

# VERIFINER: Verification-augmented NER via Knowledge-grounded Reasoning with Large Language Models

Seoyeon Kim\* Kwangwook Seo\* Hyungjoo Chae Jinyoung Yeo Dongha Lee†

Yonsei University

{emseoyk, tommy2130, mapoout, jinyeo, donalee}@yonsei.ac.kr

## Abstract

Recent approaches in domain-specific named entity recognition (NER), such as biomedical NER, have shown remarkable advances. However, they still lack of faithfulness, producing erroneous predictions. We assume that knowledge of entities can be useful in verifying the correctness of the predictions. Despite the usefulness of knowledge, resolving such errors with knowledge is nontrivial, since the knowledge itself does not directly indicate the ground-truth label. To this end, we propose VERIFINER, a post-hoc verification framework that identifies errors from existing NER methods using knowledge and revises them into more faithful predictions. Our framework leverages the reasoning abilities of large language models to adequately ground on knowledge and the contextual information in the verification process. We validate effectiveness of VERIFINER through extensive experiments on biomedical datasets. The results suggest that VERIFINER can successfully verify errors from existing models as a model-agnostic approach. Further analyses on out-of-domain and low-resource settings show the usefulness of VERIFINER on real-world applications.<sup>1</sup>

## 1 Introduction

Named entity recognition (NER) is a fundamental task in natural language processing that aims to identify entity mentions in input text and assign them to specific types (Mikheev et al., 1999; Lample et al., 2016). Previous works solve NER tasks by training models on human-annotated datasets (Kim et al., 2003; Li et al., 2016; Zhang et al., 2021), improving neural model architectures (Jeong and Kang, 2023), or leveraging external knowledge (Liu et al., 2019; Mengge et al.,

\*Equal contribution

†Corresponding author

<sup>1</sup>Our code and data are available at <https://github.com/emseoyk/VerifiNER>.

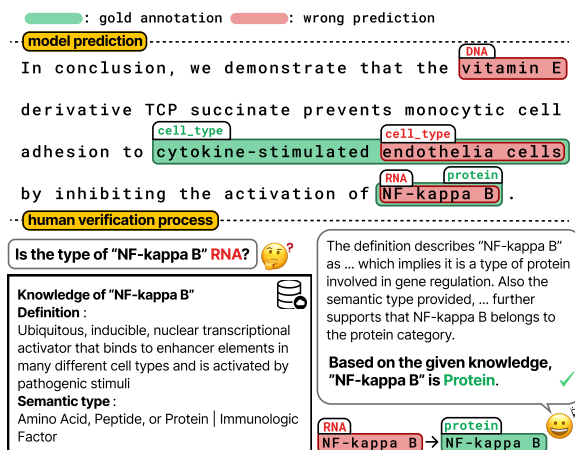


Figure 1: *Top*: Erroneous prediction of ConNER model on GENIA dataset. *Bottom*: Human verification process of correcting errors with knowledge from a KB. From knowledge, it is evident that labeling RNA is incorrect.

2020; Wang et al., 2021). More recently, simply prompting large language models (LLMs) without training can also perform NER tasks (Wang et al., 2023a; Ashok and Lipton, 2023).

Despite the promising results of these approaches, they still produce plausible but imprecise outputs. The risk to produce such errors is especially salient in domains that require expert-level knowledge such as the biomedical domain. Taking Figure 1 for example, “*NF-kappa B*” is incorrectly labeled as “RNA” type and “*endothelia cells*” is identified with wrong span. Erroneous prediction is a significant threat to their application in domains where high precision is required (Dai, 2021; Karim et al., 2023).

In this paper, we seek to minimize these errors by incorporating knowledge into the inference process. Figure 1 suggests that knowledge can serve as a useful evidence for humans to verify that the type of “*NF-kappa B*” is “protein”. However, avoiding the errors during inference time is challenging for neural models, because they lack in

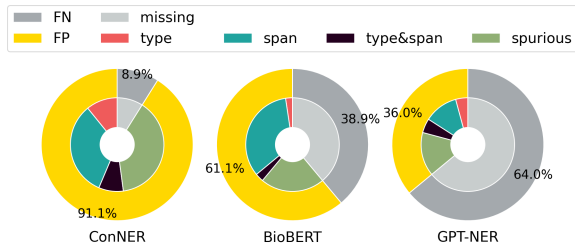


Figure 2: Ratio of error types (%) of NER models. Detailed statistics are reported in Table 7.

factual evidence that can assist the models to judge the correctness of predictions. Furthermore, even if the models are equipped with knowledge, there exists a mismatch between knowledge and entity prediction that discourages models from properly detecting and correcting the errors. For example, the definition of “*NF-kappa B*” as “an ubiquitous, inducible, nuclear transcriptional activator” does not explicitly indicate that the type is “protein”.

To resolve the aforementioned challenge, we aim to propose a post-hoc verification framework that identifies errors from existing NER methods and revises them into more faithful predictions. We employ knowledge base (KB) to provide factual evidence to existing NER systems, and leverage reasoning and in-context learning ability of LLMs to verify entities via knowledge-grounded reasoning. By verifying errors of the previous models in a post-hoc manner, our work exhibits remarkable performance without re-training models.

To this end, we introduce VERIFINER, a novel Verification-augmented NER via Knowledge-grounded Reasoning. With the notion that the errors can be alleviated with factual knowledge and contextual cue, our framework verifies entities in terms of *factuality* and *contextual relevance*. For factuality verification, VERIFINER formulates queries from each predicted entity to retrieve knowledge from KB, then reassigns its span and type based on the knowledge. The contextual relevance verification module employs the reasoning ability of LLMs to consider input context when selecting correct entity from candidates that are passed from previous step.

