Analysing Lightweight Large Language Models for Biomedical Named Entity Recognition on Diverse Ouput Formats

Pierre Epron[1,2], Adrien Coulet[1*], Mehwish Alam[2*]
Inria, Inserm, Université Paris Cité, HeKA, UMR 1346, Paris, France[1]
Télécom Paris, Institut Polytechnique de Paris, France[2]
pierre.epron@inria.fr, adrien.coulet@inria.fr, mehwish.alam@telecom-paris.fr

Abstract

Despite their strong linguistic capabilities, Large Language Models (LLMs) are computationally demanding and require substantial resources for fine-tuning, which is unadapted to privacy and budget constraints of many healthcare settings. To address this, we present an experimental analysis focused on Biomedical Named Entity Recognition using lightweight LLMs, we evaluate the impact of different output formats on model performance. The results reveal that lightweight LLMs can achieve competitive performance compared to the larger models, highlighting their potential as lightweight yet effective alternatives for biomedical information extraction. Our analysis shows that instruction tuning over many distinct formats does not improve performance, but identifies several format consistently associated with better performance.

Keywords: Large Language Models, Named Entity Recognition, Biomedicine, Resource-constrained NLP

1. Introduction

Generative Named Entity Recognition (G-NER) offers a promising paradigm shift from traditional span-based or classification-based approaches by framing entity extraction as a text generation task (Xu et al., 2024a). In this setting, structured prediction plays a central role as it enables models to generate outputs with predefined structures. For example, in Information Extraction (IE), structured prediction inputs unstructured texts and outputs meaningful structured representations of entities, relations or events.

Recent advances in this field have been made possible by instruction-tuned Large Language Models (LLMs) trained on vast multi-domain datasets (Wang et al., 2023b; Zhou et al., 2024). While these models enable strong generalization, their large scale is inappropriate for some applications. For example, in the biomedical domain, reducing the scope of the domain could lead to smaller and faster models, better suited to the budget and privacy constraints associated with many applications in healthcare.

In addition to model size, the choice of output format also plays a crucial role. Current research heavily relies on a single output format, such as JSON or a corpus-specific template, yet such fixed formats may introduce bias and restrict usability.

In this work, we investigate instruction tuning on top of lightweight LLM for the task of NER in the biomedical domain, and specifically examine how the size of the LLM and the choice of a single output format influences NER performance. We designed experiments across twelve different output formats and analyze their impact on model effectiveness.

Our results indicate that lightweight LLMs are fully capable of performing biomedical NER, and that tuning a format-agnostic, generative system yields only marginal performance gains. This underscores the approach's flexibility, practical applicability in resource-constrained settings, and inherent robustness to potential biases, making it a compelling alternative to larger, more rigid models. Additionally, we observed that in contrast to our initial intuition, instruction tuning over many distinct formats does not improve performance. However, we identify output format consistently associated with better performance. This article is organized as follows.

Section 2 discusses related works and positions our contribution w.r.t. the State-of-The-Art (SoTA). Section 3 presents the proposed method, while Section 4 and 5 detail experimental analysis for investigating the impact on G-NER of model size and output format. Finally, sections 6 and 6 conclude and discuss limitations of the current work. You can find the code on this repository: https://github.com/PierreEpron/MF-NER

2. Related Work

Since the rise of LLMs there has been remarkable progress in the task of IE. This section discusses most recent methods for NER based on LLMs. Others are excluded for the consistency of comparison and space limitation. See Xu et al. (Xu et al., 2024b) for a recent and detailed survey.

Prompting based Approaches. Zero-shot Information Extraction (ZIE) aims to reduce reliance

*These authors share last authorship.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2458–2470
11–16 May 2026.

on annotated data while maintaining strong performance. Inspired by LLMs such as GPT-3 and ChatGPT, ChatIE (Wei et al., 2023) reformulates ZIE as a multi-turn question-answering problem using a two-stage framework. It is evaluated on NER, entity-relation extraction and event extraction tasks across multiple languages. ChatIE achieves performance competitive with fully supervised models, demonstrating the potential of resource-efficient (i.e., without training) IE approaches based only on inferencing. In parallel, LLMs with code-style prompts have demonstrated notable success. In (Bi et al., 2024), the authors frame knowledge graph construction as a code completion task, utilizing schema-aware prompts and rationale-enhanced generation to improve extraction accuracy.

Few-shot learning approaches address cross-domain adaptation with limited data. GPT-NER (Wang et al., 2023a) recognizes entities by surrounding them with the special characters `#` and `@`. They evaluate multiple solutions of few-shot demonstration retriever and report their best results which were achieved using entity level embeddings. Paolini et al. (Paolini et al., 2021) unify structured generation for diverse IE tasks, while Chen et al. (Chen et al., 2023) propose a collaborative domain-prefix tuning for cross-domain NER.

