Format Matters: A Critical Evaluation of Output Formats for Prompting LLMs in SLU and NER

Pierre Lepagnol[1,2], Sahar Ghannay[1], Thomas Gerald[1], Christophe Servan[1,3], Sophie Rosset[1]
[1]Université Paris-Saclay, CNRS, LISN, France
[2]SCIAM, France
firstname.lastname@lisn.fr

[3]AMIAD, Pôle Recherche, France
firstname.lastname@polytechnique.edu

Abstract

Output format is often an unreported factor in LLM evaluations for structured NLP tasks such as Slot Filling or Named Entity Recognition. This work proposes to explore the impact of the output structured format generated by LLMs. We show that measured performance and reliability depend on the requested format (JSON, XML or inline Key-Values). A study is performed across four SLU and three NER benchmarks, considering 13 instruction-tuned open-weight LLMs, using standardized and open-source prompts and parsers. This format-specific evaluation reveals statistically significant swings of 2–46 F_1points, depending on model and dataset. Additionally, we propose a lightweight selection procedure to determine the best format per model-dataset combination using only a small development slice; thus reducing trial-and-error in practice.

Keywords: evaluation, prompting, output formats, SLU, NER, large language models

1. Introduction

Why does the output format matter when prompting LLMs for structured understanding tasks? In Spoken Language Understanding (SLU) and Named Entity Recognition (NER) systems, the output must be machine-readable (e.g. slot lists or entity spans). The same prediction can be expressed in multiple ways, such as key-value lines, JSON, XML or inline tags, and these choices affect both the model output and the evaluation protocol.

Figure 1 illustrates these formats. Although they appear equivalent to humans, each triggers distinct failure modes (e.g. malformed braces, tag mismatches); Tam et al. (2024) show that even small syntactic differences cause parser failures, and token overhead or nesting further affects smaller models. Consequently, results reported with different output formats are often not comparable unless formatting and parsing are explicitly controlled. Also, the growing interest of grammar-constrained decoding (Geng et al., 2023) and structured runtimes (Zheng et al., 2024) further motivates measuring structure validity alongside task accuracy.

Despite rapid progress in prompting for structured NLP tasks such as slot filling (He and Garner, 2023; Mirza et al., 2024; Zhu et al., 2024; Qin et al., 2024; Wang et al., 2023; Lepagnol et al., 2025) or NER (Wang et al., 2025; Xie et al., 2024), most studies fix a single output format, limiting cross-paper comparability and obscuring format sensitivity. For NER, alternative outputs (e.g. IOB tags, inline tags, JSON lists) introduce different parser sensitivities and metric choices, yet direct format comparisons remain rare.

Our study addresses this issue by treating output format as a primary hyperparameter rather than a presentation detail. We present a format-specific review and protocol that encompass 13 models, 4 slot-filling benchmarks and 3 NER tasks.

This work is structured around two central research questions:

- RQ1: To what extent do output formats affect measured performance and reliability?
- RQ2: How can we select a suitable format without resorting to exhaustive trial and error?

Section 2 reviews related work. Sections 3–4 describe our methodology and experimental setup, Section 5 reports results, and Section 6 concludes.

2. Related Work

We review structured NLP tasks (SLU and NER), instruction-tuned LLMs, and discrepancies in format comparisons.

2.1. Structured NLP tasks: SLU and NER

Within structured prediction, SLU approaches target intent detection and slot filling in goal-oriented systems, while NER identifies and labels spans.

2.1.1. SLU

Canonical SLU formulations decompose meaning into *intent* and *slots* (Qin et al., 2021b). Early joint

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 7485–7497
11–16 May 2026.

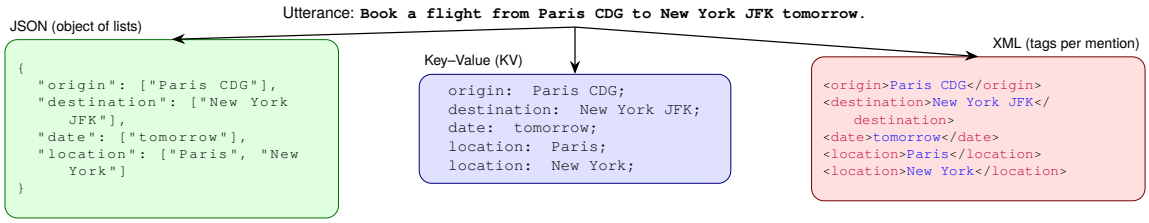

Figure 1: Data Format Comparison: Key-Value, JSON, and XML Representations of Location and Date Information

models exploited *implicit* inter-task signals via parameter sharing (e.g., RNNs/BiRNNs with attention) (Zhang and Wang, 2016; Hakkani-Tür et al., 2016). Later work imposed *explicit* interactions, first with pipelines where intent guides slots (Goo et al., 2018; Qin et al., 2019), then bidirectional designs exchanging signals during inference (Wang et al., 2018; E et al., 2019; Zhang et al., 2019a). Transformer-based co-interactive frameworks further systematize this bidirectional flow (Servan et al., 2006; Alavoine et al., 2024; Qin et al., 2021a). Despite these advances, most approaches assume substantial labeled data.

2.1.2. NER

NER classically appears as sequence labeling. Early neural systems combined recurrent encoders with CRFs (Hammerton, 2003; Collobert et al., 2011; Lample et al., 2016), while transformer encoders with task-specific fine-tuning later became dominant (Devlin et al., 2019).

