not all languages are supported by all models (Table 9). Our selection aims to include widely supported languages to ensure fair comparisons across models.

Acknowledgments

We acknowledge grants SCANNER-UDC (PID2020-113230RB-C21) funded by MICIU/AEI/10.13039/501100011033; GAP (PID2022-139308OA-I00) funded by MICIU/AEI/10.13039/501100011033/ and ERDF, EU; LATCHING (PID2023-147129OB-C21) funded by MICIU/AEI/10.13039/501100011033 and ERDF, EU; and TSI-100925-2023-1 funded by Ministry for Digital Transformation and Civil Service and "NextGenerationEU" PRTR; as well as funding by Xunta de Galicia (ED431C 2024/02), and Centro de Investigación de Galicia "CITIC", funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS). We also extend our gratitude to CESGA, the supercomputing center of Galicia, for granting us access to its resources.

References

Mofetoluwa Adeyemi, Akintunde Oladipo, Ronak Pradeep, and Jimmy Lin. 2024. Zero-Shot Cross-Lingual Reranking with Large Language Models for Low-Resource Languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–656, Bangkok, Thailand. Association for Computational Linguistics.

David J. Aldous. 1990. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM J. Discret. Math.*, 3(4):450–465.

Lluís Alemany-Puig, Juan Luis Esteban, and Ramon Ferrer-i Cancho. 2021. The linear arrangement library. A new tool for research on syntactic dependency structures. In *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, pages 1–16, Sofia, Bulgaria. Association for Computational Linguistics.

Lluís Alemany-Puig, Juan Luis Esteban, and Ramon Ferrer i Cancho. 2022. Minimum projective linearizations of trees in linear time. *Information Processing Letters*, 174:106204.

Lluís Alemany-Puig and Ramon Ferrer-i Cancho. 2024. The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, 12(1):1–42.

Mark Anderson and Carlos Gómez-Rodríguez. 2022. The impact of edge displacement Vaserstein distance on UD parsing performance. *Computational Linguistics*, 48(3):517–554.

Mark Anderson, Anders Søgaard, and Carlos Gómez-Rodríguez. 2021. Replicating and extending "Because their treebanks leak": Graph isomorphism, covariants, and parser performance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1090–1098, Online. Association for Computational Linguistics.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pages 70–98, Toronto, ON, Canada. Association for Computational Linguistics.

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency parsing using LLMs. *Preprint*, arXiv:2310.19462.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting Language Models for Linguistic Structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Ramon Ferrer-i-Cancho, Carlos Gómez-Rodríguez, and Juan Luis Esteban. 2018. Are crossing dependencies really scarce? *Physica A: Statistical Mechanics and its Applications*, 493:311–329.

Ramon Ferrer-i Cancho, Carlos Gómez-Rodríguez, Juan Luis Esteban, and Lluís Alemany-Puig. 2022. Optimality of syntactic dependency distances. *Physical Review E*, 105(1).

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024a. Gemma: Open Models Based on Gemini Research and Technology. *Preprint*, arXiv:2403.08295.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024b. Gemma 2: Improving Open Language Models at a Practical Size. *Preprint*, arXiv:2408.00118.

Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and Robert West. 2023. Sok: Memorization in general-purpose large language models. *arXiv preprint arXiv:2310.18362*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *Preprint*, arXiv:2401.04088.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 478–485, Barcelona, Spain.

Jianling Li, Meishan Zhang, Peiming Guo, Min Zhang, and Yue Zhang. 2023. LLM-enhanced Self-training for Cross-domain Constituency Parsing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8174–8185, Singapore. Association for Computational Linguistics.

Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. 2020. An empirical comparison of unsupervised constituency parsing methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3278–3283, Online. Association for Computational Linguistics.

Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024. Self-Prompting Large Language Models for Zero-Shot Open-Domain QA. 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 296–310, Mexico City, Mexico. Association for Computational Linguistics.

Boda Lin, Xinyi Zhou, Binghao Tang, Xiaocheng Gong, and Si Li. 2023. ChatGPT is a Potential Zero-Shot Dependency Parser. *Preprint*, arXiv:2310.16654.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*, 21:171–193.

Ailsa Meechan-Maddon and Joakim Nivre. 2019. How to parse low-resource languages: Cross-lingual parsing, target language annotation, or both? In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 112–120, Paris, France. Association for Computational Linguistics.

Albert Nijenhuis and Herbert Wilf. 1978. *Combinatorial Algorithms for Computers and Calculators*, second edition. Academic Press.

OpenAI. 2024. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael

Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Y. Shiloach. 1979. A minimum linear arrangement algorithm for undirected trees. *SIAM J. Comput.*, 8(1):15–32.