CodeIE (Li et al., 2023b) leverages code-style prompts with in-context examples to achieve superior few-shot performance. Inverse generation, where structured data is converted into text or questions, has emerged as another effective strategy. SynthIE (Josifoski et al., 2023) generates high-quality synthetic data by reversing task directions, enabling downstream models to surpass previous benchmarks on the task of relation extraction.

RAG based Approaches. Retrieval Augmented Generation (RAG) based methods enhance model performance by leveraging auxiliary knowledge. Li et al. (Li et al., 2023a) introduce a two-stage multimodal NER framework that heuristically retrieves refined knowledge to improve entity prediction, while Amalvy et al. (Amalvy et al., 2023) generate synthetic context datasets and train neural retrievers to support NER on long documents. Code4UIE (Guo et al., 2024) addresses the challenges of non-unified prompts and limited in-context learning by introducing a Python class–based schema that standardizes diverse IE tasks and incorporates retrieval-augmented mechanisms.

Instruction Tuning based Approaches. (Hu et al., 2023) propose entity-to-text augmentation by manipulating entity lists and employing diversity beam search to improve dataset richness for NER. Instruction tuning benefits from these synthetic datasets as well. UniNER (Zhou et al., 2024) distills ChatGPT into smaller student models through mission-focused instruction tuning for open NER, while Ding et al. (Ding et al., 2024) incorporates negative instances in generative NER training on *The Pile* open-source corpus, improving zero-shot performance on unseen entity domains. Supervised fine-tuning approaches further enhance model capabilities. DeepStruct (Wang et al., 2022) pretrains LLMs on task-agnostic corpora to improve structural understanding, while GIELLM (Gan et al., 2023) fine-tunes LLMs on mixed datasets for Japanese IE, leveraging mutual reinforcement effects to improve performance across multiple tasks. GoLLIE (Sainz et al., 2024) fine-tunes LLMs with the instructions related to the annotation guidelines across a small set of IR tasks.

Wang et al. (Wang et al., 2023b) model IE as instruction-guided text generation. An option mechanism and auxiliary tasks refine span, relation, and event extraction. This improves structural and semantic understanding.

Other Approaches. Iterative self-improvement approaches such as ProgGen (Heng et al., 2024) guide LLMs through self-reflection to generate domain-relevant attributes and proactively construct NER context data, improving the quality of generated datasets. Constrained decoding strategies have been used to ensure structured outputs, e.g., (Geng et al., 2023) introduce grammar-constrained decoding with input-dependent grammars, while (Zaratiana et al., 2024) propose a text-to-graph framework generating linearized graphs with a transformer encoder-decoder architecture, a pointing mechanism and dynamic vocabularies for joint entity and relation extraction.

While existing studies have explored various methods for G-NER, our work experimentally demonstrates that lightweight LLMs, when used with suitable output formats, can achieve performance competitive with larger models.

3. Methodology

In this work, we focus on Causal Language Models (CLMs), which generate text autoregressively by predicting each token given its preceding context. Trained with the Next Token Prediction (NTP) objective, CLMs naturally align with text generation tasks, making them well-suited for instruction tuning. We adopt this framework to treat NER as a text generation problem, where entities are produced directly as model outputs. Compared to encoder-decoder architectures like BART, CLMs are more compatible with instruction-following objectives.

3.1. Instruction Tuning

Instruction tuning involves fine-tuning a model on a dataset specifically designed for a given downstream task, in this case, NER. Such datasets comprise pairs composed of one natural language instruction and its corresponding expected outputs. Formally, these datasets are represented as:

$$D_{instr} = \{(x_i, y_i)\}_{i=1}^{N} \quad (1)$$

where x_i denotes a natural language instruction, y_i is the corresponding target output, and N is the total number of instruction–response pairs. Instruction-tuning can further be adapted for the task of NER as follows (Zhou et al., 2024):

$$D_{instr} = \{(x_i, y_i, t_i, d_i)\}_{i=1}^{N} \quad (2)$$

where t_i represents the $i-th$ entity type to be recognized by the model and d_i represents a document used to encode characteristics specific to the dataset. For example, this document be an annotation guidelines that specifies recommendations for ambiguous cases such as "lung tumor" that can either be considered as an `BODY PART` or as a `DISEASE` depending on the context. Similarly, we add f_i that represents the specified output format to enrich the instruction-tuning dataset as follow:

$$D_{instr} = \{(x_i, y_i, t_i, d_i, f_i)\}_{i=1}^{N} \quad (3)$$

3.2. Formats

In this study, a format refers to the specific manner in which retrieved entities are represented within the system output. Each format defines a unique encoding structure. For example, the BIO (Beginning, Inside, Outside) is a widely adopted format for representing entities in annotated corpora used for training and evaluating supervised NER models. For this study, we selected twelve formats that ensure broad coverage of commonly used representation variants. We tested these different formats in different configurations of instruction tuning: training with either a single format or by mixing several (see Section 4.4 for the configurations). The rationale behind our choices is detailed below, with an example corresponding to the user prompt as shown in Listings 1 and 2.