For effective demonstration of our framework, we conduct experiments on biomedical NER which requires domain-expert knowledge to solve the task. While our approach is model-agnostic, we validate that VERIFINER can be applied to both fine-tuned models (Lee et al., 2019; Jeong and Kang, 2023) and a LLM-based NER method (Wang

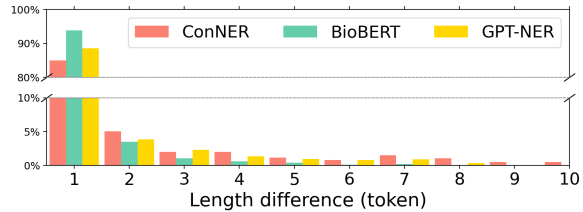


Figure 3: Distribution of token-level deviation between predicted and gold spans for each model. The x-axis represents the difference in token length between entities where the predicted span and gold span overlaps.

et al., 2023a) with significant improvement. In addition, we evaluate our framework under out-of-distribution and low-resource settings, showing its advantages on real-world scenarios.

To the best of our knowledge, this is the first work that solves NER with a verification module that exploits reasoning ability of LLMs. The main contributions of this work are as follows:

- We propose a novel framework for identifying and resolving errors via knowledge-grounded reasoning by utilizing knowledge and LLMs.
- We present VERIFINER, a post-hoc verification module that corrects entity prediction with respect to factuality and contextual relevance.
- We demonstrate the effectiveness and generalization ability of VERIFINER through extensive experiments and analyses.

## 2 Preliminary Analysis

In this section, we first define the common error types of NER we aim to resolve. Then, we provide further analysis on error cases in existing models to gain insight on how to rectify the errors.

**Error Types in NER.** For evaluating NER model predictions against the gold annotations, error cases can be categorized as either false positives (FP) or false negatives (FN) based on whether the model incorrectly identifies entities or fails to recognize the gold entities that is present. Within the FP, errors can occur from mismatches of type or span.

Depending on the combinations of these mismatches, errors are classified into more fine-grained error types, including Type, Span, Type&Span and Spurious errors. We provide definitions and examples of each error type in Table 1.

**Error Analysis on NER Models.** We conduct an analysis on existing errors using two fine-tuned models, *i.e.*, ConNER (Jeong and Kang, 2023),

Gold annotation: "... cell adhesion to cytokine-stimulated [endothelial cells] <sub>cell_type</sub> by .."			
Category	Error type	Definition	Example
FP	Type	Wrong type is assigned to an entity	"... cell adhesion to cytokine-stimulated [endothelial cells] <sub>cell_line</sub> by .."
	Span	Predicted span partially overlaps with ground truth, but incorrect	"... cell adhesion to [cytokine-stimulated endothelial cells] <sub>cell_type</sub> by .."
	T&S	Both type and span are predicted incorrectly	"... cell adhesion to [cytokine-stimulated endothelial cells] <sub>cell_line</sub> by .."
	Spurious	A completely incorrect entity is predicted where gold annotation does not exist	"... [cell adhesion] <sub>cell_line</sub> to cytokine-stimulated endothelial cells by .."
FN	Missing	A gold annotation for entity exists but not predicted by a model	"... cell adhesion to cytokine-stimulated endothelial cells by .."

Table 1: *Top*: Ground truth span and type of “endothelial cells”. *Bottom*: Definition and example per error type. T&S stands for Type&Span. Green denotes ground truth, and red denotes error in span and/or type.

BioBERT (Lee et al., 2019), and one prompting-based LLM, *i.e.*, GPT-NER (Wang et al., 2023a). Figure 2 shows the ratio of different error types for each model. In case of the fine-tuned models, false positive errors take the majority of the total error types. For prompting-based LLM, FP cases take more than one third. As low precision in domain-specific NER is a crucial problem (Dai, 2021; Karim et al., 2023) and FP constitute a significant proportion across all three models, we focus on correcting the FP cases from the initial predictions. By doing so, we can boost the NER performance regardless of the model that comes beforehand.

To gain clues on how to resolve the errors, we examine where existing models fall short to make precise predictions. Among the FP cases, we take a closer look on errors (*i.e.*, Type, Span, and T&S) where the models’ predictions partially overlap with the gold annotation, but either the type or span is incorrect. In Figure 2, almost 60% of FP cases are partially overlapping predictions for all three models. For span error, in Figure 3, more than 80% of predicted entities show difference in length with gold annotation within a margin of two tokens. This implies that NER models resort to plausible yet incorrect predictions with a small margin. Thus, instead of completely discarding the wrong entity predictions, we use them as queries to initialize our verification process in a post-hoc manner.

### 3 VerifNER

In this section, we propose VERIFNER, a framework that verifies errors from NER models with external knowledge and the reasoning ability of LLMs. In this paper, we define the term *verification* as a process that also includes revision. Motivated by the observations in Section 2, we design our framework to particularly focus on enhancing precision; that is, we first identify errors in predicted

entities and then correct them into accurate outputs. The overall framework is illustrated in Figure 4.

Formally, given an input sequence  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$  of  $n$  tokens and a predefined type set  $\mathcal{T}$ , our goal is to produce a revised entity prediction  $\bar{e}$  by verifying an entity  $e = (s, t)$  that is originally predicted by an off-the-shelf NER model, where  $s = [x_{\text{beg}} : x_{\text{end}}]$  and  $t \in \mathcal{T}$  represent its span and type, respectively. In this sense, the verification process targets both the span and type of the entity. Note that span verification must precede type verification, because predicting the type  $t$  of an entity  $e$  depends on the semantics of  $e$  from its span  $s$ . For example, the type of “PEBP2” is “protein”, but a longer span, “PEBP2 site”, is classified as “DNA”; thus, we first identify spans and then proceed to determine types.

Specifically, we first verify the factuality of the span and type (Sections 3.1 and 3.2), then verify whether the factually approved entity is relevant to the input context (Section 3.3) for precise revision.

#### 3.1 Span Factuality Verification

In this step, we rectify span error through collecting candidate spans from the predicted entity and verifying them by leveraging a KB. Based on the observation in Section 2, we assume that the gold entity is likely to be adjacent to the predicted entity. Therefore, we expand the range of candidate spans around the predicted entity span  $s$  in both directions to increase the likelihood of the gold entity within our candidate set.