2.2. Instruction-tuned LLMs

Instruction tuning (Wei et al., 2022a) addresses cross-task generalization by fine-tuning language models with task instructions and desired outputs (Zhang et al., 2023). Aligning next-word prediction with user intent increases controllability and reliability, leading to models such as InstructGPT (Ouyang et al., 2022) and FLAN-T5 (Wei et al., 2022a), which often outperform their base counterparts. Instruction datasets are commonly built by templating text-label pairs into instruction-output pairs (Muennighoff et al., 2023; Longpre et al., 2023). These models are chosen because they unify diverse structured NLP tasks under a single prompting interface.

2.2.1. SLU/NER via instruction.

Recently, generalist models trained on diverse mixtures target zero-shot transfer (Zaratiana et al., 2024). In-context and instruction-following paradigms enable prompting without further training (Brown et al., 2020; Liu et al., 2023). Concurrently, some work optimizes prompts continuously (prompt tuning) (Ma et al., 2022; Layegh et al., 2023; Hu et al., 2023), while others explore prompt formulations and structures (Wei et al., 2022b; Ashok and Lipton, 2023; Wang et al., 2025; Naguib et al., 2024; Villena et al., 2024; Cocchieri et al., 2025). Nevertheless, the community lacks a standardized, widely adopted NER prompting protocol, complicating fair comparisons across rapidly evolving LLMs.

For SLU specifically, LLM prompts have been used to jointly address intent and slots (Pan et al., 2023; He and Garner, 2023; Mirza et al., 2024), including studies of inter-task interaction with ChatGPT (Zhu et al., 2024) and cross-task prompting that makes information exchange explicit across models (Qin et al., 2024). Orthogonally, information retrieval has been used to enrich prompts (primarily for slot filling) in few-shot regimes (Lepagnol et al., 2025).

2.3. Discrepancies in Format Comparisons

A recurring confounder in LLM-based systems is output *format*. Generations vary from natural language to JSON objects to IOB-style tags, and can materially affect reported F_1. Broadly, prompting strategies split into two families: *Tagging* prompts insert markers around mentions (Naguib et al., 2024; Wang et al., 2025); *Listing* prompts return entities as a structured list (Ashok and Lipton, 2023; Mirza et al., 2024; Zhu et al., 2024; Lepagnol et al., 2025; Cocchieri et al., 2025; Villena et al., 2024). Tagging prompts require as many inferences as the number of labels, while listing prompts require only a single inference for all labels. Therefore, we focus on listing prompts in our study.

There are several implementations in the listing-prompt literature: concise inline lists (Ashok and Lipton, 2023), JSON/XML serializations (Mirza et al., 2024; Villena et al., 2024), key-value tables (Lepagnol et al., 2025), natural-language statements (Zhu et al., 2024), or (span, label) tuples (Cocchieri et al., 2025). Most studies fix one representation and parser, without comparing format sensitivity; this motivates our format-specific evaluation protocol. We therefore want to disentan-

gle modeling effects from formatting and parsing effects.

2.4. Position

Our work addresses this blind spot by elevating output format to a first-class experimental factor, providing evidence of its impact and practical selection methods for fair and reproducible SLU/NER evaluation.

3. Methodology

This section presents our evaluation protocol, parsers, and our selection procedure to quantify how output formats affect measured performance and to choose a reliable format with minimal assumptions and auditable steps.

3.1. Task Formulation

We study two similar sequence labeling tasks, single-turn slot-filling (SF) for SLU and Named Entity Recognition (NER). Figure 2 illustrates this task formulation. The same formalization applies to both tasks: Given a sentence x and a list of labels $\mathcal{Y}$, the system outputs a set $\{(y_i, \tilde{x}_i)\}$ where $y_i \in \mathcal{Y}$ and $\tilde{x}$ is a part of x.

Exemple :

Utterance: "Book a flight from Paris CDG to New York JFK tomorrow"

Slots: {origin, destination, date, location}

Output: {(origin, "Paris CDG"), (destination, "New York JFK"), (date, "tomorrow"), (location, "Paris"), (location, "New York")}

Figure 2: Example of a sequence labeling task

We use the macro F_1score as the evaluation metric, with exact span+label matching (case insensitive) as outlined in Section 4.

3.2. Output Formats Comparison (RQ1)

We provide a detailed overview of the formats we compare, how we convert generations into slot predictions, and how we statistically assess differences.

3.2.1. Formats Compared

We evaluate three widely used formats: JSON, XML, and Key–Value (KV). We use flat schemas and allow repeated labels and multiple mentions, to capture nested annotations. Figure 1 illustrates the differences between the formats in a concrete example.

3.2.2. Format-specific Parsing

Raw generations are converted into predictions using format-specific regular expressions. During parsing, we remove any duplicates or repetitions (i.e. the same span with the same label) from the prediction. This distinction clarifies whether failures occur in terms of syntax or task performance.

3.2.3. Statistical Significance Testing

We assess whether observed F_1differences across formats are statistically significant with a percentile bootstrap on paired differences (Keller et al., 2005) conducted independently for each model–dataset combination.

For every pair of formats $(\text{Format}_1, \text{Format}_2)$ we draw 3,000 bootstrap samples (with replacement) of the evaluation, then we compute the difference $F_1^{\text{format}_1} - F_1^{\text{format}_2}$ on each sample, and form a 95% confidence interval from the empirical quantiles of the resulting distribution.