Listing 1: Prompt

```
The task you need to complete is
named entity recognition. Follow {
dataset} guidelines.

{format}

Text: {text}
```

Listing 2: Prompt Arguments

```
dataset = Genia
text = These results suggest that
BCL6 plays a role in activated
lymphocytes as an immediate early
gene.
```

The following *conv_term* and *single_tag* formats are selected because those are used in the SoTA approaches, i.e., UniNER and GPT-NER respectively. We also selected a *multi_tag* format that uses standard XML tags (e.g., <person></person>).

conv_term

```
user :: Type: cell_line
assistant :: Answer: []
user :: Type: protein
assistant :: Answer: ["BCL6"]
```

single_tag

```
assistant :: Answer: These results
suggest that BCL6 plays a role in
@@activated @@lymphocytes\#\#\#\# as
 an immediate early gene.
```

multi_tag

```
assistant :: Answer: These results
suggest that <protein>BCL6</protein>
 plays a role in <cell_type>
activated <cell_type>lymphocytes</
cell_type></cell_type> as an
immediate early gene.
```

The *single_code* and *multi_code* formats are selected because they outperform other formats in code completion tasks (Li et al., 2023c). Although our work does not directly address code completion, we include these formats for completeness and fair comparison, as they are supported by recent LLMs.

single_code

```
assistant :: Answer: These results
suggest that <protein>BCL6</protein>
 plays a role in <cell_type>
activated <cell_type>lymphocytes</
cell_type></cell_type> as an
immediate early gene.
```

multi_code

```
assistant :: Answer:
'''py
def named_entity_recognition(
input_text):
        """ extract entities from
        the input_text. """
```

```
input_text = "These results
suggest that BCL6 plays a
role in activated
lymphocytes as an immediate
early gene."
entity_list = []
# extracted entities for
cell_line, protein, RNA, DNA
, cell_type types.
entity_list.append({"text":
"activated lymphocytes", "
type": "cell_type"})
entity_list.append({"text":
"BCL6", "type": "protein"})
entity_list.append({"text":
"lymphocytes", "type": "
cell_type"})
```

We select *single_term* and *multi_term* because they are standard formats for string extraction. Similarly, for the segment extraction task, we choose *single_span* and *multi_span*. Since triple extraction is a common task within IE, we add *multi_triple* format. Indeed, NER can be seen as a triple extraction where the predicate of the triple is forced to be “rdf:type" or “is a".

single_term

```
assistant :: Answer: ["activated
lymphocytes", "lymphocytes"]
```

multi_term

```
assistant :: Answer: [{"text": "
activated lymphocytes", "type": "
cell_type"}, {"text": "BCL6", "type
": "protein"}, {"text": "lymphocytes
", "type": "cell_type"}]
```

single_span

```
assistant :: Answer:
[[48,69],[58,69]]}
```

multi_span

```
assistant :: Answer: [{"span": [48,
69], "type": "cell_type"}, {"span":
[27, 31], "type": "protein"}, {"span
": [58, 69], "type": "cell_type"}]
```

multi_triple

```
assistant :: Answer:
activated lymphocytes; is a;
cell_type
BCL6; is a; protein
lymphocytes; is a; cell_type
```

Finally, *multi_bio* and *multi_brat* are well known annotation schema that model might have process multiple time during its pretraining or first instruction tuning.

multi_bio

```
assistant :: Layer 1: O O O O B-
protein O O O O B-cell_type I-
cell_type O O O O O
Layer 2: O O O O O O O O O O B-
cell_type O O O O O
```

multi_brat

```
assistant :: Answer:
T1        cell_type 48 69
activated lymphocytes
T2        protein 27 31        BCL6
T3        cell_type 58 69
lymphocytes
```

Please note that all considered formats enable the representation of nested entities, but that *single_span*, *multi_span*, and *multi_bio* do not enable representing discontinuous entities. *single_span* cannot be adapted to discontinuous entities, but the two other formats could by relying on indexing and BIOHD. However, we avoided these adaptations for the sake of simplicity.

3.3. Training

Model. We parameterize the conditional distribution of the output sequence given the input instruction using an auto-regressive language model:

$$P_\theta(y|x) = \prod_{t=1}^{T} P_\theta(y_t|x, y_{<t}) \tag{4}$$

where θ denotes the learnable parameters of the model, T is the length of the output sequence y, y_t is the token at position t, and $y_{<t}$ represents the preceding tokens in the sequence.