Anders Søgaard. 2020. Some languages seem easier to parse because their treebanks leak. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2765–2770, Online. Association for Computational Linguistics.

Yuanhe Tian, Fei Xia, and Yan Song. 2024. Large Language Models Are No Longer Shallow Parsers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7131–7142, Bangkok, Thailand. Association for Computational Linguistics.

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

Ke Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288, Hong Kong, China. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023. Zero-shot cross-lingual summarization via large language models. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 12–23, Singapore. Association for Computational Linguistics.

Weizhi Wang, Zhirui Zhang, Yichao Du, Boxing Chen, Jun Xie, and Weihua Luo. 2021. Rethinking zero-shot neural machine translation: From a perspective of latent variables. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4321–4327, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Haoran Xu and Philipp Koehn. 2021. Zero-shot cross-lingual dependency parsing through contextual embedding transformation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 204–213, Kyiv, Ukraine. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, et al. 2024. Universal dependencies 2.14. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with QLoRA. In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, Singapore. Association for Computational Linguistics.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

A Post-processing

Figure 3 breaks down the process of obtaining a valid dependency tree (fully connected, no cycles and only one root) from the raw output of the LLMs. In the second post-processing step, to enforce the unique-root constraint we randomly selected a root from the subset of outgoing arcs from node `0`, or just a random node if there are no such arcs. Out-of-range arcs were resolved by replacing the head with the root node. To break cycles and connect all components, our post-processing algorithm performs a breadth-first search from the root node, removing those arcs that create cycles and connecting disconnected nodes to the root node.

Prompt example

```
In dependency parsing the CoNLL format for the sentence <The
trial begins again Nov 28 .> is:
1 The _ _ _ _ 2 det _ _
2 trial _ _ _ _ 3 nsubj _ _
3 begins _ _ _ _ 0 root _ _
4 again _ _ _ _ 3 advmod _ _
5 Nov. _ _ _ _ 3 obl:tmod _ _
6 28 _ _ _ _ 5 nummod _ _
7 . _ _ _ _ 3 punct _ _
Now return the CoNLL format for the sentence: <What if Google
Morphed Into GoogleOS ?>
```

Raw output

```
Sure! This is the CoNLL format for the sentence <What if
Google Morphed Into GoogleOS ?>
1 What _ _ _ 0 nsubj _ _
2 if _ _ _ _ 4 mark _ _ _
3 Google _ _ _ _ 4 nsubj
4 Morphed _ _ _ _ 0 root _ _
5 into _ _ _ _ 6 case _ _
6 GoogleOS _ _ _ _ 8 nmod _ _
7 ? _ _ _ _ 4 punct _ _
Let me know if (...)
```

(1) Well-formatted output

```
1 What _ _ _ _ 0 nsubj _ _
2 if _ _ _ _ 4 mark _ _ _
3 Google _ _ _ _ 4 nsubj _ _
4 Morphed _ _ _ _ 0 root _ _
5 into _ _ _ _ 6 case _ _
6 GoogleOS _ _ _ _ 8 nmod _ _
7 ? _ _ _ _ 4 punct _ _
```

(2) Valid dependency tree

```
1 What _ _ _ _ 4 nsubj _ _
2 if _ _ _ _ 4 mark _ _ _
3 Google _ _ _ _ 4 nsubj _ _
4 Morphed _ _ _ _ 0 root _ _
5 into _ _ _ _ 6 case _ _
6 GoogleOS _ _ _ _ 4 nmod _ _
7 ? _ _ _ _ 4 punct _ _
```

Figure 3: Dependency parsing prompt and the resulting tree after the second post-processing step. Figure 1 showed the original tree.

B Additional results

Table 3 shows the reference to each model used in our experimental study. All of them are publicly available in HuggingFace. Tables 4 to 7 show the performance of each approach aggregating the prediction by its part-of-speech tag and Table 8 breaks down the ratio of post-processing steps performed in each experiment. Figures 4 to 6 display the performance on the French$_{\text{GSD}}$, German$_{\text{GSD}}$ and Hind$_{\text{HDBT}}$ treebanks with respect to dependency displacement (signed dependency distance), following the definition of Anderson and Gómez-Rodríguez (2022), i.e., dependent index minus head index.