We collect a set of candidate spans  $\tilde{\mathcal{S}}$  by extending the left and right offsets of a span  $s$  with hyperparameter  $\alpha$ , then enumerate sub-sequences within the offsets  $[x_{\text{beg}-\alpha} : x_{\text{end}+\alpha}]$ , so that each candidate partially or completely overlaps with the span  $s$  of the predicted entity  $e$ . As shown in Figure 4 (a), {“human mononuclear”, “human”,

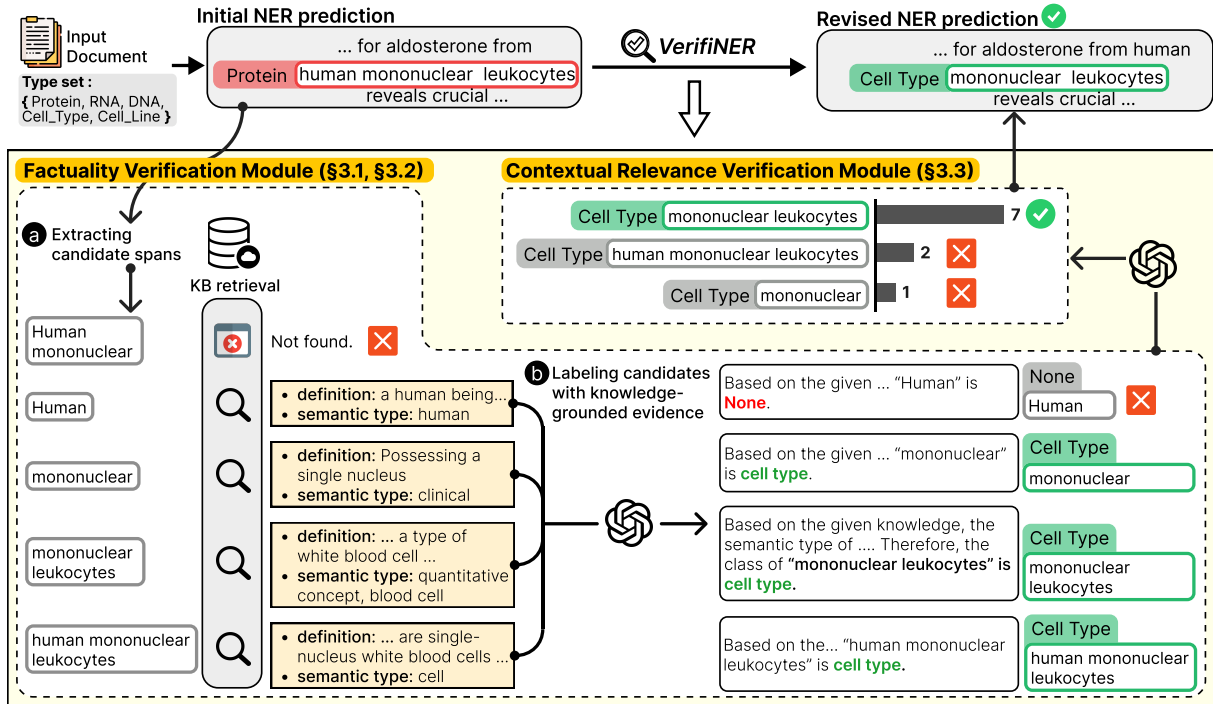


Figure 4: Overview of VERIFINER framework. (1) Using entity prediction by existing models, we (a) extract candidate spans to retrieve knowledge from KB and verify *factuality* of span accordingly. Then (b) using retrieved knowledge, we verify *factuality* of type by generating knowledge-grounded evidence. (2) Lastly, we take consistency voting to select a candidate that is the most *contextually relevant*, with the help of the reasoning ability of LLMs.

“from human”, ..., “mononuclear”, “mononuclear leukocytes”, “human mononuclear leukocytes”} can be extracted from the predicted entity “human mononuclear”.

Then, we search the KB to prune only factually valid candidates.<sup>2</sup> Using each candidate span  $\tilde{s}$  ( $\in \tilde{\mathcal{S}}$ ) as a query, we check whether knowledge for each candidate exists in the KB. If a candidate is found in the search, we consider its factuality is verified and collect its associated knowledge  $k$ . On the other hand, if the candidate is not defined in the KB, we assume it is a noisy candidate and do not consider it further in the remaining process. For example, in Figure 4 (a), the search result for “human mononuclear” is not found, and this candidate is excluded from the next step. In consequence, we obtain a pruned set of candidate spans  $\tilde{\mathcal{S}}$ .

### 3.2 Type Factuality Verification

As we collect the set of candidate spans  $\tilde{s}$  and their associated knowledge  $k$ , we proceed to re-assign types to  $\tilde{s}$  grounded on the retrieved knowledge.

<sup>2</sup>We assume a domain-specific NER scenario where all gold entities exist entirely within the KB. More explanation is in Appendix A.

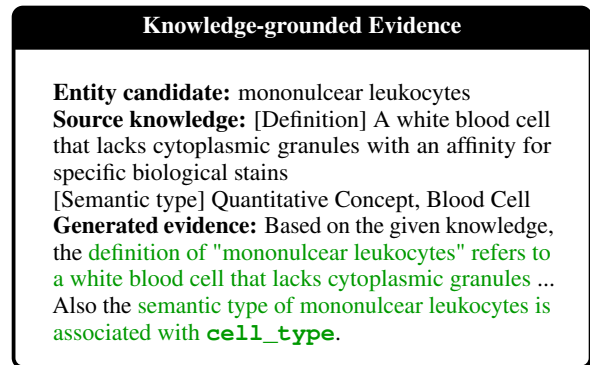


Figure 5: Example of generated evidence used for type factuality verification.

