Lost in Formatting: How Output Formats Skew LLM Performance on Information Extraction

Rishi Ravikumar, Nuhu Ibrahim and Riza Batista-Navarro
Department of Computer Science, University of Manchester, United Kingdom
{rishi.ravikumar, nuhu.ibrahim, riza.batista}@manchester.ac.uk

Abstract

We investigate how the choice of output format influences the performance of fine-tuned large language models on information extraction tasks. Based on over 280 experiments spanning multiple benchmarks, models and formats, we find that output formatting is a critical yet largely overlooked hyperparameter. Remarkably, in some cases, changing only the output format shifts F1 scores by over 40% despite using the same model. We further observe that no single format consistently dominates across settings, and the optimal choice depends on factors like model family and dataset characteristics. Overall, these results demonstrate that *informationally equivalent* output formats can produce substantial performance variation, highlighting the need to treat output formatting as a key factor in building accurate and reliable information extraction systems.

1 Introduction

Information extraction (IE) systems, powered by Large Language Models (LLMs), are increasingly deployed in high-stakes domains such as biomedicine, where errors in extraction can suppress critical information and limit downstream applications (Tian et al., 2024). The scale of the challenge is amplified by volume: PubMed alone indexes tens of millions of articles, with thousands added daily (Lee et al., 2019). While biomedicine provides an especially urgent case, this vulnerability is universal across IE applications, from cybersecurity to finance, where incomplete or inaccurate extraction cascades through pipelines, systematically distorting downstream models and decision processes.

LLMs have dramatically advanced the accuracy and efficiency of IE systems (Adam et al., 2024), yet they remain strikingly fragile. Superficial factors that do not alter task requirements or difficulty can still cause large swings in performance. For instance, outputs vary substantially depending on prompt phrasing or structuring (He et al., 2024), and class predictions shift significantly when label sets change, even when they encode the same decision boundary (Lu et al., 2025). In this paper, we focus on **output formatting**—a seemingly trivial design choice in LLM fine-tuning that does not affect task requirements or difficulty—and show that it exposes fundamental fragilities in LLMs. On the biomedical BC5CDR dataset (Li et al., 2016), for example, changing only the output format results in the model missing over 40% of entity mentions. Such omissions cascade through downstream pipelines, undermining reliability across domains where accurate extraction is essential.

Despite its substantial effect, *output formatting is almost always overlooked*. Most studies adopt a single default format, typically motivated by convenience or legacy convention, without justification or systematic comparison (Dagdelen et al., 2024; Wang et al., 2023; Guluzade et al., 2025; Zhang et al., 2024b; Dukić and Snajder, 2024; Zhang et al., 2024a). Only a few studies have investigated the effects of output formatting, but these focus exclusively on legacy encoder–decoder LLMs (Raman et al., 2022; He and Choi, 2023) and the experiments are restricted to narrow task sets and limited benchmarks. Our work provides a timely and necessary extension, offering the first large-scale study of output formatting effects in modern decoder-only LLMs, across diverse model families, sizes, and benchmarks spanning Named Entity Recognition (NER), Event Detection (ED), and Semantic Role Labelling (SRL).

Our experiments reveal that output formatting is a critical hyperparameter and that untapped performance gains can be realised through the careful selection of output format. Importantly, we find no universally optimal format; the best choice depends on factors such as dataset characteristics and model family.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5498–5513
March 24-29, 2026 ©2026 Association for Computational Linguistics

This work therefore provides: (1) an evaluation framework for systematically assessing the effect of output formatting on LLMs[1]; (2) a large-scale empirical study across multiple formats, benchmarks and models; and (3) insights into when and why different formats succeed or fail. Taken together, these contributions provide actionable guidance for researchers and practitioners: until LLMs become robust to format variation, output formatting should be treated as a key hyperparameter for reliable and accurate information extraction. More broadly, our findings highlight the importance of developing models that are less mercurial and more grounded in the underlying semantics of a task.

2 Related Work

It is well established that prompt formatting can significantly influence model responses (Tam et al., 2024; He et al., 2024; Sclar et al., 2024; Liu et al., 2025; Zhang et al., 2025). Similar dynamics are likely involved in output formatting, yet its role in shaping model responses has received comparatively little attention.

Some prior works have examined this question in the context of legacy encoder–decoder models, focusing on sequence labelling tasks. Raman et al. (2022) study the impact of output formatting on mT5 (Xue et al., 2021), evaluating several formats across tasks such as NER and semantic parsing. They identify a format—referred to in this paper as *token-level*—as the 'empirically strongest' option; however, our experiments demonstrate that it does not always yield optimal performance. Moreover, the benchmarks they employ are relatively simple, with even mBERT (Devlin et al., 2019) achieving F1 scores of over 80% on most tasks, which limits the strength of their conclusions.

He and Choi (2023), in contrast, investigate output formatting effects on BART (Lewis et al., 2020), across tasks such as NER, part-of-speech tagging and dependency parsing. Unlike Raman et al. (2022), they report no universally superior format. However, like the former study, their analysis remains confined to a single model, tested on a narrow set of benchmarks and formats.

Together, these works provide early evidence that output formatting matters, but their findings are increasingly outdated: neither explores whether the effect generalises across different model families and sizes, nor whether it persists in modern decoder-only LLMs that now dominate the field.

A more recent attempt in this direction is by Guo et al. (2024), who seek to identify the 'ideal training sample' for fine-tuning LLMs on IE tasks. Their analysis considers factors such as instruction format, output format and chain-of-thought, aiming to find configurations that maximise extraction quality. However, the study is narrow in scope: it evaluates only a small set of highly similar output formats on a handful of decoder models of comparable size, leaving much of the design space unexplored. Notably, the format they identify as strongest is, in our experiments, consistently among the weakest-performing.

3 Evaluation Framework

3.1 Benchmark Datasets

We focus on three sequence-labelling–based IE tasks: Event Detection, Named Entity Recognition and Semantic Role Labelling. These tasks are both fundamental to information extraction and widely studied, making them particularly suitable for this investigation.

For each task, we select two benchmarks that vary along three dimensions: (1) domain, (2) recency of creation and, (3) label set size and complexity. Varying domains ensures that our findings generalise across different contexts and application areas. Including both older and more recent benchmarks increases the temporal diversity of the study, while variation in label set size and granularity captures tasks of differing difficulty and ambiguity. Specifically, for ED we use ACE2005 (Doddington et al., 2004) and PHEE (Sun et al., 2022); for NER, OntoNotes-5.0 (Weischedel et al., 2013) and BC5CDR (Li et al., 2016); and for SRL, SpRL-2012 (Kordjamshidi et al., 2012) and NounAtlas (Navigli et al., 2024). Together, this selection provides a diverse and representative evaluation suite. Table 1 presents an overview of the benchmarks, including statistics such as average sample length and average number of annotations per sample. Task definitions and details of the pre-processing steps are provided in Appendix A.3.

3.2 Output Formats

Drawing on prior research and established corpus annotation conventions (see Appendix A.2), we systematically design six output formats.

[1] Our code is available at https://github.com/rishi-ravikumar/OutputFormattingExperiments

Task	Benchmark	Year	Domain	# Categories	# Samples (After Processing)	Avg. Sample Length (Tokens)	Avg. Annotations Per Sample	Avg. Annotation Length (Tokens)
ED	ACE2005	2005	News/Mixed	33	3998	22.4	1.3	1.0
	PHEE	2022	Pharmaceuticals	2	4827	22.1	1.0	1.1
NER	OntoNotes-5.0	2013	Mixed	36	42193	23.8	2.5	2.2
	BC5CDR	2016	Biomedical	2	1500	187.9	19.2	1.3
SRL	SpRL-2012	2012	Spatial Cognition	3	948	17.5	4.5	1.3
	NounAtlas	2024	Mixed	27	28064	34.8	2.5	4

Table 1: Overview of benchmarks used in this study. Token counts are computed using whitespace-delimited tokens.

We use the following simplified NER example sentence to illustrate the formats:

```
Alice visited Paris.
```

After each format description, we show how the extracted entities from this sentence are represented in that format.