Abbrv.	Repository
v1-2b	google/gemma-2b
v1-7b	google/gemma-7b
v2-9b	google/gemma-2-9b
v2-27b	google/gemma-2-27b
v2-7b	meta-llama/Llama-2-7b-chat-hf
v2-13b	meta-llama/Llama-2-13b-chat-hf
v2-70b	meta-llama/Llama-2-70b-chat-hf
v3-8b	meta-llama/Meta-Llama-3-8B-Instruct
v3-70b	meta-llama/Meta-Llama-3-70B-Instruct
v3.1-8b	meta-llama/Llama-3.1-8B-Instruct
v3.1-70b	meta-llama/Llama-3.1-70B-Instruct
v3.2-1b	meta-llama/Llama-3.2-1B-Instruct
v3.2-3b	meta-llama/Llama-3.2-3B-Instruct
v1-7b	mistralai/Mistral-7B-Instruct-v0.1
v2-7b	mistralai/Mistral-7B-Instruct-v0.2
v3-7b	mistralai/Mistral-7B-Instruct-v0.3
x1-7b	mistralai/Mixtral-8x7B-Instruct-v0.1
x1-22b	mistralai/Mixtral-8x22B-Instruct-v0.1
nemo	mistralai/Mistral-Nemo-Instruct-2407
large	mistralai/Mistral-Large-Instruct-2407

Table 3: HuggingFace reference to the instruction-based models used in our experiments.

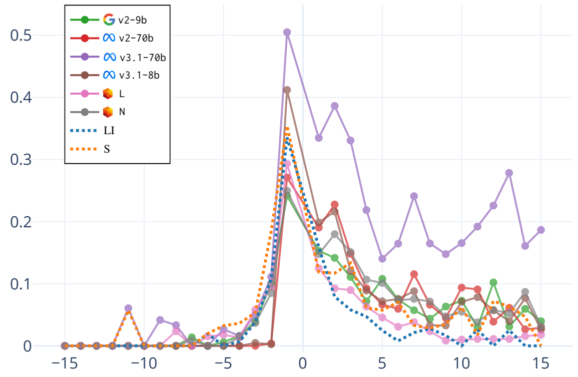

Figure 4: F-score across displacements in the French$_{\text{GSD}}$ test set.

C Official language support

Table 9 shows which of our four target languages are supported by each of the models we used, according to the official documentation provided with each model.

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
A	9.45	6.05	7.74	6.35	5.16	6.96	25.00	8.56	10.70	6.32	6.43	13.29	9.27	3.65	16.51	6.33	5.13
L	6.26	5.90	14.80	1.94	0.14	0.58	24.17	6.16	22.14	11.56	13.47	23.11	21.12	1.04	14.68	8.41	53.85
R	51.06	36.28	43.62	44.72	46.68	55.67	30.00	16.00	26.57	73.50	37.34	22.29	15.28	16.67	25.69	2.19	2.56
RD	8.56	6.29	8.84	6.35	5.70	6.06	18.33	8.53	9.41	6.63	7.50	12.37	10.50	3.39	21.10	6.79	20.51
RD*	8.95	6.00	8.42	6.74	5.56	5.80	21.67	7.88	11.62	7.70	6.43	13.00	10.66	4.95	11.93	6.18	10.26
LI	23.94	15.88	25.00	22.62	14.79	22.83	30.83	12.11	22.14	29.74	24.85	20.65	15.12	11.72	11.93	7.68	25.64
LI*	25.11	19.03	25.26	23.59	19.13	26.20	25.83	11.34	19.56	28.97	27.12	17.62	14.41	9.11	14.68	3.72	15.38
S	24.94	23.70	20.49	24.11	22.25	24.78	35.83	14.74	17.71	27.43	21.52	21.52	18.38	11.72	18.35	9.75	17.95