3.3. Format Selection (RQ2)

We investigate whether it is possible to automatically select the best format for a given model-dataset pair using only a small development set. Our approach consists in selecting the format that scores the best on the development set.

The evaluation is based on the difference in F_1score between the *selected format* (`JSON`, `KV`, `XML`) and the *oracle* format on the test set (more details in Section 5.2).

4. Experimental Material

This section provides an overview of the experimental material used, including prompts, generation settings, datasets and models.

All implementation details, including retrieval configuration, prompts, decoding settings and parsers, are available to ensure reproducibility[1].

4.1. Prompt Design & Generation settings

Figure 3 illustrates an example of the `KV` prompt format for the Slot Filling task, structured into four components.

We choose to include 10 examples in the prompt according to the current state of the art (Mirza et al., 2024; Lepagnol et al., 2025; Naguib et al., 2024; Wang et al., 2025).

As proposed by Lepagnol et al. (2025), examples were selected using BM25 similarity scoring (Robertson and Zaragoza, 2009), which retrieves contextually relevant examples from the train

[1] https://gitlab.lisn.upsaclay.fr/phd-pierre-lepagnol/format-constrained-for-slu.git

split. We generate 512 tokens using greedy decoding to ensure reproducibility.

Constrained decoding generation (CDG) enforces valid output structure by restricting the model's vocabulary to tokens permitted by a formal grammar at each decoding step. However, Beurer-Kellner et al. (2024) show that CDG often impairs performance due to misalignment between learned token distributions and grammar constraints, forcing suboptimal token selection. Given these limitations, we perform unconstrained decoding in our experiments.

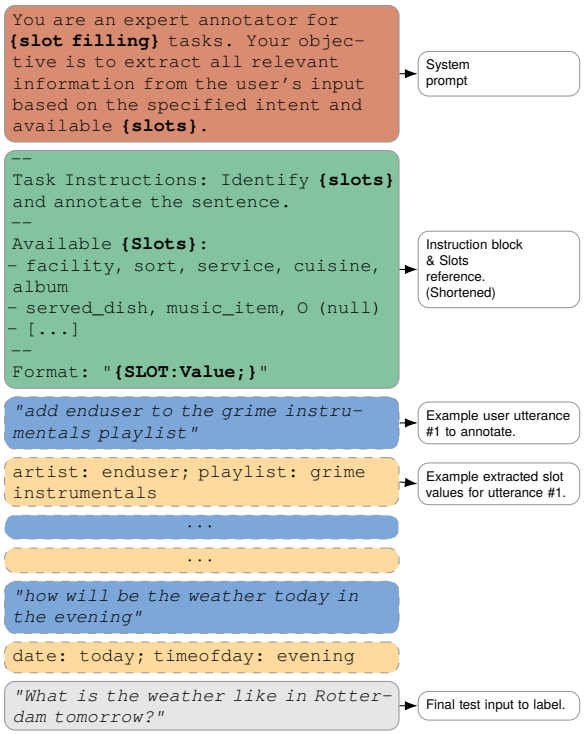

Figure 3: Example of prompt template with Key-Value format for Slot Filling task: In red, system prompt. In green are the task instructions with allowed slots. In blue, examples with labeled slots. In gray is the current utterance we want to label. In **bold**, specific part depending on task (SF/NER)

4.2. Datasets & Evaluation Metrics

Our work uses four well-established SLU and three NER datasets. For each dataset, we report the number of sentences in the training, development, and test splits, as well as the number of label types.

- **ATIS** is an English dataset with queries about airline travel information (Hemphill et al., 1990). It contains 4,978 training, 500 development, and 893 test sentences, and the number of label types is 79 in training, 67 in development, and 69 in test.
- **SNIPS** is an English dataset from the SNIPS personal assistant (Coucke et al., 2018). It contains 13,084 training, 500 development, and 700 test sentences, with 39 label types in all splits.
- **SLURP** is an English dataset simulating single-turn interactions between users and a voice-controlled assistant (Bastianelli et al., 2020). It contains 11,514 training, 2,033 development, and 2,974 test sentences, with 55 label types in training and 53 in development and test.
- **MEDIA** is a French dataset about hotel reservations and information (Bonneau-Maynard et al., 2005; Alavoine et al., 2024). It contains 13,712 training, 1,367 development, and 3,767 test sentences, and the number of label types is 144 in training, 104 in development, and 125 in test. We used the 2022 relaxed scoring version (Laperrière et al., 2022), which simplifies attributes by excluding specifiers, enriched with intent labels from Alavoine et al. (2024).

For NER, we rely on three complementary datasets:

- **CoNLL-2003** (Tjong Kim Sang and De Meulder, 2003) is a manually-annotated multilingual NER dataset released as part of the CoNLL shared tasks. It contains 14,986 training, 3,465 development, and 3,683 test sentences, and four entity types. Mentions of persons, locations, organizations, and miscellaneous entities are annotated. We use the English data of the 2003 version, which consists of Reuters news stories between 1996 and 1997.
- **WikiNER** (Nothman et al., 2013) is a multilingual, silver-standard NER dataset. It contains 129,376 training, 500 development, and 14,398 test sentences, and three entity types. It consists of a 2010 snapshot of Wikipedia in nine languages. Hyperlinks referring to persons, locations or organizations were automatically annotated. We use the English version of this dataset.
- **E3C** (Magnini et al., 2021) is a European multilingual corpus (Italian, English, French, Spanish, and Basque) of semantically annotated clinical narratives. Texts are collected from multiple publicly-available sources such as abstracts extracted from CC-licensed journals. We use the gold standard material available from the English versions of this dataset. It contains 100 training, 101 development, and 312 test sentences, and six entity types: actors, body parts, events, RMLs (measurements and test results), and clinical entities.