Standoff JSON: The output is formatted as a list of JSON objects, where each object contains a span and its corresponding label. Variations of this format have been used in Ying et al. (2025); joon Choi and jun Park (2025); Guo et al. (2024); Dagdelen et al. (2024); Xiao et al. (2024); NuMind (2024); Raman et al. (2022); Yan et al. (2021); Ahmad et al. (2021); Wu et al. (2023).

```
[{ “entity_type”: “PER”, “entity_span”: “Alice” },
 { “entity_type”: “GPE”, “entity_span”: “Paris” }]
```

Standoff Tuple: The output is formatted as a list of tuples, where each tuple consists of a span and its corresponding label. Standoff Tuple is particularly similar to Standoff JSON, differing slightly in the presentation of span-label pairs. Variations of this format have been used in Lu et al. (2021); Li et al. (2023); Wang et al. (2022, 2023); Josifoski et al. (2022).

```
((Alice; PER) (Paris; GPE))
```

Inline XML: The output contains the original text, with relevant spans enclosed in XML tags. Variations of this format have been used in Athiwaratkun et al. (2020); Paolini et al. (2021); Wang et al. (2025); He and Choi (2023); Daza and Frank (2018).

```
<PER> Alice </PER> visited <GPE> Paris </GPE> .
```

Inline Linear: The output contains the original text, with each token accompanied by its label in the `token|label` format. Labels are assigned using the BIO (Begin-Inside-Outside) scheme. Variations of this format have been used in (He and Choi, 2023; Raman et al., 2022).

```
Alice|B-PER visited|O Paris|B-GPE .|O
```

Column: Inspired by column-based annotation schemes (see Appendix A.2), the output is formatted such that each token from the original text and its BIO label appear adjacently, delimited by a space. Each token-label pair is separated by a newline character. Inline Linear and Column are particularly similar, differing only in minor syntactic presentation.

```
Alice B-PER\nvisited O\nParis  B-GPE\n. O
```

Freetext: The output is expressed as natural-language phrases. This format is motivated by the observation that LLMs are likely to understand and generate natural-looking, free-form text more effectively than rigid, structured representations (He and Choi, 2023).

```
‘Alice’ is an entity of type ‘PER’;
‘Paris’ is an entity of type ‘GPE’;
```

When adapting benchmarks to the output formats, we define tokens using whitespace-based tokenization to ensure consistency across models with different tokenizers.

In what follows, we refer to the Standoff JSON and Standoff Tuple formats collectively as the *standoff* formats, and the Inline Linear and Column formats as the *token-level* formats, where appropriate.

Importantly, although the six output formats differ in their syntactic presentation, **they are equivalent in terms of the information they encode**. In all cases, the underlying span-label assignments are identical, and the model has full access to the input sentence, allowing it to, in principle, recover the same information. Given the input sentence, any format can be converted to any other format without requiring new information. The observed differences in performance therefore arise not from variations in content, but from the model's sensitivity to the syntactic structure and representation of the output.

3.3 Models

Model	Parameters (B)
Qwen2.5-3B-Instruct (Qwen et al., 2025)	3
Phi-3.5-mini-instruct (Microsoft, 2024)	3
Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	7
Llama3.1-8B-Instruct (Meta, 2024)	8
Mistral-Nemo-Instruct-2407 (Mistral AI, 2024)	12
Qwen2.5-14B-Instruct (Qwen et al., 2025)	14
Qwen2.5-32B-Instruct (Qwen et al., 2025)	32
OLMo-2-0325-32B-Instruct (AI2, 2025)	32

Table 2: Overview of the LLMs used in this study.

Table 2 presents an overview of the LLMs evaluated in this study. We consider a diverse set of decoder-only models ranging from 3B to 32B parameters and drawn from multiple model families (e.g., Qwen, Phi, Mistral). For efficiency, we perform supervised fine-tuning of all models using QLoRA (Dettmers et al., 2023), with 4-bit–quantized base weights. Further details on hyperparameters are provided in Appendix A.1.

3.4 Prompt Formatting

Prompt design can substantially influence model behavior. To control for this factor, we adopt a minimal prompting strategy across all experiments. For ED, NER and the SpRL-2012 benchmark (SRL), the prompt strictly consists of the raw text only, with no additional instructions, special characters or label/type specifications. For NounAtlas (SRL), where the semantic frame is required as input, the prompt consists of the raw text and the frame, separated by a special delimiter:

```
<text> [[## frame ##]] <frame>
```

This setup ensures that any observed differences in performance arise from output formatting rather than variations in the prompt.

3.5 Evaluation Criteria

We adopt strict evaluation criteria across all experiments: a prediction is considered correct only if both the span and its associated label exactly match the gold annotation. Performance is reported using the micro-averaged F1 score (class-independent), which we refer to throughout simply as the F1 score.

4 Results and Discussion

Figure 1 reports the results,[2] grouped by benchmark. Each cell shows the F1 score obtained by a model fine-tuned with the corresponding output format, with cell colouring indicating relative performance. Precision and recall are reported separately in Tables 5 and 6 (Appendix). Figure 2 summarises these results by plotting, for each benchmark, the mean F1, precision and recall across models for each output format. Based on these results, we identify a set of broad trends that characterise the effect of output formatting across benchmarks and models.

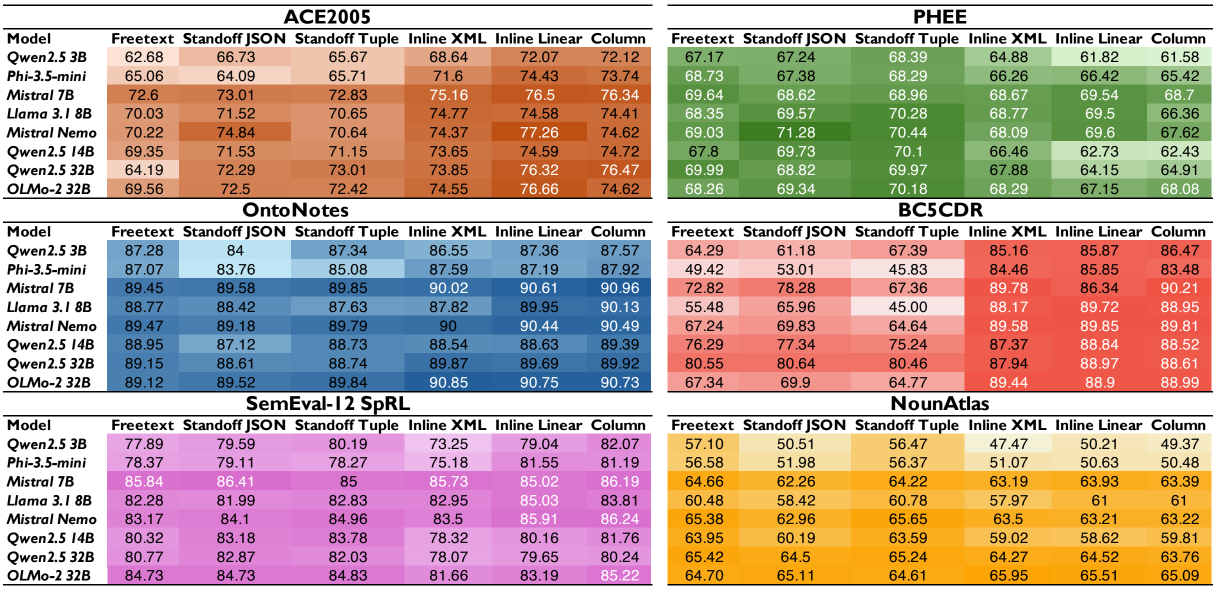
Token-level formats (Inline Linear and Column) outperform standoff formats (Standoff JSON and Standoff Tuple) on half of the benchmarks. On ACE2005, OntoNotes and BC5CDR, the token-level formats generally outperform the standoff formats (Figure 2a, c, d). This difference is particularly pronounced for BC5CDR, reaching over 40% in certain instances.