While knowledge serves as a reliable source for verifying candidates, directly applying it for verification is challenging. It is due to the fact that knowledge often lacks explicit indications regarding whether a candidate is correctly labeled. In Figure 5, the knowledge of “mononuclear leukocytes”, i.e., “a white blood cell ...” and “Quantitative Concept” do not exactly match with “cell\_type”.

To this end, we leverage the reasoning ability of LLMs to project knowledge into predefined types. We accomplish this by generating evidence  $k'$  to



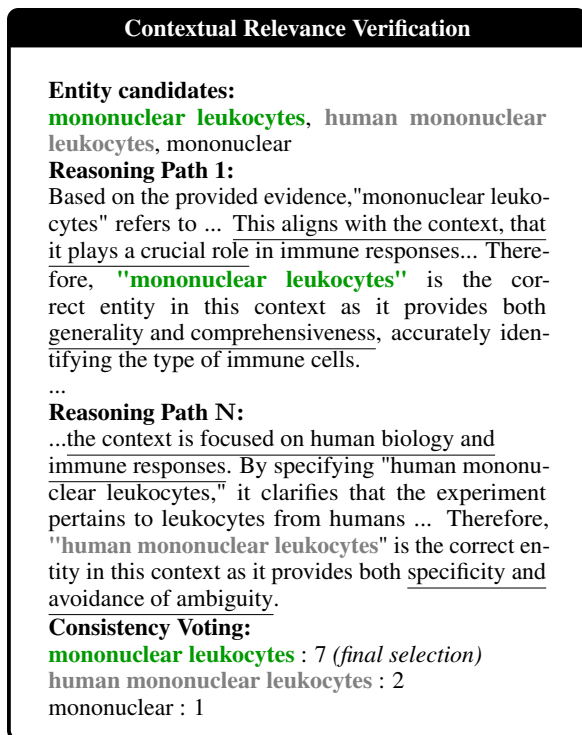


Figure 6: Example of generated diverse reasoning paths for contextual relevance verification.

assist in assigning types grounded on knowledge. Specifically, we rationalize the source knowledge  $k$  into verbalized form to generate  $k'$ . Then, we provide  $(\tilde{s}, k')$  with the predefined label set  $\mathcal{T}$  and prompt the LLM to re-assign type  $\tilde{t}$  based on the evidence  $k'$ . An example of knowledge-grounded evidence is provided in Figure 5. Based on provided definition and semantic type of the candidate "mononuclear leukocytes", LLM generates a knowledge-grounded evidence and assigns type to the candidate.

If the knowledge is irrelevant to the domain, the LLM will assign  $\tilde{t}$  as NONE. Consequently, each entity will have a set of factuality-verified candidates  $\tilde{e} \in \tilde{\mathcal{E}}$ , where each candidate is  $\tilde{e} = (\tilde{s}, \tilde{t})$ .

### 3.3 Contextual Relevance Verification

Lastly, we select a final candidate entity from  $\tilde{\mathcal{E}}$  based on contextual relevance for revising the prediction  $e$  to  $\bar{e}$ . A candidate  $\tilde{e}$  can be a valid entity  $\bar{e}$  if (1) it is semantically relevant to the input context  $\mathcal{X}$ , and (2) its knowledge-grounded evidence  $k'$  is aligned well with the context compared to other candidates. To determine the final candidate grounded on knowledge and context, we employ in-context learning ability of LLMs.

When verifying contextual relevance of the can-

didate, both local and global contextual information should be considered. However, reasoning only once may lead to insufficient attention on limited contexts. Therefore, we sample multiple reasoning paths to gather answers that reflect various aspects of the context. This process resembles the human annotation process of gathering various opinions from multiple annotators and converging them into a single consistent answer through discussion. To facilitate this process, we employ self-consistency (Wang et al., 2023b) and use consistency voting to select the candidate that is most suitable for the context. As demonstrated in Figure 6, LLMs generate reasoning paths that are properly grounded on knowledge.

Given the input context  $\mathcal{X}$ , candidates  $\tilde{\mathcal{E}}$ , and evidence for each candidate  $k'$ , we prompt the LLMs to sample  $N$  reasoning paths where each path selects a single candidate that has the most faithful evidence and is relevant to  $\mathcal{X}$ . We conduct majority voting over the collected  $N$  answers and select the answer that receives the most votes. Finally, the prediction is revised as  $\bar{e} = (\bar{s}, \bar{t})$ . The prompts used in our framework are provided in Appendix D.

## 4 Experiments

In this section, we design our experiments to answer the following research questions:

- **RQ1:** Can VERIFINER faithfully identify and revise errors?
- **RQ2:** Can VERIFINER induce generalizability to fine-tuned models in various test distributions?
- **RQ3:** Can VERIFINER effectively mitigate the low-resource challenge in domain-specific NER?

### 4.1 Experimental Settings

We evaluate our framework on two biomedical datasets, each with different set of predefined types: BC5CDR (Li et al., 2016) is annotated with Chemical and Disease, while GENIA (Kim et al., 2003) includes five types, including cell\_line, cell\_type, DNA, RNA, and protein. We use the Unified Medical Language System (UMLS) (Bodenreider, 2004) as a KB, which is a database containing over two millions of biomedical terminologies annotated with their definitions, semantic types, and lexical relationships. We employ ChatGPT as an LLM to implement our framework (OpenAI, 2023).<sup>3</sup> Following Jeong and

<sup>3</sup>We use gpt-3.5-turbo-1106 model among the various versions of ChatGPT.