Loss Function. The training objective is based on minimizing the negative log-likelihood of the correct output sequence given the instruction. For a single instruction–response pair (x, y), the instruction-tuning loss is defined as:

$$\mathcal{L}_{instr}(x, y) = -\sum_{t=1}^{T} log P_\theta(y_t|x, y_{<t}) \tag{5}$$

which corresponds to the standard cross-entropy loss over tokens. More explicitly, the cross-entropy formulation is given by:

$$\mathcal{L}_{CE} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{\mathcal{V}} y_i^{(t)} log\ \hat{y}_i^{(t)} \tag{6}$$

where $\mathcal{V}$ is the vocabulary size, $y_i^{(t)}$ is the one-hot encoding of the correct token at step t, and $\hat{y}_i^{(t)}$ is the vector of predicted probabilities distribution over the vocabulary.

4. Experimental Setting

4.1. Datasets

We selected eight BioNER datasets for our analysis. See Table 1 for detailed statistics of each dataset.
AnatEM (Pyysalo and Ananiadou, 2014) contains PubMed abstracts, annotated for anatomical entities, including organs, tissues, and body parts.
BioCreative II Gene Mention (BC2GM) (Smith et al., 2008) is a PubMed-based corpus annotated for gene and protein mentions.
BC4CHEMD (Krallinger et al., 2015) is also a PubMed-based corpus annotated for chemical compound and drug names.
BC5CDR (Li et al., 2016) is a benchmark corpus for BioNER and relation extraction, focusing on chemical and disease entities.
CADEC (Karimi et al., 2015) is a corpus of user-generated generated content from forum posts about medication experience (AskAPatient.com) annotated for drugs, adverse events, and related attributes.
GENIA (Kim et al., 2003) is a MEDLINE-derived corpus manually annotated for proteins, DNA, RNA, cell types, and cell lines using a structured ontology.
NCBI Disease (Doğan et al., 2014) is a PubMed-based corpus manually annotated for disease mentions, normalized to MEDIC identifiers.
PGxCorpus (Legrand et al., 2020) is a PubMed-based pharmacogenomics corpus annotated for genes, drugs, phenotypes, and their interactions.

4.2. Baselines

We compare our performances against UniNER and InstructIE (See Section 2). The baselines are further complemented with a standard BERT model and GLiNER, when available. GLiNER is a lightweight, open-source NER model that leverages a bidirectional transformer encoder, such as BERT. Unlike conventional NER systems restricted to a fixed set of entity types, GLiNER supports a dynamic set of entity types using natural language prompts, making it well suited for zero-shot generalization across diverse domains. For PGxCorpus the baseline reported is from the original study, which employed Convolutional Neural Network for evaluation. For CADEC, we compare against the state-of-the-art Grid-Tagging approach (Liu et al., 2022), which frames the task as a classification of relationships between words as head, tail, or neighbor.

4.3. Language Models

We first conducted experiments using Qwen2.5-0.5B-Instruct[1] , a relatively lightweight LLM that demonstrates strong performance (Yang et al., 2024). To enable a reliable comparison between formats, it was important to select a model that could be trained multiple times without excessive computational cost. Following the same rationale, we replicated our experiments with Llama-3.2-1B-Instruct[2], a more recent model from the Meta LLaMA family (Dubey et al., 2024) featuring twice the number of parameters. We consider these two models as lightweight with regards to larger ones used in baseline experiments such as (Touvron et al., 2023; Chiang et al., 2023; Chung et al., 2024).

4.4. Training Configuration

We perform instruction tuning experiments using different format con- figurations. In the following we describe the formats used during training:

- *all* includes all the formats defined above in a balanced way.
- *7best* mixes only the seven best formats from the best results obtained with the *all* configuration (conv_term, multi_tag, multi_term, multi_triple, single_code, single_tag, single_term);
- *term_ner* includes *multi_term* and *single_term* and was used to evaluate how the use of the same multi and single formats could improve each other's performance when used in combination;
- Finally, the *only* configuration concerns models trained with a single format only. We did this training for the two best formats: *conv_term*, *multi_triple*. And with *multi_term* and *single_term* to compare with *term_ner*.

Following the experimental settings in UniNER and InstructIE, each split is a sampling since the total number of examples is too large. For the training set, we took a maximum of $10,000$ examples per dataset. We sampled 200 examples for the development set and 300 examples for the test set. For train and dev, we randomly selected one format depending on the configuration of the format and the compatibility of the dataset. We constrained examples from datasets containing discontinuous entities not to be associated with incompatible formats (*multi_bio*, *multi_tag*, *single_tag*). To ensure a robust evaluation, we tested each format from each configuration on the full test set.

The training hyper-parameters were chosen similar to UniNER and InstructIE. We trained each model for 4 epochs with a validation for each 1/4 of epoch. The validation is evaluated with micro-f1 and we used the best checkpoint for testing.