The underlying reason for the performance disparity stems from the prediction task imposed by each format category. The standoff formats require a model to directly generate relevant spans and their corresponding labels in a structured representation. Conversely, token-level formats require the model to generate each token from the input text, followed by a label. This token-by-token labelling compels a model to explicitly consider the significance and role of every token and *provides dense supervision over non-entity tokens during training*. As a result, the likelihood of missing relevant spans is reduced. In contrast, standoff formats do not enforce this explicit token-level consideration, and non-entity tokens remain implicit, increasing the likelihood of missed spans.

The disparity between the standoff and token-level formats is particularly pronounced for BC5CDR. As shown in Table 1, BC5CDR contains significantly longer samples (on average more than six times longer than those in the dataset with the second longest samples) and has a very high annotation density, with roughly 19 spans per sample. These properties make standoff formats significantly more brittle: relevant spans are frequently omitted because standoff formats encourage a global view of the prediction task. In contrast, token-level formats require the model to make explicit token-wise predictions across the entire sequence, which substantially reduces the chance of missing spans in lengthy texts with many annotations.

Additionally, samples in BC5CDR tend to

[2] A full-page version is available in the Appendix.



	ACE2005						PHEE					
Model	Freetext	Standoff JSON	Standoff Tuple	Inline XML	Inline Linear	Column	Freetext	Standoff JSON	Standoff Tuple	Inline XML	Inline Linear	Column
Qwen2.5 3B	62.68	66.73	65.67	68.64	72.07	72.12	67.17	67.24	68.39	64.88	61.82	61.58
Phi-3.5-mini	65.06	64.09	65.71	71.6	74.43	73.74	68.73	67.38	68.29	66.26	66.42	65.42
Mistral 7B	72.6	73.01	72.83	75.16	76.5	76.34	69.64	68.62	68.96	68.67	69.54	68.7
Llama 3.1 8B	70.03	71.52	70.65	74.77	74.58	74.41	68.35	69.57	70.28	68.77	69.5	66.36
Mistral Nemo	70.22	74.84	70.64	74.37	77.26	74.62	69.03	71.28	70.44	68.09	69.6	67.62
Qwen2.5 14B	69.35	71.53	71.15	73.65	74.59	74.72	67.8	69.73	70.1	66.46	62.73	62.43
Qwen2.5 32B	64.19	72.29	73.01	73.85	76.32	76.47	69.99	68.82	69.97	67.88	64.15	64.91
OLMo-2 32B	69.56	72.5	72.42	74.55	76.66	74.62	68.26	69.34	70.18	68.29	67.15	68.08

	OntoNotes						BC5CDR					
Model	Freetext	Standoff JSON	Standoff Tuple	Inline XML	Inline Linear	Column	Freetext	Standoff JSON	Standoff Tuple	Inline XML	Inline Linear	Column
Qwen2.5 3B	87.28	84	87.34	86.55	87.36	87.57	64.29	61.18	67.39	85.16	85.87	86.47
Phi-3.5-mini	87.07	83.76	85.08	87.59	87.19	87.92	49.42	53.01	45.83	84.46	85.85	83.48
Mistral 7B	89.45	89.58	89.85	90.02	90.61	90.96	72.82	78.28	67.36	89.78	86.34	90.21
Llama 3.1 8B	88.77	88.42	87.63	87.82	89.95	90.13	55.48	65.96	45.00	88.17	89.72	88.95
Mistral Nemo	89.47	89.18	89.79	90	90.44	90.49	67.24	69.83	64.64	89.58	89.85	89.81
Qwen2.5 14B	88.95	87.12	88.73	88.54	88.63	89.39	76.29	77.34	75.24	87.37	88.84	88.52
Qwen2.5 32B	89.15	88.61	88.74	89.87	89.69	89.92	80.55	80.64	80.46	87.94	88.97	88.61
OLMo-2 32B	89.12	89.52	89.84	90.85	90.75	90.73	67.34	69.9	64.77	89.44	88.9	88.99

	SemEval-12 SpRL						NounAtlas					
Model	Freetext	Standoff JSON	Standoff Tuple	Inline XML	Inline Linear	Column	Freetext	Standoff JSON	Standoff Tuple	Inline XML	Inline Linear	Column
Qwen2.5 3B	77.89	79.59	80.19	73.25	79.04	82.07	57.10	50.51	56.47	47.47	50.21	49.37
Phi-3.5-mini	78.37	79.11	78.27	75.18	81.55	81.19	56.58	51.98	56.37	51.07	50.63	50.48
Mistral 7B	85.84	86.41	85	85.73	85.02	86.19	64.66	62.26	64.22	63.19	63.93	63.39
Llama 3.1 8B	82.28	81.99	82.83	82.95	85.03	83.81	60.48	58.42	60.78	57.97	61	61
Mistral Nemo	83.17	84.1	84.96	83.5	85.91	86.24	65.38	62.96	65.65	63.5	63.21	63.22
Qwen2.5 14B	80.32	83.18	83.78	78.32	80.16	81.76	63.95	60.19	63.59	59.02	58.62	59.81
Qwen2.5 32B	80.77	82.87	82.03	78.07	79.65	80.24	65.42	64.5	65.24	64.27	64.52	63.76
OLMo-2 32B	84.73	84.73	84.83	81.66	83.19	85.22	64.70	65.11	64.61	65.95	65.51	65.09

Figure 1: Evaluation results (F1) grouped by benchmark.

contain multiple instances of the same entity. We observe that with the standoff formats, this often leads models, especially smaller models, to fall into a loop, repeatedly predicting the same entity until the maximum sequence length is reached. This tendency to over-predict results in a large number of false positives, severely affecting precision. The token-level formats, in contrast, are immune to such repetition because they constrain a model to structure its response around the input tokens, thereby providing a guiding framework.

For PHEE, the performance relationship between standoff and token-level formats is reversed. Unlike with the aforementioned benchmarks, standoff formats achieve slightly higher performance on PHEE compared to token-level formats (Figure 2b). As shown in Table 1, PHEE is structurally simple: each sample contains on average only 1.0 annotations (the lowest across all benchmarks) with spans averaging just 1.1 tokens in length. In this setting, the main advantage of token-level formats—explicit, token-level predictions over the entire sequence and dense supervision over non-entity tokens—provides little benefit. Since the task effectively requires predicting only a single short span per sample, token-level formats are mildly detrimental for most models because the prediction is excessively granular, introducing noise without improving extraction accuracy.

For Qwen models in particular, the disparity is markedly sharper. Analysis of model outputs on PHEE shows that, when using token-level formats, Qwen models frequently predict spans that appear earlier in the sentence than the gold annotation. In contrast, predictions under standoff formats tend to align more closely with gold span positions. This pattern highlights another potential drawback of token-level formats, especially in settings like PHEE where annotations are sparse: because they rely on token-wise predictions, they can lead the model to overestimate the role of earlier tokens, resulting in premature span predictions.

Performance trends are less apparent in the SRL benchmarks. For NounAtlas, F1 scores remain generally consistent across the standoff and token-level formats (Figure 2f). Our analysis of model responses suggests that this is driven by the greater diversity in both sample lengths (Figure 4) and annotation lengths (Figure 3) in the NounAtlas dataset, compared to other benchmarks. This variability leads different formats to counterbalance each other's strengths and weaknesses. For example, we see that the token-level formats perform better on samples with numerous relevant spans, whereas the standoff formats show better performance on samples with fewer, longer spans. As a result, the performance gains across formats tend to average out.