Methods	GENIA				BC5CDR			
	P	R	F1	$\Delta$ F1	P	R	F1	$\Delta$ F1
GPT-NER (Wang et al., 2023a)	56.44	42.15	48.26	-	79.84	<b>47.48</b>	59.55	-
+ Manual Mapping	37.53	32.65	34.93	-13.33	51.82	36.98	43.16	-16.39
+ LLM-revision	52.97	<b>46.77</b>	49.68	+1.42	77.21	44.53	56.48	-3.07
+ LLM-revision w/ CoT	53.57	44.54	48.64	+0.38	76.49	44.91	56.59	-2.96
+ VERIFINER (Ours)	<b>72.37</b>	44.95	<b>55.46</b>	<b>+7.20</b>	<b>91.01</b>	46.92	<b>61.92</b>	<b>+2.37</b>
ConNER (Jeong and Kang, 2023)	74.13	<b>96.69</b>	83.92	-	84.90	<b>96.47</b>	90.32	-
+ Manual Mapping	43.62	94.50	59.69	-24.23	53.98	94.52	68.71	-21.61
+ LLM-revision	63.64	86.64	73.38	-10.54	80.35	93.07	86.25	-4.07
+ LLM-revision w/ CoT	64.85	86.92	74.28	-9.64	78.14	92.99	84.92	-5.40
+ VERIFINER (Ours)	<b>79.07</b>	91.82	<b>84.97</b>	<b>+1.05</b>	<b>94.77</b>	91.61	<b>93.16</b>	<b>+2.84</b>
BioBERT (Lee et al., 2019)	54.51	65.30	59.42	-	79.93	<b>95.98</b>	87.22	-
+ Manual Mapping	30.57	24.39	27.14	-32.28	38.65	65.78	48.69	-38.53
+ LLM-revision	52.63	65.01	58.17	-1.25	60.79	77.74	68.23	-18.99
+ LLM-revision w/ CoT	52.21	63.49	57.30	-2.12	59.43	78.66	67.71	-19.51
+ VERIFINER (Ours)	<b>77.45</b>	<b>67.75</b>	<b>72.31</b>	<b>+12.89</b>	<b>94.02</b>	91.17	<b>92.57</b>	<b>+5.35</b>

Table 2: Results of VERIFINER on GENIA and BC5CDR compared to baselines. The performance is evaluated on test set based on the entity-level exact matching.  $\Delta$ F1 indicates the improvement on F1 from the initial models.

Kang (2023), we measure the performance utilizing entity-level Precision (P), Recall (R), and F1.

## 4.2 NER Models

To validate the effectiveness of VERIFINER as a model-agnostic approach, we employ two groups of NER models as a test-bed for our evaluation: For fine-tuned models, we use **ConNER** (Jeong and Kang, 2023) and **BioBERT** (Lee et al., 2019), trained for the NER task on each dataset. For prompting-based LLMs, we consider **GPT-NER** (Wang et al., 2023a), which predicts entity spans of interest and their types in a generative manner via few-shot prompting.

## 4.3 Baseline Methods

We compare VERIFINER with other methods that revise the initial prediction in a post-hoc manner. (1) **Manual Mapping**: Naive use of external knowledge to verify predictions. We reassign entity types by manually mapping semantic types found in the KB to the predefined labels. Assuming that entities with the same semantic type will have the same label accordingly, the purpose of Manual Mapping is to solely rely on the KB and not the reasoning function of LLM. To construct a semantic type-label map, we use entities from the train set to retrieve the semantic types from KB and manually align them with the label. For example, if the KB retrieves [“Organic Chemical”, “Pharmacologic Substance”] as semantic types of “cypoterone acetate” and its label is “Chemical”, then we align the semantic

type “Pharmacologic Substance” to the label “Chemical”. The statistics of semantic types assigned to each label are in Appendix A. (2) **LLM-revision**: Simple revision using an LLM, where the model re-examines the input context and generates revised context based on marked predicted entities. (3) **LLM-revision w/ CoT**: We incorporate zero-shot CoT (Kojima et al., 2022) in addition to LLM-revision. We provide prompts for LLM-revision and w/CoT in Appendix D.

## 5 Results

### 5.1 Effectiveness of VERIFINER (RQ1)

**Comparison with Baselines.** From the results in Table 2, we have the following observations: (1) For all NER models, VERIFINER consistently achieves significant improvements over initial predictions on both datasets. This demonstrates the effectiveness of our model-agnostic verification method. (2) VERIFINER also outperforms other revision baselines by notable margins. When comparing LLM-revision and w/ CoT to ours, we find that relying solely on the internal knowledge of LLMs degrades their performance, necessitating a reliable external knowledge source to faithfully verify entity predictions. (3) While the incorporation of reliable knowledge is essential for verification, the performance drop in Manual Mapping additionally highlights the need for an intermediate reasoning process to bridge the gap between the retrieved knowledge and model predictions. (4) As VERIFINER aims to precisely correct entity predic-

Methods		P	R	F1
GPT-NER	+VERIFINER	<b>72.37</b>	<b>44.95</b>	<b>55.46</b>
	w/o Consistency Voting (CV)	71.57	42.95	53.68
	w/o evidence, CV	43.49	34.32	38.36
	w/o KB	46.05	32.93	38.40
ConNER	+VERIFINER	<b>79.07</b>	<b>91.82</b>	<b>84.97</b>
	w/o Consistency Voting (CV)	77.63	86.14	81.66
	w/o evidence, CV	52.70	91.09	66.77
	w/o KB	50.82	82.42	62.87

Table 3: Ablation results on GENIA.

tions, an inevitable decrease in recall is expected. However we notice that VERIFINER achieves considerable increase in precision across all NER models without a severe drop in recall. Across all datasets and models, the average increase in precision is 20.03%, while recall decreases by only 1.09%. This indicates that our approach can faithfully correct errors without imposing a significant trade-off between precision and recall. We provide further case study in Appendix C.

**Ablation Study.** To validate the effectiveness of each component in our framework, we conduct ablation studies as follows: excluding consistency voting (w/o CV), omitting knowledge-grounded evidence generation and consistency voting (w/o evidence, CV), generating evidence based on internal knowledge of LLMs (w/o KB). In Table 3, we have the following observations for each ablated approach: (1) Verifying contextual relevance with multiple reasoning paths is helpful for enhanced accuracy, suggesting that diverse aspects within the context should be considered to select the consistent answer. (2) Knowledge-grounded evidence largely affects the final NER accuracy, and this implies that it is important to bridge the gap between the collected knowledge and the target application (or task). (3) Parametric knowledge from the LLM still lacks in domain expertise, supporting the necessity of employing external knowledge in our framework.