[1] Qwen2.5-0.5B-Instruct

[2] Llama-3.2-1B-Instruct

dataset	train	dev	test	nested	discont	labels
AnatEM	5861	2118	3830	-	-	1
bc2gm	12500	2500	5000	-	-	1
bc4chemd	30682	30639	26364	-	-	1
bc5cdr	4560	4581	4797	-	-	2
CADEC	5317	1140	1140	-	∼6.5-8.5%	5
GENIA	15023	1669	1854	∼2-2.5%	-	5
ncbi	5432	923	940	-	-	1
PGx	661	142	142	∼4-4.5%	∼0.8-1.8%	10

Table 1: Distribution of datasets used for this work. Columns nested and discount are a range approximation. Example for GENIA_NER, each split have between 2 and 2.5 percent of nested entities.

We used micro precision, recall, and f1 for both validation and test as those are standard metrics for NER. In order to evaluate the variability of the sampling procedure, we draw 3 samples for train, dev, and test. Each format configuration is trained and tested on each of the 3 samples and we reported the mean and standard deviation for each metrics over the 3 draws.

5. Results

The experimental results obtained indicate two primary findings. First, it is observed that despite the discrepancy in model size between the two base models (500M for Qwen-2.5 and 1B for Llama-3.2), the performance does not significantly decline. This is supported by the fact that the same applies to the baselines considered, which are all based on the models with 7B+ number of parameters. In the medical field, lightweight LLMs appear to be adequate. Secondly, the maximum difference between a format trained alone and trained jointly other formats is 0.05, which is residual. As a consequence, training with several formats simultaneously does not compromise their performance. This creates flexible models for different real-world applications.

5.1. Overall results

In Table 2, we can see that even though Llama-3.2 can perform slightly better, Qwen-2.5 performance are very similar regardless of the configuration. For the best configuration, the F1 score of *conv_term* is 0.80 for both models. We also observe that the performances of the configuration *all* are lower due to some low performance of some formats. More specifically, Table 3 shows that the F1 scores of *multi_brat*, *multi_span*, and *single_span* are less than 0.15. The effectiveness of these formats depends on the model's ability to extract character spans, a task that yielded suboptimal results in our case potentially due to the use of lightweight LLMs. Finally, we can see that the performance of each format does not decline when there are trained with multiple format configurations.

Table 4 compares formats that are not compatible with discontinuous entities with the best overall formats. We observe good performance for *single_tag* and *multi_tag* formats, i.e., 0.83 and 0.82 respectively. While these two formats seem more appropriate for simple entities, they have the disadvantage of not covering all entities.

5.2. Baseline comparison

In Table 5, we see that UniNER slightly outperforms all our models on each dataset, despite being trained on multiple domains and in a single format. However, UniNER uses a model with 7 billion parameters, whereas our models have a maximum of 1 billion. Additionally, our models are tuned solely on biomedical domain, and their stability is demonstrated by the low standard deviation with three samples. From these observations, we can conclude that lightweight LLMs perform similarly than larger ones on a specific domain with less computation time.

5.3. Limitations of *multi_triple*

In Table 5, we observe that the format *multi_triple* has an f-score close to 0 for the AnatEM dataset. In Table 6, we observe that this is due to the recall of Qwen *only* training configuration. When looking at the results, in majority of the cases the terms were correctly extracted but the assigned entity type was "disease" when the only possible label of AnatEM is "anatomy" (see Listing 3).

Listing 3: Example of the AnatEM Dataset

```
Text: The mechanisms that confer
this rapid metastatic capacity to
lung tumors are unknown .
Entity: (lung tumors, anatomy)
Triple extracted: lung tumors; is a;
 disease
```

x	Qwen2.5-0.5B			Llama-3.2-1B		
	P↑	R↑	F1↑	P↑	R↑	F1↑
all	0.65 (±0.01)	0.52 (±0.01)	0.58 (±0.01)	0.67 (±0.01)	0.54 (±0.01)	0.60 (±0.01)
7best	0.81 (±0.01)	0.75 (±0.01)	0.78 (±0.01)	**0.82** (±0.00)	0.74 (±0.01)	0.78 (±0.00)
multi_triple	0.79 (±0.02)	0.68 (±0.01)	0.73 (±0.01)	0.80 (±0.01)	0.76 (±0.01)	0.78 (±0.01)
conv_term	0.81 (±0.01)	**0.79** (±0.01)	**0.80** (±0.00)	0.81 (±0.01)	**0.79** (±0.01)	**0.80** (±0.01)
term	0.81 (±0.01)	0.71 (±0.02)	0.76 (±0.01)	0.81 (±0.01)	0.73 (±0.01)	0.77 (±0.01)
multi_term	**0.82** (±0.01)	0.70 (±0.02)	0.75 (±0.01)	0.82 (±0.01)	0.70 (±0.02)	0.75 (±0.01)
single_term	0.78 (±0.01)	0.76 (±0.02)	0.77 (±0.01)	0.80 (±0.02)	0.75 (±0.01)	0.77 (±0.01)

Table 2: Precision (P), recall (R), f-score (F1) for each model and formats configuration. **Bold** is for best for each model. Underline is for best overall.