4.3. Models

Experiments were conducted on open-weight models from 4 families: `Llama models`, `SmolLM2`

Task	ATIS			SNIPS			SLURP			MEDIA		
Model Name	JSON	KV	XML	JSON	KV	XML	JSON	KV	XML	JSON	KV	XML
Llama-3.1-8B-Instruct	**91.45**	91.40	91.31	*__89.21__	88.21	87.55	*59.12	*__64.27__	*55.76	62.32	*__64.32__	62.21
Llama-3.2-1B-Instruct	79.80	**80.13**	*77.81	**74.34**	71.56	73.01	53.51	*__58.54__	53.85	54.98	**55.07**	54.41
Llama-3.2-3B-Instruct	**87.23**	86.37	86.44	83.90	84.03	**84.31**	*50.36	*__55.65__	*51.85	*57.43	*__63.09__	*58.95
Qwen2.5-0.5B-Instruct	76.42	**76.68**	74.83	**69.95**	69.00	68.04	50.04	*__51.57__	49.53	*55.52	*__56.66__	*53.26
Qwen2.5-1.5B-Instruct	85.75	**86.52**	86.00	81.71	**82.32**	82.03	*52.42	*__58.06__	*50.80	59.32	**59.56**	59.02
Qwen2.5-3B-Instruct	**90.38**	89.64	88.91	**84.51**	83.41	84.30	*56.95	*60.37	*__61.47__	61.36	**61.47**	61.15
Qwen2.5-7B-Instruct	90.84	**90.91**	90.68	**88.42**	87.55	86.82	62.28	*__66.52__	61.37	**62.46**	62.26	*61.46
Phi-3-medium-128k-instruct	**89.28**	89.22	88.93	**86.82**	85.62	86.40	63.57	*__65.59__	63.12	*61.55	*__63.45__	*62.61
Phi-3-mini-128k-instruct	87.47	87.36	**87.54**	83.72	**83.87**	83.65	58.80	*__61.01__	58.54	*55.61	*__65.01__	*18.11
Phi-4-mini-instruct	**88.81**	87.77	87.55	84.12	83.83	**84.55**	51.18	*__54.89__	51.50	61.27	*__62.28__	60.53
SmolLM2-1.7B-Instruct	84.31	**84.32**	*82.30	**80.25**	79.60	80.07	58.33	**60.75**	59.54	*59.99	*61.15	*__62.82__
SmolLM2-135M-Instruct	69.52	70.58	**70.70**	63.65	*__65.78__	62.53	**53.81**	52.58	53.53	**49.01**	*35.67	48.86
SmolLM2-360M-Instruct	75.73	**75.83**	75.53	71.80	**72.41**	71.89	*54.72	56.43	**56.83**	53.29	53.94	*__54.97__

Table 1: F_1 scores for SLU tasks. Models are grouped by families and sorted by size. Best performance per format is in **bold**. * marks statistically significant scores over other formats for each task.

Task	CONLL2003			WIKINER			E3C		
Model Name	JSON	KV	XML	JSON	KV	XML	JSON	KV	XML
Llama-3.1-8B-Instruct	*67.09	68.34	**68.62**	*72.99	*__75.87__	*74.00	51.76	**55.38**	53.57
Llama-3.2-1B-Instruct	*51.48	*__55.77__	*48.21	*48.53	*__55.91__	*43.13	37.14	**38.87**	*28.60
Llama-3.2-3B-Instruct	*58.57	**60.77**	59.88	*68.36	**70.86**	70.56	46.93	**48.55**	*38.23
Qwen2.5-0.5B-Instruct	*49.79	**53.84**	53.55	*52.66	*__57.46__	*56.09	33.40	**35.79**	*28.62
Qwen2.5-1.5B-Instruct	59.10	*__61.00__	58.48	61.00	*__62.61__	60.50	*__40.37__	35.33	36.47
Qwen2.5-3B-Instruct	*65.14	**68.04**	67.75	*67.57	*__70.77__	*68.58	50.61	*__53.41__	48.60
Qwen2.5-7B-Instruct	*71.89	*__76.18__	*74.61	*72.33	*__75.19__	*73.63	48.50	*__54.13__	49.55
Phi-3-medium-128k-instruct	*__64.31__	*61.60	*63.22	*68.19	*65.35	*__70.08__	**52.49**	50.98	49.68
Phi-3-mini-128k-instruct	*58.57	59.88	**60.26**	64.73	64.59	*__66.03__	*__51.18__	*47.63	*44.93
Phi-4-mini-instruct	*__56.95__	54.33	55.40	65.56	*__66.59__	65.83	**41.35**	39.59	40.01
SmolLM2-1.7B-Instruct	**54.80**	54.28	54.58	57.35	*54.22	**57.77**	35.28	**36.25**	34.40
SmolLM2-135M-Instruct	43.00	**45.08**	44.16	*45.33	*__49.22__	*48.48	23.47	*__29.35__	24.05
SmolLM2-360M-Instruct	47.91	47.73	**49.08**	49.36	**49.59**	48.86	*21.83	24.42	**24.46**

Table 2: F_1 scores for NER tasks. Models are grouped by families and sorted by size. Best performance per format is in **bold**. * marks statistically significant scores over other formats for each task.