**Analysis on Error Correction.** To have a better understanding of how VERIFINER works, we provide an in-depth analysis on error correction rate compared to other revision baselines. Figure 7 shows that VERIFINER rectifies errors more faithfully than the baselines across all error types. Notably, 52% and 78% of errors are corrected in total for GENIA and BC5CDR, respectively. Upon examining the results for each error type,

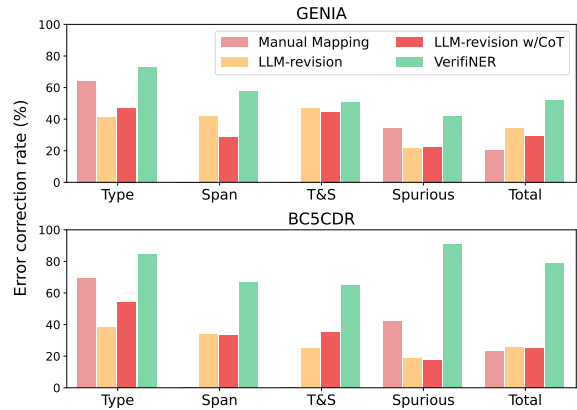


Figure 7: Error correction performance of VERIFINER compared to other revision baselines. We report averaged rates over all three NER models. Note that the results for Manual Mapping on Span and T&S does not appear as it can not correct span errors.

VERIFINER successfully corrects the majority of type errors, demonstrating its effective execution of knowledge-grounded re-typing. For span error, other revision baselines show significantly lower correction rates, while VERIFINER successfully rectifies it twice as much. When comparing the results across datasets, VERIFINER tends to struggle more on GENIA compared to BC5CDR, due to more fine-grained type set of GENIA. However, even on GENIA, VERIFINER manages to correct over half of the errors in most cases while other revision baselines tend to remain at less than one-third. Additionally, in the case of BC5CDR, it achieves correction rates close to 90% for spurious errors, highlighting its capability to properly filter out invalid entities.

## 5.2 Generalizability of VERIFINER (RQ2)

Although fine-tuned NER models show promising accuracy in in-distribution settings, *i.e.*, the train and test data are sampled from the same dataset (or distribution), they can hardly be applied to unseen labels or shifted distribution, lacking robustness in out-of-distribution (OOD) settings. Thus, we investigate if our VERIFINER framework can also prove effectiveness in OOD scenarios, where fine-tuned models are trained on the datasets that are distinct from the test distribution.

**Unseen Distribution.** We apply VERIFINER to the NER models fine-tuned on a source dataset, and evaluate it on a target dataset whose labels are not seen during the training (*i.e.*, train on GENIA and infer on BC5CDR, and vice versa). Intuitively, the

Source → Target	BC5CDR → GENIA			GENIA → BC5CDR		
	P	R	F1	P	R	F1
GPT-NER	56.44	42.15	48.26	79.84	47.48	59.55
ConNER	N/A	N/A	N/A	N/A	N/A	N/A
+ VERIFINER	<b>58.15</b>	<b>77.42</b>	<b>66.42</b>	<b>76.74</b>	<b>57.42</b>	<b>65.69</b>
BioBERT	N/A	N/A	N/A	N/A	N/A	N/A
+ VERIFINER	<b>66.49</b>	<b>87.25</b>	<b>75.47</b>	<b>77.64</b>	<b>71.17</b>	<b>74.27</b>

Table 4: Evaluation on the unseen distribution settings.

Source → Target	GENIA' → GENIA			BC5CDR' → BC5CDR		
	P	R	F	P	R	F
GPT-NER	56.44	42.15	48.26	79.84	47.48	59.55
ConNER	69.97	94.10	80.26	81.46	89.47	85.28
+ VERIFINER	<b>74.16</b>	90.03	<b>81.48</b>	<b>94.28</b>	85.42	<b>89.63</b>
BioBERT	35.36	64.50	45.68	61.75	69.95	65.59
+ VERIFINER	<b>78.91</b>	<b>69.68</b>	<b>74.01</b>	<b>94.16</b>	<b>71.65</b>	<b>81.38</b>

Table 5: Evaluation on the shifted distribution settings.

fine-tuned models themselves cannot be evaluated in the cross-dataset setting due to the different entity type sets between the source and target datasets. Thus, we denote them as not applicable (N/A) in Table 4. On the contrary, the fine-tuned models equipped with VERIFINER perform much better than the prompting-based LLM (GPT-NER), and even achieve comparable performance to baselines trained on the target dataset. This suggests that VERIFINER can heighten the performance of fine-tuned NER models on unseen datasets, where the training dataset is not accessible.

**Shifted Distribution.** Additionally, we evaluate the performance of VERIFINER in scenarios where training and test datasets have different distributions but their entity type sets are identical. Specifically, we split the original train data  $train$  into subsets  $train'_1, train'_2 \subset train$ , making their shifted type distributions distinct from the distribution of target, *i.e.*,  $\mathcal{D}_{train'_1}, \mathcal{D}_{train'_2} \neq \mathcal{D}_{test} (\approx \mathcal{D}_{train})$ , but their type sets are the same, *i.e.*,  $\mathcal{T}_{train'_1}, \mathcal{T}_{train'_2} = \mathcal{T}_{test}$ . Then, using each subset as training data, we fine-tune NER models (*i.e.*, ConNER and BioBERT) to perform inference on  $test$ . Please refer to Appendix A for experimental details and results of other subsets. In Table 5, VERIFINER brings a significant improvement on both models. For both BioBERT and ConNER, our approach is even comparable to the in-distribution setting that is trained on the target distribution. In particular, VERIFINER even outperforms BioBERT trained on the target dataset of GENIA by 14.59% on F1 score. This highlights VERIFINER’s ability to generalize without overfitting to the source dis-

tribution, as it verifies predictions based on reliable external knowledge sources.