	Qwen2.5-0.5B			Llama-3.2-1B		
	all↑	7best↑	only↑	all↑	7best↑	only↑
conv_term	0.78 (±0.00)	0.78 (±0.01)	**0.80 (±0.00)**	0.79 (±0.01)	0.79 (±0.01)	**0.80 (±0.01)**
multi_brat	0.13 (±0.01)	-	-	0.17 (±0.01)	-	-
multi_code	0.56 (±0.01)	-	-	0.57 (±0.01)	-	-
multi_span	0.10 (±0.01)	-	-	0.12 (±0.01)	-	-
multi_term	0.74 (±0.01)	**0.75 (±0.01)**	**0.75 (±0.01)**	0.74 (±0.01)	0.74 (±0.01)	**0.75 (±0.01)**
multi_triple	**0.78 (±0.01)**	**0.78 (±0.01)**	0.73 (±0.01)	**0.79 (±0.01)**	**0.79 (±0.00)**	0.78 (±0.01)
single_code	**0.73 (±0.01)**	**0.73 (±0.01)**	-	**0.74 (±0.01)**	0.73 (±0.01)	-
single_span	0.11 (±0.01)	-	-	0.13 (±0.01)	-	-
single_term	0.77 (±0.01)	**0.78 (±0.01)**	0.77 (±0.01)	**0.79 (±0.00)**	0.78 (±0.00)	0.77 (±0.01)

Table 3: F-score for each base model and formats with respect to their training configuration. **Bold** is for best for each model. Underline is for best overall.

	Qwen2.5-0.5B	Llama-3.2-1B
conv_term	0.81 (±0.01)	0.82 (±0.01)
multi_bio	0.44 (±0.07)	0.60 (±0.03)
multi_tag	0.82 (±0.01)	**0.83 (±0.01)**
single_tag	**0.83 (±0.01)**	**0.83 (±0.01)**

Table 4: F-score without discontinuous datasets for each base model and formats with respect to their training configuration. **Bold** is for best for each model. Underline is for best overall.

In general, it is semantically correct to tag the term as a disease. But the instruction specifically refers to the anatomy type. The model clearly fails to learn how to use the requested type for the *multi_triple* format. It still performs as intended in *all* and *7best* configurations, leading to the conclusion that the usage of multiple formats help in learning the appropriate type in such specific configuration.

5.4. Complex Entities

Even though complex entities represent only a small proportion of the entities, they are important in the biomedical domain. All the previous studies on generative NER do not take into account complex entities (i.e., nested or discontinuous). In Table 7, it can be seen that overall nested entities are correctly recognized even if there is still a gap with not-complex entities. However, this is not the case for discontinuous entities. It seems that no particular format handles the complex entities better than others. The large gap between the performance of *multi_span* and *single_span* could be explained because we exclude these formats for PGxCorpus what restrict the experiment to simpler nested entities to recognize.

5.5. Qualitative Analysis of span errors

We investigated why span experiments was not working properly. To this end, we examined several erroneous outputs in detail. In addition to the usual issues, such as unrecognized and spurious entities, we found that many errors arise from shifts in the span boundaries. Examples 4 illustrate this phenomenon. Although the observed shifts seem to indicate that the shift is always negative, no constant value emerges, and generally the shift ranges between 1 and 5.

These observations suggest that smaller models may lack the capacity to reliably manipulate

	Ours		Supervised			ZeroShot	
	7best↑	conv_term↑	Base↑	UIE↑	UniNER↑	UniNER↑	GliNER↑
AnatEM	0.87	0.87	0.86	**0.89**	**0.89**	0.25	0.34
GENIA	0.72	0.77	0.73	0.76	**0.78**	0.54	0.56
bc2gm	0.79	0.80	0.81	0.81	**0.82**	0.46	0.48
bc4chemd	0.85	0.87	0.87	0.88	**0.89**	0.48	0.43
bc5cdr	0.86	0.86	0.85	**0.89**	**0.89**	0.68	0.66
ncbi	**0.88**	0.87	0.80	0.86	0.87	0.60	0.62
CADEC	0.70	**0.76**	0.74	-	-	-	-
PGX	0.62	**0.71**	0.59	-	-	-	-

Table 5: F-score of our best model vs. the baselines for each dataset. Best results are in **bold**.