`models`, `Phi models`, and `Qwen models`.

Models were selected to provide comparison across different sizes (150M to 14B parameters) and training data mixtures, ensuring reproducibility through open weights.

Experiments were conducted on a single NVIDIA A100 GPU using vLLM (Kwon et al., 2023) for efficiency.

5. Results

We now report how output formats shape measured performance and reliability and how practitioners can identify a suitable format without exhaustive trials, based on the protocol outlined in Section 3 and the material in Section 4.

RQ1: To what extent do output formats affect measured performance and reliability? Subsection 5.1 quantifies the magnitude of format sensitivity across 13 instruction-tuned LLMs $\times$ 4 SLU and 3 NER benchmarks, highlighting statistically significant performance gaps.

RQ2: How can we select a suitable format without resorting to exhaustive trial and error? Subsection 5.2 evaluates data-driven selection strategies against an oracle upper bound, showing that lightweight procedures can recover near-optimal formats with limited development data.

5.1. Format Performance Impact Analysis (RQ1)

Table 1 and Table 2 present the results obtained for the SLU tasks and the NER tasks, respectively. Statistical significance between formats is indicated using a star (*) beside the score; when all three scores have stars for a task, this means that each format differs significantly from the others in the considered task.

Across the 156 model-dataset-formats combinations (13 models $\times$ 3 formats $\times$ 4 SLU datasets) reported in Table 1 and the 117 combinations (13 models $\times$ 3 formats $\times$ 3 NER datasets) reported in Table 2, one can observe that the choice of out-

put format has a measurable impact on F_1 scores. Following the protocol described in Section 3.2.3, pair-wise tests found significant format differences in 65 of 156 SLU combinations and 78 out of 117 NER combinations.

We conduct our analysis at two levels of sensitivity: the dataset level and the model level. The results of these analyses are presented in the following subsections.

5.1.1. Dataset-Dependent Format Sensitivity

From a dataset-level perspective, we assess the sensitivity of each dataset to the output format. We report in Table 3 for each dataset the spread of F_1 between best and second-best format, and between best and worst format, averaged across all evaluated models.

On one hand, we identify two datasets that seem insensitive to the output format: ATIS and SNIPS. These datasets can be described as format-agnostic. ATIS and SNIPS show very few significant pairwise spreads in Table 1. Only 7 pairwise format differences are significant for the ATIS task, and 6 for the SNIPS task. Their average best–worst spread is $\simeq$1.0 F_1 (ATIS) and $\simeq$1.28 F_1 (SNIPS).

On the other hand, SLURP, and MEDIA, exhibit many more significant pairwise differences and, in several cases, larger spreads. On SLURP, 27 pairwise differences are significant, albeit with a best–worst spread of 4.02 F_1 (indicating consistent but moderate format effects). On MEDIA, 25 are significant with a much larger best–worst spread of 6.31 F_1 (with a maximum of 46.90 F_1 for a single model–dataset pair), pointing to stronger format sensitivity.

For the NER tasks, results presented in Table 2 show that all three datasets are format sensitive. WikiNER shows the highest sensitivity with 32 (over 39) pairwise differences that are significant with a best–worst spread of 2.77 F_1. For CoNLL-2003, 25 pairs are significant (64.1%) with a best–worst spread of 3.58 F_1. Finally, for E3C, 21 pairwise differences are significant with a best–worst spread of 5.23 F_1.

SLU Datasets:	ATIS	SNIPS	SLURP	MEDIA
F_1^{best}-F_1^{second}	0.32	0.67	2.75	1.69
F_1^{best}-F_1^{worst}	1.00	1.28	4.02	6.31
NER Datasets:	**CONLL2003**	**WIKINER**	**E3C**	
F_1^{best}-F_1^{second}	1.14	1.66	2.43	
F_1^{best}-F_1^{worst}	2.77	3.58	5.23	

Table 3: Performance spreads between best, second-best, and worst performing format across datasets, averaged across models. Values show F1 score differences with higher values indicating greater performance variability between formats.

This dataset-level format sensitivity observed across all these tasks may be explained by two main factors. First, it is possible that ATIS and SNIPS, being widely used benchmark datasets, have partially leaked into the pre-training data of the evaluated models, which could make them less sensitive to formatting variations. Second, these datasets are inherently simpler, featuring fewer labels, and utterances that are generally closer to the token distributions seen during model training.

Consequently, while ATIS and SNIPS appear largely format-agnostic, more complex datasets such as SLURP, MEDIA, and especially WikiNER exhibit pronounced performance variability across formats, necessitating more careful format selection.

5.1.2. Model-Level Format Sensitivity

Model	F_1^{best}-$F_1^{\text{sec.}}$	F_1^{best}-F_1^{worst}
SmolLM2-360M-Instruct	0.50	1.35
Phi-4-mini-instruct	1.36	1.84
SmolLM2-1.7B-Instruct	0.67	1.98
Phi-3-medium-128k-instruct	1.12	2.31
Qwen2.5-3B-Instruct	1.06	2.62
Qwen2.5-1.5B-Instruct	2.02	2.69
Llama-3.1-8B-Instruct	1.74	2.92
Qwen2.5-7B-Instruct	1.87	2.97
Qwen2.5-0.5B-Instruct	1.13	3.60
Llama-3.2-3B-Instruct	1.69	3.89
SmolLM2-135M-Instruct	1.38	4.41
Llama-3.2-1B-Instruct	2.84	5.91
Phi-3-mini-128k-instruct	2.44	8.45

Table 4: Performance spreads between best and second-best format, and between best and worst format averaged across all evaluated datasets. Values represent F1 score differences, with higher values indicating greater performance variability between models.