### 5.3 Robustness of VERIFINER in Low-resource Scenarios (RQ3)

To assess how our framework can elevate robustness of NER models in low-resource scenarios, we investigate the performance changes of two fine-tuned models and those combined with VERIFINER, while varying the scale of training data for fine-tuning. For this, we generate training datasets of varying sizes, by randomly sampling examples from the original dataset across a range from 5% to 100%. Figure 8 illustrates that post-hoc verification of VERIFINER leads to notable enhancements in the performance of both BioBERT and ConNER across all low-resource scenarios examined in both datasets. Specifically, the gap between the fine-tuned models and those with VERIFINER becomes large as the number of training examples decreases. Furthermore, VERIFINER consistently achieves high precision irrespective of the number of training examples. In conclusion, VERIFINER is robust in diverse low-resource scenarios within the biomedical domain, where the scarcity of available datasets has remained as a long-standing challenge.

## 6 Related Work

**Named Entity Recognition.** A conventional approach for solving NER task is to predict probability distribution of entity types and sequentially label each token (Jeong and Kang, 2023; Lee et al., 2019). In domain-specific NER, data scarcity challenge has been a consistent challenge. Recent works attempt to solve the issue with data augmentation (Chen et al., 2021; Zhou et al., 2022). Other works mitigate the challenge with domain adaptation methods (Zhang et al., 2021; Yang et al., 2022; Xu et al., 2023). In most recent years, with the advent of LLMs, a line of work reformulate task definition of NER in a generative manner to leverage LLMs’ capabilities in NER tasks. Ashok and Lipton (2023) prompts an LLM to produce a list of potential entities with explanations that justify their compatibility with the provided entity type definitions. Wang et al. (2023a) instructs an LLM to recognize entities by using special tokens to surround them.

**Knowledge-augmented LMs.** For knowledge augmentation, previous works (Guu et al., 2020; Yamada et al., 2020; Borgeaud et al., 2022)



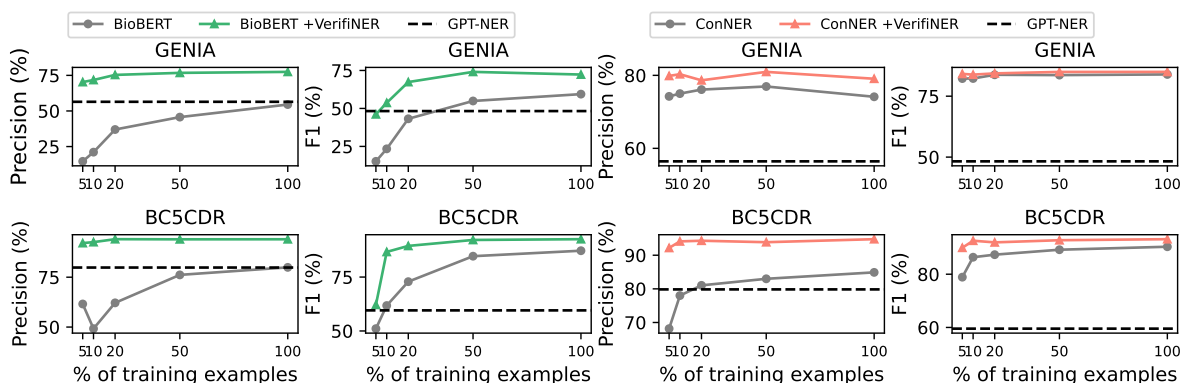


Figure 8: Precision and F1 scores of different NER models (w/ VERIFINER) in the low-resource settings.

show performance improvements on knowledge-intensive tasks. Zhang et al. (2019) utilize both large-scale corpora and knowledge graphs to train a language model and take full advantage of various knowledge simultaneously. Qin et al. (2021) proposes a novel contrastive learning framework to obtain a deep understanding of the entities and their relations in text. Similarly, later works (Lewis et al., 2020; Kang et al., 2022; Li et al., 2022; Izcard et al., 2022; Chae et al., 2023) augment LMs with external knowledge during fine-tuning. However, they lack in generalization ability across different domains, and require changing model architectures or additional training steps on each task and dataset.

Considering these challenges, recent methods (Lazaridou et al., 2022; Trivedi et al., 2022; Peng et al., 2023) use generalization capabilities of large or instruction-tuned LMs to incorporate external knowledge. Baek et al. (2023) first retrieve relevant facts from a knowledge graph and prepend the retrieved facts to the input question, which is then forwarded to LLMs to generate the answer. Zhao et al. (2023) post-edits reasoning chains according to external knowledge to increase prediction factuality.

**NER with External Knowledge.** There are a number of works that augment external knowledge on NER task. Nie et al. (2021) proposes a KB-aware NER framework which incorporates KB knowledge to via type projection to alleviate heterogeneity between KB knowledge and NER type scheme. Fu et al. (2023) proposes a dictionary-based method that recognizes biomedical entities from input text via synonym generalization. However, such dictionary-based methods yet fail to generalize over multiple domains.

## 7 Conclusion

In this paper, we present VERIFINER, a verification framework that identifies errors from existing NER methods and revises them into more faithful predictions via knowledge-grounded reasoning. It is the first attempt to solve NER with verification. Specifically, we introduce VERIFINER, a plug-and-play verification module which verifies the factuality and contextual relevance of entity with knowledge-grounded reasoning. Through extensive experiments, we demonstrate effectiveness of our framework over baselines. It is worth noting that VERIFINER is also robust in out-of-domain settings and low-resource scenarios with remarkable performance. Considering the feasibility of VERIFINER as a model-agnostic approach and its demonstrated generalization capabilities, we expect our work will enhance reliability in NER and be applicable to other domains in future research.