For SpRL-2012, as with NounAtlas, overall F1 scores are similar across the standoff and token-level output formats (Figure 2e). The top-performing models—Mistral 7B, Llama 3.1 8B

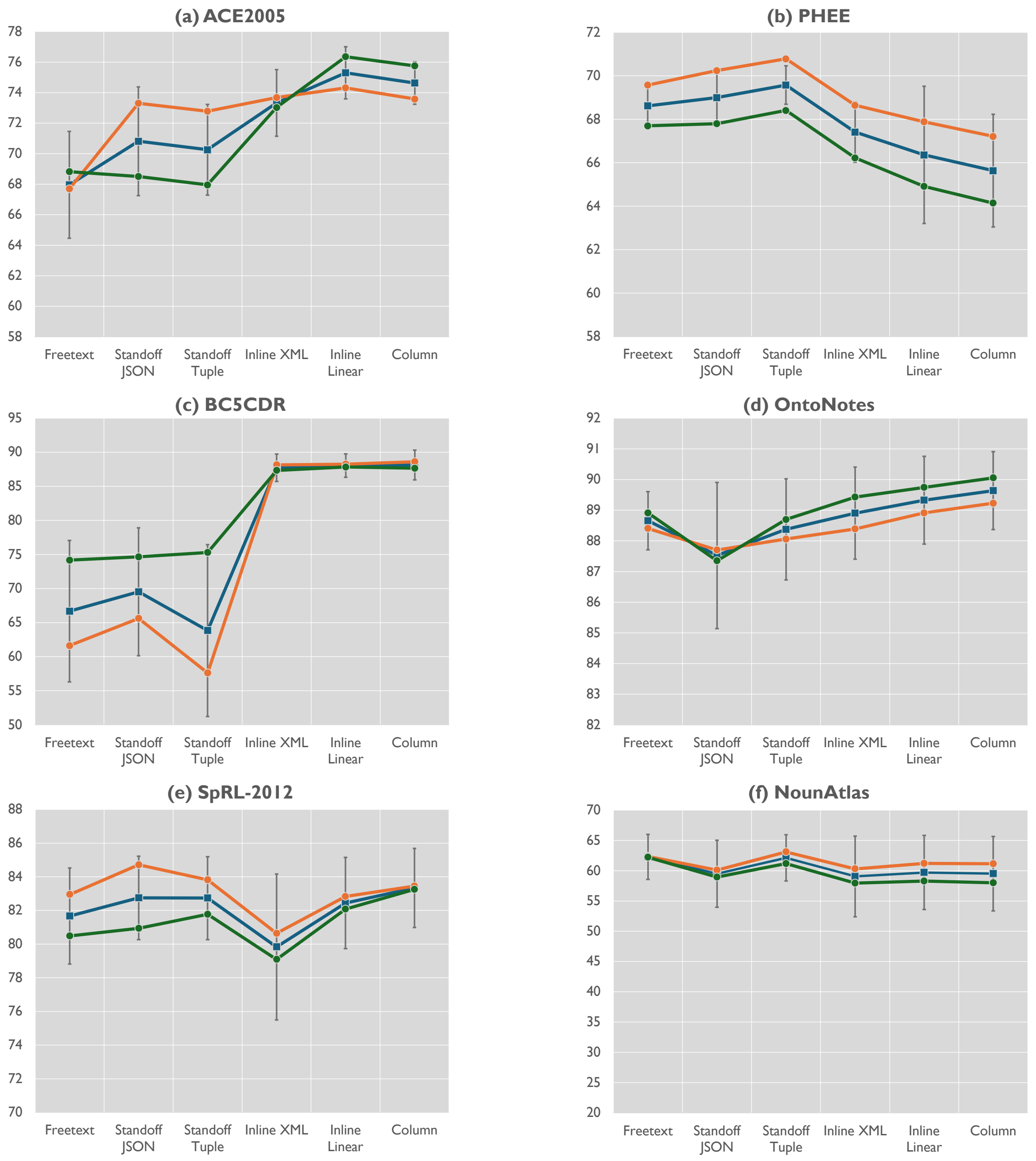


Figure 2: Mean F1■, P●, and R● scores for all output formats, averaged across models. Error bars indicate one standard deviation for F1.

and Mistral Nemo—show minimal differences in precision, recall and F1, demonstrating robustness to format choice. For other models, however, token-level formats achieve higher recall because they predict more spans on average (note: the dataset contains 4.5 annotations per sample on average), but slightly lower precision due to boundary misalignments. These opposing effects largely offset each other, resulting in similar F1 scores.

Inline XML is a midpoint between the standoff and token-level formats. Inline XML generally yields intermediate performance on the ED and NER benchmarks, positioning itself between the standoff and token-level formats. However, on SpRL-2012 and NounAtlas (where standoff and token-level formats perform comparably), Inline XML tends to underperform relative to both. Conceptually, Inline XML occupies an

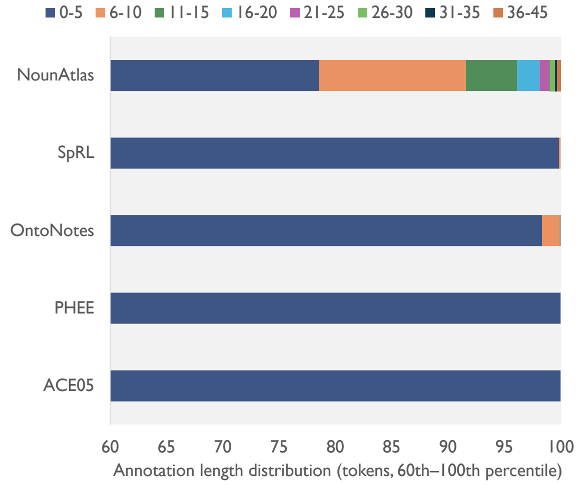


Figure 3: Distribution of gold-standard annotation lengths in the upper 40% of test examples (by token count). NounAtlas (top) displays the greatest spread in annotation lengths.

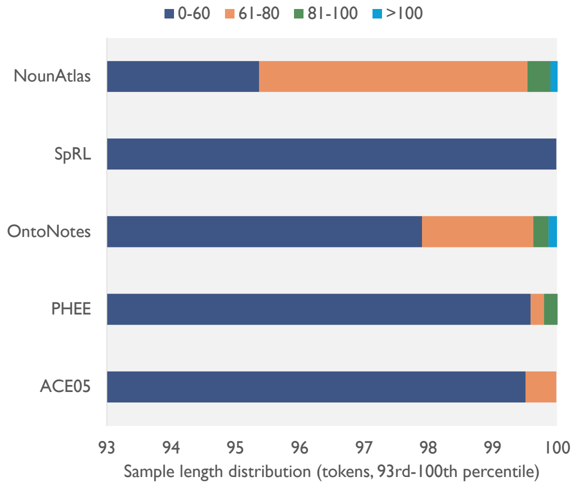


Figure 4: Distribution of test-set sample lengths by token count. Only the upper 7% of samples (by token count) are shown to emphasize the diversity in longer examples, though the diversity exists across the full dataset. NounAtlas (top) exhibits the greatest spread in sample length. BC5CDR (excluded) is an outlier due to its significantly higher average sample length.

intermediate position: it neither encourages a global view of the prediction task, as in standoff formats that operate over complete spans, nor enforces granular predictions, as in token-level formats. As a result, when the advantages of these two extremes balance out, the hybrid nature of Inline XML provides little benefit, leading to weaker performance in comparison.

It must be noted that for Mistral 7B, Llama 3.1 8B, Mistral Nemo and OLMo-2 32B, Inline XML frequently performs on par with the best formats, suggesting that certain models are better equipped to handle this format.

Freetext results in similar performance to standoff formats. He and Choi (2023) report that the Freetext format yields the best results for BART on three of four IE tasks, attributing this to its closer resemblance to natural language. In contrast, our experiments show that Freetext performs comparably to standoff formats across models and benchmarks. This suggests that modern LLMs are increasingly capable of handling structured representations directly, diminishing the need for natural-language–like output formulations.

Closely related formats yield effectively identical performance. All evaluated formats differ purely in syntax and are informationally equivalent. However, two format pairs in particular—Inline Linear vs. Column and Standoff JSON vs. Tuple—exhibit only trivial structural differences. This similarity is reflected in their performance: the average F1 difference between Inline Linear and Column is just 0.84% across all models and benchmarks, and between Standoff JSON and Tuple only 2.29%. These negligible gaps confirm that when syntactic variation is reduced to this level, it has no practical effect on performance.

Mistral models consistently deliver superior performance. We calculate the average rank of each model across all formats and benchmarks (Figure 5). Mistral 7B achieves the lowest average rank of 2.36, closely followed by Mistral Nemo (12B) at 2.64 (lower the rank, better the performance). The third lowest, and the largest model among those evaluated, is OLMo-2 32B, with a rank of 2.94. On excluding the Mistral models, we observe that *performance generally increases (rank decreases) with model size.*

Token-level formats are, on average, superior to other formats. Although no single output format consistently outperforms others, a crude average across models and datasets (Figure 6) reveals a clear overall tendency: token-level formats achieve the highest mean F1. This suggests that, on average, token-level formats yield stronger performance than alternative formats.