Complementing the dataset-level view, we now analyze model-level format sensitivity. Table 4 displays the spread in F_1 model-wise between best and second-best format, and between best and worst format averaged across all evaluated datasets.

To understand how different models respond to output format, we aggregate format sensitivity across all seven datasets. The 13 models span a wide sensitivity spectrum, from robust (minimal format impact) to highly sensitive (up to 40-point degradation).

We identify three distinct groups based on their format sensitivity.

The most robust group—SmolLM2-360M, SmolLM2-1.7B, Phi-4-mini, and Phi-3-medium—exhibits best–worst spreads below 2.3 points (Table 4).

The second group shows modest sensitivity to format choice. Qwen2.5-0.5B/3B/7B, Llama-3.1-8B-Instruct, and Llama-3.2-3B-Instruct fall into this category, exhibiting average best–sec. spreads around 1.1–2 points and best–worst gaps around 2.3–3.9 points.

Finally, the third group is the one that shows the highest sensitivity. Phi-3-mini-128k-instruct shows the most extreme sensitivity with the largest best-worst spread ($\simeq$8.45) and a dramatic per-dataset swing of $\simeq$46.90 points on MEDIA (XML 18.11 vs. KV 65.01). Llama-3.2-1B-Instruct exhibits similarly high sensitivity (best–sec. $\simeq$2.84; best–worst $\simeq$5.91), with a per-dataset swing of $\simeq$12.78 on WikiNER. SmolLM2-135M-Instruct, despite having a moderate best–worst spread ($\simeq$4.41), can still swing up to $\simeq$13.34 points on MEDIA.

Format robustness does not simply scale with parameter count—the 360M SmolLM2 outperforms several 1B+ models—suggesting that training mixture matters more than size. For practitioners, robust models can safely rely on a single format choice (often KV), while highly sensitive models require careful, dataset-specific format selection to avoid substantial performance degradation.

5.1.3. Format Sensitivity Preferences

Building on the sensitivity analysis, clear format preferences emerge across datasets. On SLURP, KV dominates, winning 10 out of 13 model comparisons. Here, KV scores better than alternatives with an average $+3.39$ F_1 advantage. WikiNER shows the same trend: KV wins 10 out of 13 comparisons with a $+1.70$ F_1 mean margin. On MEDIA, KV wins 9 over 13 times with a $+2.11$ F_1 advantage on average, whereas XML and JSON each lead in only 2 of 13 comparisons. On CoNLL-2003, KV is preferred, winning 7 of 13 comparisons with an average $+1.45$ F_1 advantage.

SNIPS mildly favors JSON, which leads in 7 of 13 comparisons with a small $+0.71$ F_1 average gain.

In sum, format selection has minimal impact on ATIS and SNIPS, but proves decisive on SLURP, WikiNER, and CoNLL-2003, where KV consistently outperforms alternatives. For MEDIA, both KV and XML perform comparably.

Although KV performs best across many datasets, this preference is not systematic across all models, with some models showing different format preferences depending on the specific dataset.

These results raise a practical question: rather than committing to a single format in advance, can we choose the most suitable format automatically for each dataset–model pair without exhaustive trial and error?

5.2. Automated Format Selection (RQ2)

We aim to select the best format for each pair of model and dataset by estimating the performance of each format on a development set (Max-on-Dev strategy).

For better readability, we display the results by dataset, averaged across models. For each dataset, each model has its dedicated format.

We compare three strategies averaged over 13 models per dataset: (i) Oracle (best-of-three per model), (ii) Random selection: uniform over JSON/KV/XML, and (iii) Max-on-Dev. Results appear in Table 5.

Datasets	Oracle F_1	Random	Max-on-Dev (100%)	Max-on-Dev (20%)
ATIS	84.60	84.16	84.48	84.35
SNIPS	80.52	79.87	79.93	79.89
SLURP	59.15	56.87	59.06	58.84
MEDIA	60.01	57.36	60.00	59.83
CONLL2003	59.59	58.28	59.26	58.97
WIKINER	63.69	61.96	62.85	62.36
E3C	43.20	40.68	42.03	42.03

Table 5: Format selection performance (in terms of F_1 score) across different selection strategies on 100% and 20% development set data. Oracle represents the theoretical upper bound, while Random, Max-on-Dev show performance of different selection approaches.

Max-on-Dev (Table 5) improves over Random selection by 1.01 F_1 on average, with the largest gains on format-sensitive datasets: MEDIA (+2.47), SLURP (+1.97), E3C (+1.35), versus modest gains on CoNLL-2003 (+0.69), WikiNER (+0.40), ATIS (+0.19), SNIPS (+0.02). Development-based selection thus delivers the most value precisely where format sensitivity is highest.