`v1-2b`	9.45	7.23	8.16	9.92	6.24	8.43	15.83	7.64	11.07	10.94	9.02	12.18	9.04	5.47	6.42	8.18	10.26
`v1-7b`	10.40	6.59	9.35	10.95	7.19	9.17	14.17	7.83	8.12	10.48	11.61	9.82	9.63	7.81	12.84	8.45	5.13
`v2-9b`	15.10	7.62	11.14	10.24	8.01	10.86	16.67	10.95	7.38	8.94	9.90	14.16	6.75	4.69	9.17	19.65	2.56
`v2-7b`	8.22	4.57	10.54	12.70	3.80	5.69	10.00	8.34	4.06	9.71	8.75	9.39	6.94	3.91	8.26	14.55	2.56
`v2-13b`	10.46	1.92	6.46	5.57	1.49	2.11	15.00	10.80	2.40	5.39	3.93	14.68	6.23	2.34	6.42	27.06	0.00
`v2-70b`	10.07	7.42	14.88	14.32	4.34	7.64	10.00	12.79	4.80	14.33	12.36	10.21	10.14	5.73	4.59	13.90	7.69
`v3-8b`	9.56	5.31	11.56	12.70	4.21	6.54	16.67	9.86	4.43	11.40	9.81	10.83	10.59	3.65	9.17	13.28	10.26
`v3-70b`	36.13	13.82	38.69	16.59	8.68	38.22	42.50	30.41	11.62	36.52	33.83	27.15	24.64	8.33	15.60	30.67	17.95
`v3.1-8b`	16.00	16.08	25.17	12.96	23.20	19.93	27.50	15.93	9.78	20.65	21.29	24.03	17.44	7.29	19.27	14.40	30.77
`v3.1-70b`	42.28	20.26	43.45	23.46	21.98	40.48	45.83	29.68	11.44	51.00	37.90	28.50	28.97	11.46	16.51	30.63	12.82
`v3.2-1b`	7.94	5.56	11.56	12.96	5.16	6.48	13.33	8.22	5.17	10.32	9.53	9.87	9.27	5.47	11.01	9.71	10.26
`v3.2-3b`	8.22	3.83	9.10	9.53	3.26	4.53	20.83	9.02	8.12	7.86	7.91	12.61	11.18	3.39	13.76	9.52	10.26
`v1-7b`	10.85	4.77	8.50	9.40	4.48	5.06	20.00	9.79	4.61	8.17	8.65	11.94	6.17	2.34	5.50	16.47	5.13
`v2-7b`	21.09	16.91	16.33	18.02	16.69	24.30	22.50	13.85	8.67	19.72	17.95	17.38	10.53	11.98	13.76	11.09	5.13
`v3-7b`	30.76	24.68	27.81	27.28	24.42	30.89	15.00	13.85	8.67	37.60	21.84	17.00	13.15	11.72	19.27	9.79	12.82
`x1-7b`	13.40	10.75	13.53	12.57	12.25	12.68	15.22	9.15	10.70	15.57	14.64	11.87	8.39	4.50	11.96	10.53	12.90
`x1-22b`	27.57	18.78	31.21	22.88	18.72	29.20	30.83	18.01	13.10	31.12	24.62	23.06	18.86	14.58	18.35	16.51	28.21
`nemo`	13.42	7.62	10.29	9.07	7.33	9.80	20.00	11.31	6.27	8.17	9.16	15.50	8.59	3.65	9.17	19.23	20.51
`large`	21.48	5.31	15.99	9.27	6.78	10.12	25.83	18.27	10.52	10.48	11.01	22.82	11.95	2.60	19.27	38.85	15.38