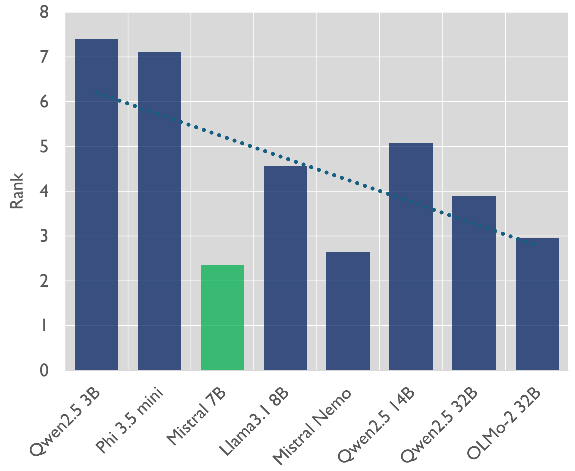


Figure 5: Average rank of evaluated models across all formats and benchmarks, where rank 1 indicates the highest performance in a metric.

However, the margin over other formats is small and does not hold uniformly across benchmarks or model families; performance differences vary considerably across settings.

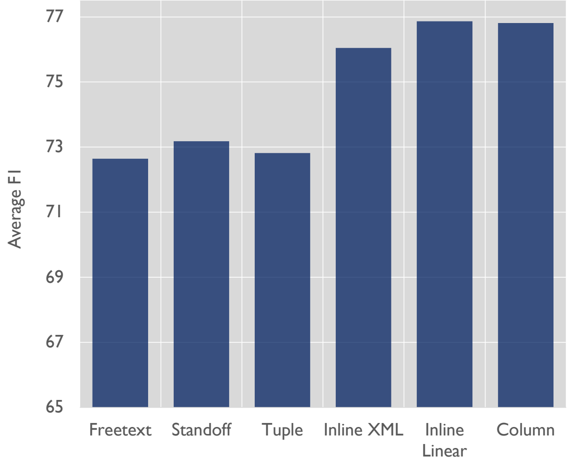


Figure 6: Average F1 of evaluated formats across all models and benchmarks.

Model robustness to output format increases with size. We quantify robustness as the average F1 difference between the best and worst performing output formats for each model across all benchmarks (Figure 7). A smaller average difference indicates that a model is less sensitive to output format choice. Overall, we observe that robustness tends to improve with model size, leading to more stable performance across formats. Model family also plays a critical role: for example, Mistral 7B displays unusually high robustness relative to its size, diverging from the size-based trend. These results emphasize that both model size and family should inform output format selection.

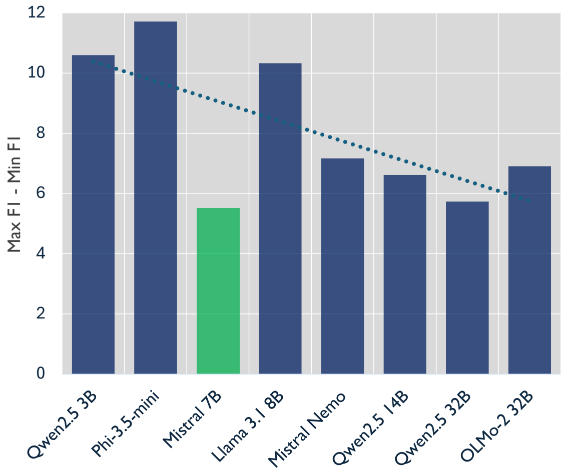


Figure 7: F1 gap between the best and worst performing output formats (‘Robustness’) for each model, averaged across benchmarks.

4.1 Guidelines for Choosing an Output Format

Our experiments show that output format substantially affects fine-tuning performance, and the optimal choice depends on dataset characteristics and the model. Output format should therefore be treated as a tunable hyperparameter and selected empirically for each application. Practical guidance from our results follows: (1) *High span density*: token-level formats tend to perform better when many spans occur per sample. (2) *Sparse annotation*: standoff or freetext formats generally yield stronger performance when annotations are sparse. (3) *Recall-oriented tasks*: token-level formats typically achieve higher recall. (4) *Long or contiguous spans*: standoff and freetext formats are less prone to boundary errors, which become more common in token-level formats as span length increases. (5) *Long documents*: token-level formats perform better on longer texts. (6) *Efficiency*: token-level, inline XML and freetext outputs are substantially longer, increasing training cost, inference latency, and pre/post-processing overhead; standoff JSON and tuple formats are significantly more compact and efficient. (7) *Character alignment*: token-level and inline XML formats preserve span offsets because labels are interspersed within the text, while standoff and freetext formats abstract this structure away.

Finally, model family and size also play an important role: larger models and certain model families exhibit greater robustness to format variation, so optimal performance is achieved by jointly tuning model and output format rather than optimizing them independently.

5 Conclusion

We present the first systematic investigation of how output formatting influences the performance of fine-tuned decoder-only LLMs on IE tasks. Through experiments spanning multiple models, benchmarks and output formats, we demonstrate that formatting choices alone can lead to substantial performance variation. Moreover, we show that no single format consistently dominates across settings, suggesting that output formatting should be treated as a tunable hyperparameter rather than a fixed design choice. Finally, based on our results, we provide practical guidelines for output format selection.

More broadly, our findings highlight the sensitivity of current models to surface-level syntactic variation, pointing to an important direction for future work: the development of models that are inherently more robust to superficial representational differences. Such robustness is crucial for building accurate and dependable IE systems.

Limitations

Our study focuses on publicly available models to ensure reproducibility and accessibility. While this restricts our analysis to smaller, open models, evaluating larger or closed-source models in future work could offer additional insights into output format sensitivity. To isolate the effects of formatting, we employ minimal prompting, though future studies might also investigate how different prompt designs interact with output formats. Finally, since our experiments are limited to non-nested span extraction, extending the analysis to nested or embedded spans could further broaden the applicability of our findings.

References

Hammaad Adam, Junjing Lin, Jianchang Lin, Hillary Keenan, Ashia Wilson, and Marzyeh Ghassemi. 2024. Clinical information extraction with large language models: A case study on organ procurement. *AMIA Annual Symposium Proceedings*, 2024:115–123.

Wasi Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian, and Kai-Wei Chang. 2021. Intent classification and slot filling for privacy policies. 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 4402–4417, Online. Association for Computational Linguistics.

AI2. 2025. Olmo 2 32b: First fully open model to outperform gpt 3.5 and gpt 4o mini. `https://allenai.org/blog/olmo2-32b`. Accessed: 2025-09-28.

Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 375–385, Online. Association for Computational Linguistics.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Angel Daza and Anette Frank. 2018. A sequence-to-sequence model for semantic role labeling. In *Proceedings of the Third Workshop on Representation Learning for NLP*, pages 207–216, Melbourne, Australia. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

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

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

David Dukić and Jan Snajder. 2024. Looking right is sometimes right: Investigating the capabilities of decoder-only LLMs for sequence labeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14168–14181, Bangkok, Thailand. Association for Computational Linguistics.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Aynur Guluzade, Naguib Heiba, Zeyd Boukhers, Florim Hamiti, Jahid Hasan Polash, Yehya Mohamad, and Carlos A. Velasco. 2025. *ELMTEX: Fine-Tuning LLMs for Structured Clinical Information Extraction. A Case Study on Clinical Reports*, page 181–185. Springer Nature Switzerland.

Biyang Guo, He Wang, Wenyilin Xiao, Hong Chen, ZhuXin Lee, Songqiao Han, and Hailiang Huang. 2024. Sample design engineering: An empirical study on designing better fine-tuning samples for information extraction with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 573–594, Miami, Florida, US. Association for Computational Linguistics.