After applying Max-on-Dev, the residual gaps to Oracle performance remain small and dataset-dependent. The smallest gaps occur on MEDIA (0.18), ATIS (0.25), and SLURP (0.31), while slightly larger gaps persist on SNIPS (0.63), CoNLL-2003 (0.62), E3C (1.17), and WikiNER (1.33). These results confirm that on datasets like SNIPS and ATIS, the marginal benefit over Random selection is minimal, whereas on SLURP and MEDIA, Max-on-Dev provides decisive improvements.

5.2.1. Variation of Development-set size

We vary the development-set proportion from 10 to 100% and report the average gap to the Oracle (lower is better). The results are presented in Table 6.

Regarding the required development-set size, our analysis reveals that most benefits are achieved with surprisingly small data splits. The average

gap to Oracle decreases sharply with small development sets before plateauing: from approximately 0.80 at 10%, 0.64 at 20%, 0.55 at 50%, and finally 0.45 at 100%. Notably, diminishing returns appear beyond 30–50% of the data, indicating that in practice, using only 20% of available data for development recovers most of the achievable improvement, particularly on format-sensitive datasets. Note that the detailed 20% split results for the different datasets are presented in Table 5.

Despite the small performance gap, the computational and environmental cost of LLM inferences can be mitigated by taking only a small data split for format selection.

Dev%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Avg Gap to Oracle	0.80	0.64	0.70	0.61	0.55	0.54	0.47	0.48	0.47	0.45

Table 6: Average gap to the Oracle over all datasets (F_1) for the format selection strategy (development-set sample from 10 to 100%). The lower the better.

6. Conclusion

This work establishes output format as a critical, reportable feature in LLM-based information extraction evaluations. Through systematic experiments across 13 instruction-tuned models and 7 datasets, we demonstrate that format choices can induce statistically significant performance swings of 2–46 F_1points, depending on the model-dataset combination.

ATIS and SNIPS are largely format-agnostic, whereas SLURP, MEDIA, WikiNER, CoNLL-2003, and E3C show substantial sensitivity (64–82% of format pairs differ significantly). Across sensitive datasets, KV formatting emerges as the most reliable default. Lightweight format selection on just 20% of development data comes within 0.63 $F_1$of the oracle, mostly eliminating the need for exhaustive experimentation.

Treating format as a first-class experimental factor addresses a reproducibility gap: format-induced variance sometimes exceeds differences between model families, showing how unreported choices can obscure true advances.

Standardizing format reporting and selection procedures will yield more reliable benchmarks and fairer model comparisons.

7. Limitations & Future Work

We evaluate three structured formats but exclude natural language generation (e.g., "Paris is a location.") and IOB/BIO tagging schemes (e.g., B-LOCATION). Our prompts use a fixed template structure with 10 BM25-retrieved examples. Prompt variations (example count, instructions, self-consistency) and sampling-based decoding (temperature, top-p) might interact with format choice in ways we have not explored.

Our 13 models represent only open-weight instruction-tuned models; conclusions may not transfer to closed-source systems (GPT-4/5, Claude, etc.). Similarly, while datasets are standard benchmarks, they are in English, except for MEDIA (in French).

Future work could extend the analysis to constrained decoding methods (e.g. grammar-constrained generation), investigate format interactions with other hyperparameters (prompt design, temperature), and test whether format preferences generalize across languages and domains.

Finally, as models become more capable, tracking how format sensitivity scales will show whether this issue diminishes or persists.

8. Bibliographical References

Dhananjay Ashok and Zachary C. Lipton. 2023. PromptNER: Prompting For Named Entity Recognition.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding llms the right way: fast, non-invasive constrained generation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *arXiv preprint arXiv:1907.02884*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of EMNLP 2020*.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370.

Alessio Cocchieri, Marcos Martínez Galindo, Giacomo Frisoni, Gianluca Moro, Claudio Sartori, and Giuseppe Tagliavini. 2025. ZeroNER: Fueling zero-shot named entity recognition via entity type descriptions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15594–15616, Vienna, Austria. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Yixin Dong, Charlie F Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. 2024. Xgrammar: Flexible and efficient structured generation engine for large language models. *Proceedings of Machine Learning and Systems 7*.

Abhimanyu Dubey et al. 2024. The llama 3 herd of models.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *ACL*, pages 5467–5471.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of EMNLP 2023*.

Sahar Ghannay, Christophe Servan, and Sophie Rosset. 2020. Neural Networks approaches focused on French Spoken Language Understanding: application to the MEDIA Evaluation Task. In *COLING'2020, 2020*, Barcelona (online), Spain.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Stefan Hahn, Marco Dinarelli, Christian Raymond, Fabrice Lefevre, Patrick Lehnen, Renato De Mori, Alessandro Moschitti, Hermann Ney, and Giuseppe Riccardi. 2011. Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1569–1583.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 715–719. ISCA.

James Hammerton. 2003. Named entity recognition with long short-term memory. pages 172–175.

Mutian He and Philip N. Garner. 2023. Can chatgpt detect intent? evaluating large language models for spoken language understanding. In *Interspeech 2023*, pages 1109–1113.

Niu Hu, Xuan Zhou, Bing Xu, Hanqing Liu, Xiangjin Xie, and Hai-Tao Zheng. 2023. VPN: Variation on Prompt Tuning for Named-Entity Recognition. *Applied Sciences*, 13(14).

Mikaela Keller, Samy Bengio, and Siew Wong. 2005. Benchmarking non-parametric statistical tests. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language

model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Amirhossein Layegh, Amir H Payberah, Ahmet Soylu, Dumitru Roman, and Mihhail Matskin. 2023. ContrastNER: Contrastive-based Prompt Tuning for Few-shot NER. *arXiv preprint arXiv:2305.17951*.