Table 4: UAS aggregated by universal part-of-speech tag in the English$_{\text{EWT}}$ test set.

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
A	4.27	3.65	3.70	5.01	3.61	3.85	22.22	4.50	3.98	4.83	2.65	4.47	1.56	0.00	4.14	0.00
L	51.23	1.08	23.82	0.84	0.00	0.81	22.22	5.57	19.03	4.47	25.92	11.80	10.94	5.13	9.62	42.86
R	19.38	36.49	36.55	55.43	29.72	89.79	0.00	1.61	44.25	39.00	2.24	15.09	3.91	23.08	0.00	3.57
RD	4.60	4.53	4.93	4.18	5.22	3.92	0.00	4.39	3.98	5.01	5.71	4.97	1.56	2.56	4.38	3.57
RD*	3.28	4.46	5.54	2.51	2.41	3.85	11.11	4.18	3.10	4.11	4.29	4.47	3.91	0.00	5.36	3.57
LI	28.41	15.74	23.41	21.73	11.65	30.22	0.00	5.47	20.35	20.21	13.67	10.46	5.47	15.38	5.24	17.86
LI*	27.09	18.38	22.18	23.40	14.46	35.63	0.00	4.98	25.22	21.11	14.90	11.21	6.25	15.38	4.26	28.57
S	13.79	24.73	21.15	26.46	18.88	34.28	0.00	9.00	21.24	21.11	9.39	12.56	7.81	15.38	7.43	14.29
`v1-2b`	10.67	6.82	11.50	10.58	6.43	12.24	0.00	5.79	12.39	8.59	9.39	6.75	3.91	10.26	9.01	10.71
`v1-7b`	10.84	9.05	11.91	9.47	5.62	13.52	0.00	4.93	9.29	9.84	7.35	5.99	3.12	5.13	7.92	14.29
`v2-9b`	16.42	9.86	11.29	11.42	7.23	19.27	11.11	9.38	3.10	11.45	8.57	6.32	2.34	7.69	17.78	3.57
`v2-7b`	10.67	16.15	17.25	26.46	13.25	40.70	22.22	4.72	19.91	22.00	5.10	8.77	2.34	17.95	2.56	0.00
`v2-13b`	9.85	11.82	17.45	14.21	8.84	27.86	0.00	5.31	11.50	15.56	5.10	8.35	3.12	10.26	5.72	3.57
`v2-70b`	30.71	8.18	21.77	6.69	3.21	24.95	11.11	17.42	6.64	15.38	21.84	8.26	6.25	12.82	19.37	21.43
`v3-8b`	13.79	12.30	16.43	17.55	6.83	28.06	11.11	4.02	20.35	17.17	9.18	9.02	1.56	10.26	7.67	3.57
`v3-70b`	41.05	19.73	28.34	8.64	6.43	47.73	11.11	26.74	9.73	22.00	34.08	15.01	5.47	5.13	21.92	28.57
`v3.1-8b`	32.02	17.30	28.95	25.63	22.49	53.75	11.11	11.36	13.72	25.04	11.02	13.58	3.12	12.82	14.86	7.14
`v3.1-70b`	53.20	27.91	37.78	13.93	14.46	59.77	0.00	27.49	15.49	28.09	26.12	23.44	5.47	15.38	29.23	21.43
`v3.2-1b`	7.22	5.41	6.16	3.90	3.61	9.87	11.11	5.47	8.85	8.77	4.69	4.81	3.91	12.82	10.96	0.00
`v3.2-3b`	10.84	5.81	8.21	9.75	2.41	12.24	11.11	7.18	6.64	7.51	10.00	7.00	3.91	5.13	9.01	14.29
`v1-7b`	11.82	6.69	10.06	4.46	4.42	12.17	0.00	6.22	6.19	8.94	9.80	5.90	2.34	7.69	14.98	14.29
`v2-7b`	21.35	19.05	17.45	14.48	12.05	33.60	0.00	11.74	7.96	19.32	11.43	10.12	3.12	20.51	12.55	21.43
`v3-7b`	26.44	29.39	31.62	31.20	21.29	70.05	11.11	10.40	16.37	29.34	13.27	12.14	3.12	17.95	11.21	14.29
`x1-7b`	16.09	6.42	9.03	6.69	4.02	15.89	22.22	10.93	6.64	7.69	7.76	6.16	5.47	5.13	27.65	0.00
`x1-22b`	29.23	20.61	26.28	14.76	17.27	38.54	22.22	14.47	18.58	21.29	17.76	13.24	7.03	23.08	14.98	7.14
`nemo`	17.24	10.34	14.17	8.64	6.43	20.76	11.11	11.95	4.42	11.63	13.88	8.94	3.91	5.13	19.12	10.71
`large`	12.97	13.38	14.99	16.43	11.24	25.42	0.00	5.63	16.37	16.28	8.37	8.60	5.47	7.69	6.21	7.14

Table 5: UAS aggregated by universal part-of-speech tag in the French$_{\text{GSD}}$ test set.