Han He and Jinho D. Choi. 2023. Unleashing the true potential of sequence-to-sequence models for sequence tagging and structure parsing. *Transactions of the Association for Computational Linguistics*, 11:582–599.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance? *Preprint*, arXiv:2411.10541.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok, Thailand. Association for Computational Linguistics.

Nancy Ide, Christian Chiarcos, Manfred Stede, and Steve Cassidy. 2017. *Designing Annotation Schemes: From Model to Representation*, pages 73–111.

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.

Soo joon Choi and Ji jun Park. 2025. Harnessing generative llms for enhanced financial event entity extraction performance. *Preprint*, arXiv:2504.14633.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States. Association for Computational Linguistics.

Parisa Kordjamshidi, Steven Bethard, and Marie-Francine Moens. 2012. SemEval-2012 task 3: Spatial role labeling. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 365–373, Montréal, Canada. Association for Computational Linguistics.

LDC. 2005. Ace (automatic content extraction) english annotation guidelines for events. Accessed: 2025-02-12.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiangnan Li, Yice Zhang, Bin Liang, Kam-Fai Wong, and Ruifeng Xu. 2023. Set learning for generative information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13043–13052, Singapore. Association for Computational Linguistics.

Jiao Li, Yueping Sun, Robert J. Johnson, Dan Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan P. Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*.

Yuanye Liu, Jiahang Xu, Li Lyna Zhang, Qi Chen, Xuan Feng, Yang Chen, Zhongxin Guo, Yuqing Yang, and Peng Cheng. 2025. Beyond prompt content: Enhancing llm performance via content-format integrated prompt optimization. *Preprint*, arXiv:2502.04295.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

Yi-Long Lu, Chunhui Zhang, and Wei Wang. 2025. Systematic bias in large language models: Discrepant response patterns in binary vs. continuous judgment tasks. *Preprint*, arXiv:2504.19445.

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

Microsoft. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Mistral AI. 2024. Mistral nemo. Accessed: 2025-05-05.

Roberto Navigli, Marco Lo Pinto, Pasquale Silvestri, Dennis Rotondi, Simone Ciciliano, and Alessandro Scirè. 2024. NounAtlas: Filling the gap in nominal semantic role labeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16245–16258, Bangkok, Thailand. Association for Computational Linguistics.

NuMind. 2024. Nuextract: A foundation model for structured extraction. numind.ai/blog/nuextract-a-foundation-model-for-structured-extraction. Accessed: 2025-05-10.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *CoRR*, abs/2101.05779.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Karthik Raman, Iftekhar Naim, Jiecao Chen, Kazuma Hashimoto, Kiran Yalasangi, and Krishna Srinivasan. 2022. Transforming sequence tagging into a Seq2Seq task. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11856–11874, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C Comeau, Rezarta Islamaj, Aadit Kapoor, Xin Gao, and Zhiyong Lu. 2024. Opportunities and challenges for chatgpt and large language models in biomedicine and health. *Briefings in Bioinformatics*, 25(1):bbad493.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Asahi Ushio and Jose Camacho-Collados. 2021. T-NER: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62, Online. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. GPT-NER: Named entity recognition via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *Preprint*, arXiv:2304.08085.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes release 5.0. LDC2013T19, Linguistic Data Consortium.

Tongtong Wu, Fatemeh Shiri, Jingqi Kang, Guilin Qi, Gholamreza Haffari, and Yuan-Fang Li. 2023. KC-GEE: knowledge-based conditioning for generative event extraction. *World Wide Web*, 26(6):3983–3999.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2024. Yayi-uie: A chat-enhanced instruction tuning framework for universal information extraction. *Preprint*, arXiv:2312.15548.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

Huaiyuan Ying, Hongyi Yuan, Jinsen Lu, Zitian Qu, Yang Zhao, Zhengyun Zhao, Isaac Kohane, Tianxi Cai, and Sheng Yu. 2025. Genie: Generative note information extraction model for structuring ehr data. *Preprint*, arXiv:2501.18435.

Kaiwen Zhang, Feiyu Su, Yixiang Huang, Yanming Li, Fengqi Wu, and Yuhan Mao. 2024a. The Application of Fine-Tuning on Pretrained Language Model in Information Extraction for Fault Knowledge Graphs . In *2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 469–473, Los Alamitos, CA, USA. IEEE Computer Society.

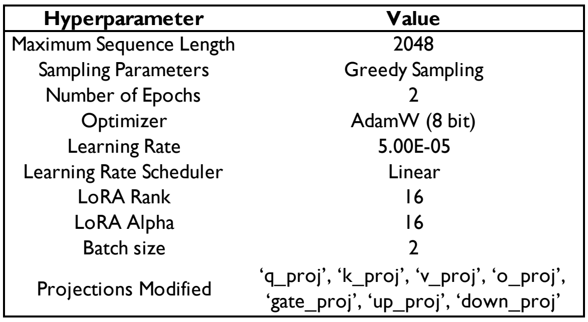
Wei Zhang, Qinggong Wang, Xiangtai Kong, Jiacheng Xiong, Shengkun Ni, Duanhua Cao, Buying Niu, Mingan Chen, Yameng Li, Runze Zhang, Yitian Wang, Lehan Zhang, Xutong Li, Zhaoping Xiong, Qian Shi, Ziming Huang, Zunyun Fu, and Mingyue Zheng. 2024b. Fine-tuning large language models for chemical text mining. *Chem. Sci.*, 15:10600–10611.

Xiang Zhang, Juntai Cao, Jiaqi Wei, Chenyu You, and Dujian Ding. 2025. Why prompt design matters and works: A complexity analysis of prompt search space in llms. *Preprint*, arXiv:2503.10084.

Appendix

A.1 Hardware and Hyperparameters

We fine-tune all models using QLoRA on NVIDIA RTX A5000 GPUs, with a cumulative training time of approximately 1400 GPU hours. Fine-tuning is performed using the Unsloth library. All models are 4-bit quantized at load time to reduce memory usage and accelerate training. The key hyperparameters used during fine-tuning are listed in Table 3.



Hyperparameter	Value
Maximum Sequence Length	2048
Sampling Parameters	Greedy Sampling
Number of Epochs	2
Optimizer	AdamW (8 bit)
Learning Rate	5.00E-05
Learning Rate Scheduler	Linear
LoRA Rank	16
LoRA Alpha	16
Batch size	2
Projections Modified	‘q_proj’, ‘k_proj’, ‘v_proj’, ‘o_proj’, ‘gate_proj’, ‘up_proj’, ‘down_proj’

Table 3: Hyperparameters used for fine-tuning

A.2 Corpus Annotation Schemes

Linguistic corpora are generally annotated in one of four ways (Ide et al., 2017):

- **Standoff**: Annotations are maintained in a separate file, rather than being embedded directly within the source text.

- **Inline XML**: Annotations are embedded directly into the document by enclosing *relevant text spans* within XML tags that contain annotation information.

- **Inline Linear**: As in Inline XML, annotations are embedded within the text; however, in this format, *every token* is surrounded by special characters that encode annotation information.

- **Column**: Each token is placed on a separate line along with its annotation.

A.3 Dataset Sourcing and Pre-Processing

A.3.1 Event Detection (ED)

ED is the task of identifying event triggers, i.e., spans of text that indicate the occurrence of events, and the corresponding event types (LDC, 2005). We use the following benchmarks:

- **ACE2005** (Doddington et al., 2004), the most widely known ED dataset, containing texts from a variety of genres such as newswire, broadcast news, weblog and conversational speech. It features 33 event types.

- **PHEE** (Sun et al., 2022), a dataset for pharmacovigilance. It defines two event types meant to indicate whether a drug-related event is an ‘Adverse event’ or a ‘Potential therapeutic event’. The data is sourced from medical case reports and biomedical literature.

Note that for ACE2005, we exclude samples that contain no events to ensure consistency with other benchmarks used in this study, such as PHEE, which do not include null examples. For both datasets, we follow the TextEE standardised benchmarking framework[3](Huang et al., 2024) to download, pre-process and split the data.