Pierre Lepagnol, Sahar Ghannay, Thomas Gerald, Christophe Servan, and Sophie Rosset. 2025. Leveraging Information Retrieval to Enhance Spoken Language Understanding Prompts in Few-Shot Learning. In *Interspeech 2025*, pages 4108–4112.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. KnowCoder: Coding structured knowledge into LLMs for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779, Bangkok, Thailand. Association for Computational Linguistics.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *CoRR*, abs/1609.01454.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. ArXiv:2301.13688 [cs].

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016a. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. ACL.

Xuezhe Ma and Eduard Hovy. 2016b. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Paramita Mirza, Viju Sudhi, Soumya Sahoo, and Sinchana Ramakanth Bhat. 2024. Illuminer: Instruction-tuned large language models as few-shot intent classifier and slot filler. In *LREC-COLING 2024*, Torino, Italy.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Marco Naguib, Xavier Tannier, and Aurélie Névéol. 2024. Few-shot clinical entity recognition in english, french and spanish: masked language models outperform generative model prompting.

Noor Nashid, Mifta Sintaha, and Ali Mesbah. 2023. Retrieval-based prompt selection for code-related few-shot learning. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 2450–2462.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Wenbo Pan, Qiguang Chen, Xiao Xu, Wanxiang Che, and Libo Qin. 2023. A preliminary evaluation of chatgpt for zero-shot dialogue understanding.

Kanghee Park, Timothy Zhou, and Loris D'Antoni. 2025. Flexible and efficient grammar-constrained decoding. In *Forty-second International Conference on Machine Learning*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Bj"orkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis of OntoNotes corpora. In *Proceedings of CoNLL 2013*, pages 143–152, Sofia, Bulgaria.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China. Association for Computational Linguistics.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197.

Libo Qin, Fuxuan Wei, Qiguang Chen, Jingxuan Zhou, Shijue Huang, Jiasheng Si, Wenpeng Lu, and Wanxiang Che. 2024. Croprompt: Cross-task interactive prompting for zero-shot spoken language understanding.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021b. A survey on spoken language understanding: Recent advances and new frontiers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4577–4584. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction.

Christophe Servan, Christian Raymond, Frédéric Béchet, and Pascal Nocera. 2006. Conceptual decoding from word lattices: application to the spoken dialogue corpus MEDIA. In *Interspeech 2006 - ICSLP*, Pittsburgh, United States.

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.

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 CoNLL-2003*, pages 142–147, Edmonton, Canada.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *Proceedings of the IEEE ICASSP*.

Fabián Villena, Luis Miranda, and Claudio Aracena. 2024. llmner: (zero|few)-shot named entity recognition, exploiting the power of large language models.

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.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314, New Orleans, Louisiana. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought

prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2023. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys*, 55(8):1–38.

Brandon T Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 583–593, Mexico City, Mexico. Association for Computational Linguistics.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual nlu.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. 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 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

Hédi Zeghidi and Ludovic Moncla. 2024. Evaluating named entity recognition using few-shot prompting with large language models.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019a. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction Tuning for Large Language Models: A Survey. ArXiv:2308.10792 [cs].

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2993–2999. IJCAI/AAAI Press.

Z. Zhang, Z. Zhang, H. Chen, and Z. Zhang. 2019b. A joint learning framework with bert for spoken language understanding. *IEEE Access*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. Sglang: Efficient execution of structured language model programs.

Jizhao Zhu, Akang Shi, Zixuan Li, Long Bai, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025. Towards robust universal information extraction: Dataset, evaluation, and solution. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28052–28070, Vienna, Austria. Association for Computational Linguistics.

Zhihong Zhu, Xuxin Cheng, Hao An, Zhichang Wang, Dongsheng Chen, and Zhiqi Huang. 2024. Zero-shot spoken language understanding via large language models: A preliminary study. In *LREC-COLING 2024*, pages 17877–17883, Torino, Italia. ELRA and ICCL.

9. Language Resource References

Nadège Alavoine, Gaëlle Laperriere, Christophe Servan, Sahar Ghannay, and Sophie Rosset. 2024. New semantic task for the french spoken language understanding media benchmark. In *LREC-COLING 2024*, LREC-COLING 2024, Torino, Italy.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. Slurp: A spoken language understanding resource package. In *EMNLP*, pages 7252–7262, Online. Association for Computational Linguistics.

H. Bonneau-Maynard, Sophie Rosset, C. Ayache, A. Kuhn, and Djamel Mostefa. 2005. Semantic annotation of the french media dialog corpus. In *Proc. Interspeech 2005*, pages 3457–3460.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco

Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proc. of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Gaëlle Laperrière, Valentin Pelloin, Antoine Caubrière, Salima Mdhaffar, Nathalie Camelin, Sahar Ghannay, Bassam Jabaian, and Yannick Estève. 2022. The spoken language understanding media benchmark dataset in the era of deep learning: data updates, training and evaluation tools. In *LREC 2022*, pages 1595–1602, Marseille, France. European Language Resources Association.

Bernardo Magnini, Begona Altuna, Alberto Lavelli, Manuela Speranza, and Roberto Zanoli. 2021. The E3C Project: European Clinical Case Corpus. *Language*, 1(L2):L3.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2013. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

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.