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
A	5.36	4.61	4.36	5.51	3.46	5.26	0.00	4.85	6.01	5.24	3.83	6.26	4.52	4.35	0.00	4.07	16.00
L	1.75	1.18	12.31	23.77	3.68	0.80	25.00	2.28	9.87	10.48	19.72	30.04	23.13	0.00	25.00	1.58	28.00
R	69.43	29.95	34.35	6.23	43.29	68.97	0.00	15.49	63.52	47.62	17.16	8.51	10.49	1.86	50.00	2.64	24.00
RD	4.77	5.73	6.85	5.80	6.49	4.95	25.00	6.33	4.72	4.76	5.25	6.36	6.22	5.59	0.00	6.18	16.00
RD*	6.33	5.04	5.92	6.23	5.84	5.31	0.00	6.62	4.29	5.71	5.39	7.44	5.88	8.07	0.00	5.35	12.00
LI	26.39	15.13	18.07	16.23	16.88	28.69	25.00	9.58	34.33	11.43	15.60	18.10	17.25	1.24	0.00	5.05	24.00
LI*	27.85	15.75	22.12	17.97	16.67	31.83	0.00	8.39	31.76	20.00	20.71	18.59	14.08	1.86	0.00	3.77	20.00
S	29.11	18.56	18.69	12.03	16.45	26.57	0.00	12.18	24.46	25.71	15.60	10.67	11.63	3.73	25.00	7.69	20.00
v1-2b	17.33	10.46	13.71	6.23	9.96	20.78	0.00	7.68	17.60	10.00	11.91	9.39	9.43	4.97	0.00	6.93	8.00
v1-7b	26.29	11.71	14.95	3.33	16.02	27.94	0.00	8.07	11.16	16.67	9.50	7.63	6.22	0.00	50.00	20.12	12.00
v2-9b	18.31	10.40	12.85	5.65	8.66	18.52	0.00	7.39	8.15	13.33	8.09	10.67	5.96	3.11	25.00	22.76	16.00
v2-7b	32.23	16.56	19.39	6.38	21.21	35.15	0.00	10.03	25.75	17.62	12.77	9.00	9.22	1.24	50.00	4.90	4.00
v2-13b	34.54	18.22	23.13	5.58	18.94	36.85	0.00	10.17	28.10	18.10	12.19	9.81	9.32	0.00	0.00	4.35	15.38
v2-70b	42.16	18.24	28.12	11.74	21.43	46.82	0.00	14.01	12.88	24.29	18.44	18.49	13.83	1.24	0.00	13.26	28.00
v3-8b	58.13	25.34	29.36	6.96	30.74	58.49	0.00	13.47	23.18	29.52	18.72	11.94	13.95	1.24	50.00	17.41	28.00
v3-70b	48.69	24.22	31.78	7.25	11.26	47.92	0.00	25.81	18.45	32.38	24.26	29.75	17.93	1.86	0.00	30.22	32.00
v3.1-8b	45.86	20.55	28.35	9.28	27.92	47.44	0.00	13.79	17.17	30.00	21.70	20.35	15.43	1.86	25.00	13.79	28.00
v3.1-70b	60.76	26.40	34.19	16.38	23.81	53.80	25.00	27.19	20.60	39.52	23.40	27.50	24.36	1.86	0.00	27.51	24.00
v3.2-1b	10.42	6.54	9.66	6.38	5.41	14.32	0.00	6.88	10.30	7.14	12.34	8.02	8.54	2.48	25.00	10.85	12.00
v3.2-3b	30.19	15.88	18.07	6.81	21.86	28.65	25.00	10.77	23.61	20.00	12.62	10.37	8.54	2.48	25.00	9.27	20.00
v1-7b	21.71	13.14	13.24	4.35	14.29	23.43	0.00	8.13	9.44	11.43	11.77	10.27	7.23	1.24	0.00	16.28	20.00
v2-7b	33.79	18.12	21.81	4.64	19.48	35.06	0.00	11.06	15.45	18.57	12.77	13.21	9.30	1.24	0.00	14.32	12.00
v3-7b	62.51	27.90	31.31	7.97	37.88	61.54	0.00	13.95	24.89	37.62	18.01	10.47	11.59	0.62	25.00	8.89	12.00
x1-7b	11.11	0.00	30.00	8.33	0.00	30.43		13.51	0.00	0.00	22.22	0.00	18.75	0.00		0.00	0.00
x2-22b	30.19	18.80	21.88	9.42	16.23	32.40	0.00	13.02	19.31	20.48	19.29	17.03	13.49	3.11	0.00	13.34	32.00
nemo	11.88	6.54	10.20	5.36	6.93	13.04	0.00	9.03	4.29	9.52	6.81	11.06	6.43	0.62	25.00	27.51	20.00
large	21.81	7.60	11.99	4.78	6.49	14.90	25.00	12.63	6.87	11.90	8.09	12.23	10.23	0.62	25.00	36.17	24.00

Table 6: UAS aggregated by universal part-of-speech tag in the German$_{\text{GSD}}$ test set.