A.3.2 Named Entity Recognition (NER)

NER is the task of identifying and classifying entities in text (Grishman and Sundheim, 1996). We use the following benchmarks:

- **OntoNotes-5.0** (Weischedel et al., 2013), a widely used dataset containing texts from various genres including phone conversations, newswires, newsgroups, broadcast news, broadcast conversations and weblogs. It features 36 entity types.

- **BioCreative V CDR (BC5CDR)** (Li et al., 2016) a dataset of PubMed articles annotated for ‘Chemical’ and ‘Disease’ mentions, created to support tasks such as drug discovery.

For OntoNotes-5.0, we use the preprocessed version and data splits created for the T-NER python library (Ushio and Camacho-Collados, 2021), and made available on HuggingFace[4]. For consistency with other benchmarks, we exclude all samples that lack named entities.

For BC5CDR, we use the original dataset released on GitHub[5]. Since the dataset is split evenly across train, development and test sets, we reshuffle and re-split the data in a 7:1:2 ratio, to retain a sufficient number of samples in the test set.

In line with the CoNLL-2003 Shared Task guidelines (Tjong Kim Sang and De Meulder, 2003), we assume that named entities are non-overlapping and continuous. This assumption is consistent with OntoNotes-5.0. For BC5CDR, however, we preprocess the data to enforce these constraints by dropping discontinuous entities and retaining the longer entity when two entities overlap.

A.3.3 Semantic Role Labelling (SRL)

SRL involves identifying predicate-argument structures in a sentence and classifying the semantic role of each argument in relation to the predicate (Gildea and Jurafsky, 2000). We use the following benchmarks:

- **SemEval-2012 SpRL (SpRL-2012)** (Kordjamshidi et al., 2012), a corpus created to support the extraction of spatial arguments and relations in text. The benchmark features three spatial arguments: ‘trajector’, ‘landmark’ and ‘spatial-indicator’.

- **NounAtlas** (Navigli et al., 2024), a dataset for studying nominal predicates. It is derived from the SemCor corpus (Miller et al., 1993) and features 27 argument roles.

In the NounAtlas corpus, each sentence contains exactly one nominal predicate. Accordingly, we use both the sentence and the associated predicate (‘frame’) as input and fine-tune a model to predict the argument spans and their roles. We use the official dataset available on HuggingFace.[6]

For SpRL-2012, only the sentence is provided as input since the task involves predicting all spatial arguments without reference to a predicate or frame. We use the version of the dataset available on GitHub[7]. Since it includes only training and test splits, we merged, shuffled and re-split the data in a 7:1:2 ratio.

[3] github.com/ej0cl6/TextEE

[4] huggingface.co/datasets/tner/ontonotes5

[5] github.com/JHnlp/BioCreative-V-CDR-Corpus

[6] huggingface.co/datasets/sapienzanlp/nounatlas_srl_corpus

[7] github.com/kolomiyets/sprl2012

Model	***Freetext***	***Standoff JSON***	***Standoff Tuple***	***Inline XML***	***Inline Linear***	***Column***
Event Detection: ACE2005						
Qwen2.5 3B	62.68	66.73	65.67	68.64	72.07	72.12
Phi 3.5 mini	65.06	64.09	65.71	71.60	74.43	73.74
Mistral 7B	72.60	73.01	72.83	75.16	76.50	76.34
Llama3.1 8B	70.03	71.52	70.65	74.77	74.58	74.41
Mistral Nemo	70.22	74.84	70.64	74.37	77.26	74.62
Qwen2.5 14B	69.35	71.53	71.15	73.65	74.59	74.72
Qwen2.5 32B	64.19	72.29	73.01	73.85	76.32	76.47
OLMo-2 32B	69.56	72.50	72.42	74.55	76.66	74.62
Event Detection: PHEE						
Qwen2.5 3B	67.17	67.24	68.39	64.88	61.82	61.58
Phi 3.5 mini	68.73	67.38	68.29	66.26	66.42	65.42
Mistral 7B	69.64	68.62	68.96	68.67	69.54	68.70
Llama3.1 8B	68.35	69.57	70.28	68.77	69.50	66.36
Mistral Nemo	69.03	71.28	70.44	68.09	69.60	67.62
Qwen2.5 14B	67.80	69.73	70.10	66.46	62.73	62.43
Qwen2.5 32B	69.99	68.82	69.97	67.88	64.15	64.91
OLMo-2 32B	68.26	69.34	70.18	68.29	67.15	68.08
Named Entity Recognition: OntoNotes						
Qwen2.5 3B	87.28	84.00	87.34	86.55	87.36	87.57
Phi 3.5 mini	87.07	83.76	85.08	87.59	87.19	87.92
Mistral 7B	89.45	89.58	89.85	90.02	90.61	90.96
Llama3.1 8B	88.77	88.42	87.63	87.82	89.95	90.13
Mistral Nemo	89.47	89.18	89.79	90.00	90.44	90.49
Qwen2.5 14B	88.95	87.12	88.73	88.54	88.63	89.39
Qwen2.5 32B	89.15	88.61	88.74	89.87	89.69	89.92
OLMo-2 32B	89.12	89.52	89.84	90.85	90.75	90.73
Named Entity Recognition: BC5CDR						
Qwen2.5 3B	64.29	61.18	67.39	85.16	85.87	86.47
Phi 3.5 mini	49.42	53.01	45.83	84.46	85.85	83.48
Mistral 7B	72.82	78.28	67.36	89.78	86.34	90.21
Llama3.1 8B	55.48	65.96	45.00	88.17	89.72	88.95
Mistral Nemo	67.24	69.83	64.64	89.58	89.85	89.81
Qwen2.5 14B	76.29	77.34	75.24	87.37	88.84	88.52
Qwen2.5 32B	80.55	80.64	80.46	87.94	88.97	88.61
OLMo-2 32B	67.34	69.90	64.77	89.44	88.90	88.99
Semantic Role Labelling: SpRL-2012						
Qwen2.5 3B	77.89	79.59	80.19	73.25	79.04	82.07
Phi 3.5 mini	78.37	79.11	78.27	75.18	81.55	81.19
Mistral 7B	85.84	86.41	85.00	85.73	85.02	86.19
Llama3.1 8B	82.28	81.99	82.83	82.95	85.03	83.81
Mistral Nemo	83.17	84.10	84.96	83.50	85.91	86.24
Qwen2.5 14B	80.32	83.18	83.78	78.32	80.16	81.76
Qwen2.5 32B	80.77	82.87	82.03	78.07	79.65	80.24
OLMo-2 32B	84.73	84.73	84.83	81.66	83.19	85.22
Semantic Role Labelling: NounAtlas						
Qwen2.5 3B	57.10	50.51	56.47	47.47	50.21	49.37
Phi 3.5 mini	56.58	51.98	56.37	51.07	50.63	50.48
Mistral 7B	64.66	62.26	64.22	63.19	63.93	63.39
Llama3.1 8B	60.48	58.42	60.78	57.97	61.00	61.00
Mistral Nemo	65.38	62.96	65.65	63.50	63.21	63.22
Qwen2.5 14B	63.95	60.19	63.59	59.02	58.62	59.81
Qwen2.5 32B	65.42	64.50	65.24	64.27	64.52	63.76
OLMo-2 32B	64.70	65.11	64.61	65.95	65.51	65.09

Table 4: Summary of results (**F1**) grouped by benchmark.