	Qwen2.5-0.5B			Llama-3.2-1B		
	P↑	R↑	F1↑	P↑	R↑	F1↑
all	0.88 (±0.03)	0.86 (±0.03)	0.87 (±0.03)	0.87 (±0.02)	0.85 (±0.02)	0.86 (±0.01)
7best	0.87 (±0.02)	0.87 (±0.02)	0.87 (±0.02)	0.88 (±0.01)	0.86 (±0.03)	0.87 (±0.02)
only	**0.80** (±0.06)	**0.03** (±0.03)	**0.06** (±0.06)	0.84 (±0.06)	0.78 (±0.03)	0.81 (±0.04)

Table 6: Precision (P), Recall (R), F1-score (F1) for each training configuration of the multi_triple format. **Bold** are worst results for each configuration.

		Qwen2.5-0.5B		Llama-3.2-1B	
		Δ Nested↓	Δ Discont↓	Δ Nested↓	Δ Discont↓
all	conv_term	**0.07**	0.46	0.09	0.43
	multi_tag	0.39	0.81	0.32	0.82
	multi_term	0.34	0.48	0.36	0.43
	multi_triple	0.10	0.54	0.08	0.51
	single_code	0.07	**0.43**	**0.06**	**0.39**
	single_tag	0.40	0.81	0.42	0.81
	single_term	0.08	0.52	0.09	0.44
7best	conv_term	0.08	0.52	0.10	0.46
	multi_tag	0.39	0.82	0.49	0.81
	multi_term	0.27	0.51	0.32	0.45
	multi_triple	0.08	0.53	0.09	0.49
	single_code	**0.07**	**0.46**	**0.06**	**0.43**
	single_tag	0.40	0.81	0.47	0.81
	single_term	0.08	0.54	0.10	0.45
term	multi_term	0.29	0.50	0.32	0.45
	single_term	**0.09**	**0.47**	**0.10**	**0.43**
only	conv_term	0.08	0.52	0.09	0.48
	multi_term	0.26	0.50	0.28	0.49
	multi_triple	**0.04**	**0.38**	0.10	**0.46**
	single_term	0.11	0.54	**0.09**	0.55

Table 7: Difference of performances for nested/discontinuous and normal entities for each base model, training configuration, and format. Lowest difference for each model is in **bold**. Underlined values are lowest difference overall.

character-level spans. Future work involving larger models, as well as simpler span manipulation tasks, could enable a more precise assessment of the necessary model capacity for accurate character-span handling.

Listing 4: "Span shift examples"

```
Text: In comparison , by day 24 ,
the majority of groups also treated
with prednisolone displayed
significantly less corneal clouding
and neovascularization .
References: corneal (111:118)
Candidates: less co (106:113)
Shift: -5

Text: A good side effect - I take
the lipitor at night because my
husband had heard it works better
taken at night .
References: lipitor (32:39)
Candidates: e lipit (30:37)
Shift: -2

Text: In one group of
hemiparkinsonian rats ( n = 5 ) ,
caffeine caused a dose - dependent
recovery of the contralateral
forepaw stepping : ED50 = 2 . 4 mu
mol / kg / day [ 95 % CI , 1 . 9 - 3
 . 1 ] ) , reaching its maximum at
the dose of 5 . 15 mu mol / kg / day
 .
References: feine ca (50:58)
Candidates: caffeine (53:61)
Shift: -3
```

6. Conclusion

This study demonstrates that instruction-tuned, lightweight LLMs can achieve competitive performance in biomedical G-NER, challenging the dominance of larger-scale models. We identified the most effective output formats (formats conv_term and multi_triple) for representing biomedical entities, including complex cases such as nested and discontinuous entities. Furthermore, we showed that training the model on multiple formats simultaneously does not degrade performance, offering increased flexibility and, in some instances, improving the model's predictions. These findings suggest promising avenues for future work, including extending this approach to other IE tasks, such as relation extraction, and exploring whether multi-task formats, such as triples, may be particularly effective in some settings.

Limitations

We focused only on two models, although many other exist. For instance, investigating BioMistral-7B, which is pre-trained on medical data would have been interesting; however, to our knowledge, no lightweight version of this type of LLM is currently available. Additionally, we did not train with the least performing formats, opting instead to concentrate on the most promising ones to limit computational costs. We also did not attempt to optimize the training hyper-parameters, instead using those from previous work that we deemed appropriate. Finally, we chose models that had already been instruction-tuned, although it would have been equally valuable to assess the performance of basic pre-trained models. These aspects of our study represent potential directions for future work.

Acknowledgments

This work benefits from a PhD funding from AI4IDF. This work was granted access to the HPC resources of IDRIS under the allocation AD011016312 made by GENCI.

Bibliographical References

Arthur Amalvy, Vincent Labatut, and Richard Dufour. 2023. Learning to rank context for named entity recognition using a synthetic dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 10372–10382. Association for Computational Linguistics.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2024. Codekgc: Code language model for generative knowledge graph construction. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 23(3):45.