	ADJ	ADP	ADV	AUX	CCONJ	DET	NOUN	NUM	PART	PRON	PROPN	PUNCT	SCONJ	VERB	X
A	4.56	4.88	4.93	4.77	5.83	5.77	4.99	2.89	6.06	5.39	4.44	5.41	2.29	4.48	11.11
L	6.34	83.96	0.66	68.81	0.00	0.00	0.09	0.00	40.32	0.29	0.09	16.78	0.76	4.97	11.11
R	73.13	0.07	22.37	0.13	27.09	83.09	25.65	75.32	28.06	32.87	31.70	13.55	0.00	5.03	55.56
RD	5.40	4.58	3.29	5.73	3.78	4.70	4.99	4.76	5.47	5.90	4.15	4.96	3.82	4.97	0.00
RD*	4.84	4.68	3.29	5.64	5.51	6.44	4.68	4.91	5.02	5.76	4.19	5.04	3.97	5.66	0.00
LI	28.98	27.87	8.55	38.46	10.08	40.13	11.14	30.30	25.11	15.89	16.34	17.02	0.76	7.08	22.22
LI*	30.20	28.24	9.87	32.94	12.44	37.85	12.12	27.99	23.93	16.69	15.93	13.22	0.92	7.71	33.33
S	23.91	23.91	11.84	27.97	10.71	24.83	13.80	19.19	19.79	15.52	13.36	16.20	4.89	11.70	11.11
v1-2b	9.77	23.07	5.26	15.43	3.15	12.62	4.74	8.23	13.59	6.78	6.35	8.02	1.83	5.58	11.11
v1-7b	16.16	8.24	6.25	4.85	6.77	21.34	6.26	11.69	8.12	8.82	8.54	4.71	0.00	15.75	0.00
v2-9b	13.76	17.97	7.89	13.34	5.35	16.11	6.63	11.54	12.11	8.82	8.18	4.30	1.22	18.00	22.22
v2-7b	14.51	12.82	5.92	11.08	5.51	21.34	5.86	12.41	10.49	9.55	9.08	7.81	1.37	5.14	0.00
v2-13b	9.91	32.48	4.61	22.66	2.68	18.12	3.96	7.22	16.10	9.91	6.17	9.34	1.83	5.26	0.00
v2-70b	10.77	28.76	7.32	20.89	4.60	22.68	3.77	10.00	19.72	9.14	8.22	10.45	0.00	3.94	
v3-8b	20.81	28.20	13.16	23.91	5.51	28.99	6.97	12.70	24.52	11.88	8.29	6.86	0.15	16.32	11.11
v3-70b	20.61	16.79	11.51	13.01	8.92	22.76	10.29	17.99	16.13	14.98	12.95	8.22	1.03	5.70	0.00
v3.1-8b	25.27	30.57	12.50	32.86	7.87	38.79	9.83	22.37	20.09	17.42	11.69	7.02	0.00	13.55	11.11
v3.1-70b	17.67	18.19	6.47	15.46	11.52	22.44	10.38	12.95	18.28	13.69	12.79	9.01	2.40	4.19	0.00
v3.2-1b	8.08	6.87	2.96	6.19	5.98	11.14	4.23	6.49	5.17	8.75	5.36	6.40	1.07	7.25	0.00
v3.2-3b	7.56	35.68	5.92	22.49	1.26	8.46	2.86	3.46	16.25	4.74	4.53	7.07	1.07	8.84	11.11
v1-7b	9.53	19.55	4.28	6.44	2.68	13.42	3.68	5.63	9.16	7.14	6.08	2.15	0.15	17.74	0.00
v2-7b	15.88	22.49	11.51	9.49	4.88	22.01	7.14	10.53	17.58	8.45	8.29	3.88	0.46	18.64	0.00
v3-7b	29.64	30.29	16.12	18.31	8.98	44.56	11.01	26.84	21.42	14.58	14.98	6.16	0.46	16.73	22.22
x1-7b	29.64	30.29	16.12	18.31	8.98	44.56	11.01	26.98	21.42	14.58	14.96	6.16	0.46	16.73	22.22
x1-22b	28.64	33.08	9.83	20.04	13.70	33.33	11.79	27.62	17.60	16.85	15.19	10.49	1.53	9.16	16.67
nemo	9.58	11.84	4.61	7.90	3.31	10.07	4.99	7.36	11.37	4.96	5.16	5.12	0.92	20.69	11.11
large	25.37	14.66	11.51	9.14	12.64	29.81	10.38	21.94	16.49	13.20	13.77	8.02	1.03	5.43	0.00

Table 7: UAS aggregated by universal part-of-speech tag in the Hindi$_{\text{HDTB}}$ test set.