Model	***Freetext***	***Standoff JSON***	***Standoff Tuple***	***Inline XML***	***Inline Linear***	***Column***
Event Detection: ACE2005						
Qwen2.5 3B	58.91	71.17	69.37	67.35	70.02	69.34
Phi 3.5 mini	67.82	66.41	71.07	73.68	73.91	74.68
Mistral 7B	74.21	74.35	74.31	75.91	76.10	75.61
Llama3.1 8B	73.82	74.19	72.09	76.44	74.91	73.69
Mistral Nemo	72.76	77.02	71.88	75.60	77.26	74.30
Qwen2.5 14B	65.61	74.42	75.00	72.38	72.41	72.19
Qwen2.5 32B	56.03	74.62	75.28	72.28	73.32	74.54
OLMo-2 32B	72.36	74.25	73.27	75.78	76.60	74.30
Event Detection: PHEE						
Qwen2.5 3B	67.68	68.56	69.66	65.78	62.64	62.46
Phi 3.5 mini	70.16	68.74	69.67	67.89	69.38	68.37
Mistral 7B	70.97	69.82	70.09	70.21	70.72	69.94
Llama3.1 8B	69.48	70.82	71.55	70.20	71.40	68.43
Mistral Nemo	70.24	72.46	71.56	69.66	71.38	68.98
Qwen2.5 14B	68.35	70.84	71.18	67.24	63.56	63.16
Qwen2.5 32B	70.13	69.92	71.12	68.67	64.71	65.85
OLMo-2 32B	69.60	70.78	71.44	69.52	69.32	70.51
Named Entity Recognition: OntoNotes						
Qwen2.5 3B	87.08	84.93	87.42	85.77	86.98	87.15
Phi 3.5 mini	87.23	83.28	83.29	87.66	86.76	87.51
Mistral 7B	88.78	89.59	89.74	89.10	90.24	90.50
Llama3.1 8B	88.64	88.39	87.46	87.54	89.56	89.68
Mistral Nemo	89.39	89.18	89.60	89.64	90.12	90.17
Qwen2.5 14B	88.52	87.71	88.75	87.80	88.04	88.96
Qwen2.5 32B	88.65	88.72	88.66	89.05	89.05	89.40
OLMo-2 32B	88.96	89.82	89.57	90.57	90.54	90.45
Named Entity Recognition: BC5CDR						
Qwen2.5 3B	64.17	59.15	69.19	86.03	86.07	86.87
Phi 3.5 mini	39.42	44.18	34.66	86.05	86.67	83.89
Mistral 7B	67.19	76.37	58.01	90.71	86.31	91.23
Llama3.1 8B	44.43	57.25	31.61	88.28	90.14	89.24
Mistral Nemo	59.10	63.58	54.61	89.92	90.05	90.43
Qwen2.5 14B	76.94	78.03	75.84	86.82	88.64	88.89
Qwen2.5 32B	81.13	81.14	81.12	87.85	88.80	88.78
OLMo-2 32B	60.56	65.27	55.85	89.53	89.36	89.57
Semantic Role Labelling: SpRL-2012						
Qwen2.5 3B	81.20	84.66	84.40	71.62	78.46	81.07
Phi 3.5 mini	79.08	80.48	77.48	77.13	82.24	81.93
Mistral 7B	86.06	86.30	85.27	87.20	86.16	87.86
Llama3.1 8B	82.02	82.47	82.36	83.94	86.49	84.69
Mistral Nemo	82.29	84.37	85.07	84.92	86.69	86.24
Qwen2.5 14B	83.01	87.41	86.90	77.17	80.00	80.61
Qwen2.5 32B	84.14	85.83	85.52	78.28	78.58	79.22
OLMo-2 32B	85.79	86.19	83.56	84.87	84.01	85.99
Semantic Role Labelling: NounAtlas						
Qwen2.5 3B	56.33	52.66	58.45	46.26	51.49	50.48
Phi 3.5 mini	57.02	51.52	57.50	54.43	53.11	53.34
Mistral 7B	65.04	61.86	64.68	65.49	65.67	65.10
Llama3.1 8B	60.93	58.14	61.21	60.25	62.85	63.07
Mistral Nemo	65.87	62.94	65.90	66.13	64.64	65.17
Qwen2.5 14B	63.48	62.49	65.35	58.08	59.40	60.65
Qwen2.5 32B	65.12	66.09	66.81	63.88	64.98	64.30
OLMo-2 32B	65.08	65.00	64.92	67.88	67.43	67.01

Table 5: Summary of results (**Precision**), grouped by benchmark.

Model	***Freetext***	***Standoff JSON***	***Standoff Tuple***	***Inline XML***	***Inline Linear***	***Column***
Event Detection: ACE2005						
Qwen2.5 3B	66.96	62.81	62.34	69.98	74.25	75.13
Phi 3.5 mini	62.52	61.92	61.10	69.63	74.96	72.82
Mistral 7B	71.05	71.71	71.40	74.42	76.91	77.09
Llama3.1 8B	66.61	69.04	69.27	73.18	74.25	75.13
Mistral Nemo	67.85	72.78	69.45	73.18	77.26	74.96
Qwen2.5 14B	73.53	68.86	67.67	74.96	76.91	77.44
Qwen2.5 32B	75.13	70.11	70.87	75.49	79.57	78.51
OLMo-2 32B	66.96	70.82	71.58	73.36	76.73	74.96
Event Detection: PHEE						
Qwen2.5 3B	66.67	65.97	67.16	64.01	61.02	60.72
Phi 3.5 mini	67.36	66.07	66.97	64.71	63.71	62.71
Mistral 7B	68.36	67.46	67.86	67.20	68.39	67.50
Llama3.1 8B	67.26	68.36	69.05	67.40	67.70	64.41
Mistral Nemo	67.86	70.15	69.35	66.60	67.90	66.30
Qwen2.5 14B	67.26	68.66	69.05	65.70	61.91	61.71
Qwen2.5 32B	69.85	67.76	68.86	67.10	63.61	64.01
OLMo-2 32B	66.97	67.96	68.96	67.10	65.10	65.80
Named Entity Recognition: OntoNotes						
Qwen2.5 3B	87.47	83.10	87.26	87.34	87.74	87.99
Phi 3.5 mini	86.91	84.25	86.94	87.53	87.62	88.34
Mistral 7B	90.12	89.57	89.96	90.97	90.97	91.43
Llama3.1 8B	88.90	88.44	87.80	88.10	90.34	90.58
Mistral Nemo	89.55	89.18	89.98	90.36	90.77	90.81
Qwen2.5 14B	89.39	86.54	88.71	89.30	89.22	89.83
Qwen2.5 32B	89.66	88.51	88.82	90.70	90.33	90.46
OLMo-2 32B	89.29	89.22	90.11	91.13	90.96	91.01
Named Entity Recognition: BC5CDR						
Qwen2.5 3B	64.40	63.36	65.67	84.31	85.68	86.08
Phi 3.5 mini	66.22	66.25	67.63	82.92	85.05	83.07
Mistral 7B	79.48	80.28	80.29	88.87	86.37	89.21
Llama3.1 8B	73.85	77.82	78.05	88.06	89.30	88.65
Mistral Nemo	77.98	77.44	79.18	89.25	89.64	89.20
Qwen2.5 14B	75.66	76.65	74.65	87.94	89.04	88.16
Qwen2.5 32B	79.98	80.15	79.80	88.04	89.15	88.45
OLMo-2 32B	75.82	75.25	77.09	89.35	88.45	88.41
Semantic Role Labelling: SpRL-2012						
Qwen2.5 3B	74.84	75.10	76.38	74.97	79.64	83.08
Phi 3.5 mini	77.66	77.79	79.08	73.31	80.88	80.47
Mistral 7B	85.62	86.52	84.72	84.32	83.91	84.59
Llama3.1 8B	82.54	81.51	83.31	81.98	83.63	82.94
Mistral Nemo	84.08	83.83	84.85	82.12	85.14	86.24
Qwen2.5 14B	77.79	79.33	80.87	79.50	80.33	82.94
Qwen2.5 32B	77.66	80.10	78.82	77.85	80.74	81.29
OLMo-2 32B	83.70	83.31	86.14	78.68	82.39	84.46
Semantic Role Labelling: NounAtlas						
Qwen2.5 3B	57.89	48.53	54.61	48.73	49.00	48.30
Phi 3.5 mini	56.15	52.45	55.28	48.10	48.37	47.91
Mistral 7B	64.28	62.66	63.75	61.05	62.29	61.76
Llama3.1 8B	60.03	58.70	60.35	55.85	59.25	59.06
Mistral Nemo	64.90	62.98	65.41	61.06	61.83	61.39
Qwen2.5 14B	64.42	58.05	61.93	59.99	57.86	58.99
Qwen2.5 32B	65.72	62.98	63.74	64.66	64.06	63.24
OLMo-2 32B	64.33	65.22	64.30	64.12	63.68	63.28

Table 6: Summary of results (**Recall**), grouped by benchmark.