Xiang Chen, Lei Li, Shuofei Qiao, Ningyu Zhang, Chuanqi Tan, Yong Jiang, Fei Huang, and Huajun Chen. 2023. One model for all domains: Collaborative domain-prefix tuning for cross-domain NER. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI*, pages 5030–5038. ijcai.org.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Yan Bowen, and Min Zhang. 2024. Rethinking negative instances for generative named entity recognition. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 3461–3475. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Chengguang Gan, Qinghao Zhang, and Tatsunori Mori. 2023. GIELLM: japanese general information extraction large language model utilizing mutual reinforcement effect. *CoRR*, abs/2311.06838.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 10932–10952. Association for Computational Linguistics.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2024. Retrieval-augmented code generation for universal information extraction. In *Natural Language Processing and Chinese Computing - 13th National CCF Conference, NLPCC 2024, Hangzhou, China, November 1-3, 2024, Proceedings, Part II*, volume 15360 of *Lecture Notes in Computer Science*, pages 30–42. Springer.

Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang. 2024. Proggen: Generating named entity recognition datasets step-by-step with self-reflexive large language models. In *Findings of the Association for Computational Linguistics, ACL*, pages 15992–16030. Association for Computational Linguistics.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S. Yu. 2023. Entity-to-text based data augmentation for various named entity recognition tasks. In *Findings of the Association for Computational Linguistics: ACL*, pages 9072–9087. Association for Computational Linguistics.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(Suppl 1):S2.

Joël Legrand, Romain Gogdemir, Cédric Bousquet, Kevin Dalleau, Marie-Dominique Devignes, William Digan, Chia-Ju Lee, Ndeye-Coumba Ndiaye, Nadine Petitpain, Patrice Ringot, et al. 2020. Pgxcorpus, a manually annotated corpus for pharmacogenomics. *Scientific data*, 7(1):3.

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

Jinyuan Li, Han Li, Zhuo Pan, Di Sun, Jiahao Wang, Wenkun Zhang, and Gang Pan. 2023a. Prompting chatgpt in MNER: enhanced multimodal named entity recognition with auxiliary refined knowledge. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2787–2802. Association for Computational Linguistics.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023b. Codeie: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 15339–15353. Association for Computational Linguistics.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023c. Codeie: Large code generation models are better few-shot information extractors. *arXiv preprint arXiv:2305.05711*.

Jiang Liu, Donghong Ji, Jingye Li, Dongdong Xie, Chong Teng, Liang Zhao, and Fei Li. 2022. Toe: A grid-tagging discontinuous ner model enhanced by embedding tag/word relations and more fine-grained tags. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:177–187.

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. In *9th International Conference on Learning Representations, ICLR*. OpenReview.net.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations, ICLR*. OpenReview.net.

Larry Smith, Lorraine K Tanabe, Rie Johnson Nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(Suppl 2):S2.

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

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*, pages 803–823. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity

recognition via large language models. *arXiv preprint arXiv:2304.10428*.

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. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023. Zero-shot information extraction via chatting with chatgpt. *CoRR*, abs/2302.10205.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024a. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024b. Large language models for generative information extraction: a survey. *Frontiers Comput. Sci.*, 18(6):186357.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. An autoregressive text-to-graph framework for joint entity and relation extraction. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19477–19487. AAAI Press.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A. Appendix

A.1. Training hyperparameters

Table 8 lists the most important hyperparameters used for training. These were chosen to align with previous works, as explained above. The complete list of hyperparameters can be found on the GitHub repository:

Training hyperparameters	
#epoch	15
#GPU	4
train batch size	2
gradient accumulation	8
learning rate	1.00E-05
optimizer	adamw
learning rate schelduder	constant
warmup	no
max new token	128
eval interval	epoch
eval batch size	8
eval metric	micro-f1

Table 8: Most important training hyperparameters

A.2. Energy consumption

All experiments were conducted using four A100s (80 GB). Table 9 lists the number of training hours for each experiment, and an estimate of power consumption. This estimate is calculated using the Green Algorithms calculator[3], based on the GPU configuration used and the location (Europe, France).

[3] https://calculator.green-algorithms.org/

	Qwen2.5-0.5B			Llama-3.2-1B		
	HH:MM:SS	gCO2e	kWh	HH:MM:SS	gCO2e	kWh
all	02:23:46	234.47	4.57	03:57:50	388.6	7.58
7best	02:11:45	214.8	4.19	03:47:43	372.2	7.26
conv_term	01:34:26	154.13	3.01	02:36:45	255.79	4.99
multi_triple	02:07:55	208.24	4.06	03:44:02	367.29	7.16
single_term	01:43:38	168.89	3.29	02:51:37	280.38	5.47
multi_term	02:03:28	201.68	3.93	03:31:07	345.97	6.75
term_ner	01:55:18	188.56	3.68	03:02:57	298.42	5.82

Table 9: Estimation of the energy consumption of each experiments