		en$_{\text{EWT}}$			fr$_{\text{GSD}}$			de$_{\text{GSD}}$			hi$_{\text{HDTB}}$		
		NP	**P1**	**P2**	**NP**	**P1**	**P2**	**NP**	**P1**	**P2**	**NP**	**P1**	**P2**
G	v1-2b	13.38	16.18	70.44	7.45	14.90	77.64	7.06	12.90	80.04	6.06	16.27	77.67
	v1-7b	14.44	16.51	69.04	6.97	10.82	82.21	10.75	0.00	89.25	8.43	0.12	91.45
	v2-9b	15.17	0.05	84.79	4.81	0.00	95.19	5.32	0.00	94.68	3.27	0.00	96.73
∞	v2-7b	24.22	0.00	75.78	1.92	2.64	95.43	4.09	8.70	87.21	5.82	5.23	88.95
	v2-13b	14.59	0.00	85.41	22.12	10.58	67.31	32.89	5.12	61.98	18.71	13.18	68.11
	v2-70b	46.89	0.00	53.11	13.70	0.00	86.30	23.34	0.10	76.56	7.47	8.71	83.82
	v3-8b	49.01	0.00	50.99	4.81	6.25	88.94	22.42	0.00	77.58	0.48	0.00	99.52
	v3-70b	58.69	0.10	41.21	33.41	0.00	66.59	28.25	0.00	71.75	24.76	12.72	62.52
	v3.1-8b	34.38	0.00	65.62	5.29	0.48	94.23	12.90	0.00	87.10	2.43	0.00	97.57
	v3.1-70b	65.86	0.05	34.09	42.79	0.24	56.97	46.57	0.00	53.43	26.14	11.34	62.52
	v3.2-1b	5.63	14.06	80.31	2.16	8.41	89.42	1.84	9.72	88.43	1.37	8.19	90.44
	v3.2-3b	5.15	14.68	80.16	2.40	10.58	87.02	3.99	9.01	87.00	2.73	12.17	85.10
■	v1-7b	16.85	0.00	83.15	5.29	0.00	94.71	5.94	0.00	94.06	0.12	0.00	99.88
	v2-7b	15.12	0.00	84.88	3.85	0.00	96.15	5.02	0.00	94.98	0.36	0.00	99.64
	v3-7b	28.17	0.05	71.79	11.06	0.00	88.94	19.55	0.00	80.45	0.71	0.00	99.29
	x1-7b	26.22	13.27	60.51	3.37	0.00	96.63	25.00	12.50	62.50	1.25	0.06	98.69
	x1-22b	41.98	29.22	28.79	32.69	26.92	40.38	37.46	26.41	36.13	30.88	33.02	36.10
	nemo	15.74	0.00	84.26	3.61	0.00	96.39	4.09	0.00	95.91	0.48	0.00	99.52
	large	18.25	0.00	81.75	22.84	15.14	62.02	7.88	0.00	92.12	18.26	11.20	70.54

Table 8: Distribution of the amount of post-processing steps performed in each zero-shot parser. **NP** represents the ratio of generated trees that did not require post-processing (only removing non-tabular lines), **P1** for those trees that only required the first post-processing step (e.g. removing extra columns) and **P2** for those trees that required of the full post-processing step (e.g. breaking cycles).

Model		**English**	**French**	**German**	**Hindi**
G	v1-2b	✓			
	v1-7b	✓			
	v2-9b	✓			
∞	v2-7b	✓			
	v2-13b	✓			
	v2-70b	✓			
	v3-8b	✓	✓		✓
	v3-70b	✓	✓		✓
	v3.1-8b	✓	✓	✓	✓
	v3.1-70b	✓	✓	✓	✓
	v3.3-1b	✓	✓	✓	✓
	v3.3-3b	✓	✓	✓	✓
■	v1-7b	✓	✓		
	v2-7b	✓			
	v3-7b	✓			
	x1-7b	✓	✓	✓	
	nemo	✓	✓	✓	✓
	large	✓	✓	✓	✓

Table 9: Language support across different models. A tick symbol (✓) indicates that the model supports the respective language, while empty cells indicate lack of support.

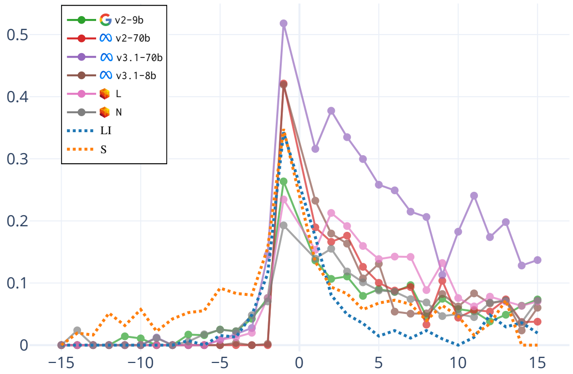

Figure 5: F-score across displacements in the German$_{\text{GSD}}$ test set.

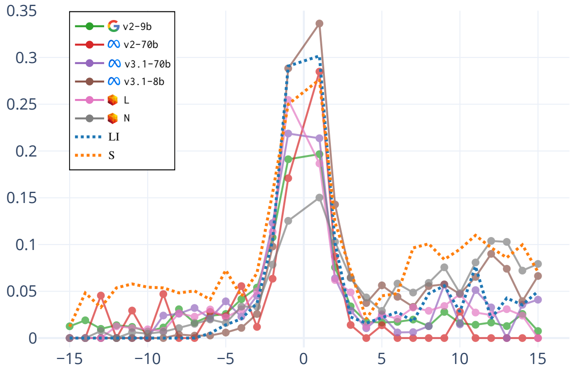

Figure 6: F-score across displacements in the Hindi$_{\text{HDTB}}$ test set.