## Limitation

Despite the remarkable performance of VERIFINER, there is still room for improvements on our framework. First of all, we conduct experiments using gpt-3.5-turbo-1106 model from OpenAI for LLM. Considering the significance of LLM as a major component in the framework, it is yet to be discovered whether the effectiveness of VERIFINER is valid with other LLMs. Thus we leave application of our framework on different open and closed LLMs for future work.

In this work, we confine the scope of our work to biomedical domain. However, as a plug-and-play framework that can be applied to any model, VERIFINER has a potential on other knowledge-

intensive domains such as legal or scientific domains. In the future, we plan to investigate its application on other domains.

Lastly, our framework necessitates multiple inferences using LLMs, which can be computationally expensive. This aspect becomes particularly significant when integrating our framework into applications that require real-time inference. We believe this limitation can be resolved by using smaller models which are distilled from LLMs.

## Ethical Consideration

The main aspect of our work with the potential for ethical pitfalls is the use of LLMs for our framework. Recent work has highlighted the risks of LLMs in hallucination (Zhang et al., 2023). This problem might be more critical in biomedical NER, where prediction can be used for clinical decision support, drug discovery, and personalized medicine. Inaccurate or hallucinated information could lead to misdiagnoses, inappropriate treatment recommendations, or erroneous scientific research directions, ultimately posing a significant risk to patient safety and public health.

We argue here that this risk is largely mitigated in our work, mainly due to our verification process that incorporates knowledge from KBs. The goal of our work is reducing such errors with verification, specifically by integrating structured data and validated information from KBs. This integration allows for cross-referencing and validation of the outputs generated by the LLM, ensuring that any identified entities and associated information align with reliable biomedical knowledge.

## Acknowledgements

This work was supported by the IITP grant funded by the Korea government (MSIT) (No. RS-2020-II201361) and the NRF grant funded by the Korea government (MSIT) (No. RS-2023-00244689).

## References

Dhananjay Ashok and Zachary C Lipton. 2023. Prompter: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

O. Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research*, 32(90001):267D – 270.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Hyungjoo Chae, Yongho Song, Kai Ong, Taeyoon Kwon, Minjin Kim, Youngjae Yu, Dongha Lee, Dongyeop Kang, and Jinyoung Yeo. 2023. Dialogue chain-of-thought distillation for commonsense-aware conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5606–5632, Singapore. Association for Computational Linguistics.

Shuguang Chen, Gustavo Aguilar, Leonardo Neves, and Tamar Solorio. 2021. Data augmentation for cross-domain named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5346–5356, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Dai. 2021. Recognising biomedical names: Challenges and solutions. *arXiv preprint arXiv:2106.12230*.

Zihao Fu, Yixuan Su, Zaiqiao Meng, and Nigel Collier. 2023. Biomedical named entity recognition via dictionary-based synonym generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14621–14635, Singapore. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *CoRR*, abs/2208.03299.

- Minbyul Jeong and Jaewoo Kang. 2023. [Consistency enhancement of model prediction on document-level named entity recognition](#). *Bioinformatics*, 39(6).
- Minki Kang, Jinheon Baek, and Sung Ju Hwang. 2022. [KALA: knowledge-augmented language model adaptation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5144–5167. Association for Computational Linguistics.
- Md Rezaul Karim, Tanhim Islam, Md Shajalal, Oya Beyan, Christoph Lange, Michael Cochez, Dietrich Rebholz-Schuhmann, and Stefan Decker. 2023. [Explainable ai for bioinformatics: Methods, tools and applications](#). *Briefings in bioinformatics*, 24(5):bbad236.
- J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. [Genia corpus—a semantically annotated corpus for biotextmining](#). *Bioinformatics*, 19(suppl1):i180–i182.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.
- Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. [Internet-augmented language models through few-shot prompting for open-domain question answering](#). *arXiv preprint arXiv:2203.05115*.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [Biobert: a pre-trained biomedical language representation model for biomedical text mining](#). *Bioinformatics*, 36(4):1234–1240.
- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Sanjiv Kumar. 2022. [Large language models with controllable working memory](#). *arXiv preprint arXiv:2211.05110*.
- Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wieggers, and Zhiyong Lu. 2016. [Biocreative v cdr task corpus: a resource for chemical disease relation extraction](#). *Database*, 2016:baw068.
- Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. [Towards improving neural named entity recognition with gazetteers](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy. Association for Computational Linguistics.
- Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. [Coarse-to-Fine Pre-training for Named Entity Recognition](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Online. Association for Computational Linguistics.
- Andrei Mikheev, Marc Moens, and Claire Grover. 1999. [Named entity recognition without gazetteers](#). In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8.
- Binling Nie, Ruixue Ding, Pengjun Xie, Fei Huang, Chen Qian, and Luo Si. 2021. [Knowledge-aware named entity recognition with alleviating heterogeneity](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13595–13603.
- OpenAI. 2023. [Chatgpt](https://openai.com/blog/chatgpt). <https://openai.com/blog/chatgpt>.
- Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. [Check your facts and try again: Improving large language models with external knowledge and automated feedback](#). *arXiv preprint arXiv:2302.12813*.
- Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. [ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3350–3363. Association for Computational Linguistics.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions](#). *arXiv preprint arXiv:2212.10509*.
- 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*.

- Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. [Improving named entity recognition by external context retrieving and cooperative learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1800–1812, Online. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Jingyun Xu, Changmeng Zheng, Yi Cai, and Tat-Seng Chua. 2023. [Improving named entity recognition via bridge-based domain adaptation](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3869–3882, Toronto, Canada. Association for Computational Linguistics.
- Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. [LUKE: deep contextualized entity representations with entity-aware self-attention](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.
- Linyi Yang, Lifan Yuan, Leyang Cui, Wenyang Gao, and Yue Zhang. 2022. [FactMix: Using a few labeled in-domain examples to generalize to cross-domain named entity recognition](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5360–5371, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Tao Zhang, Congying Xia, Philip S. Yu, Zhiwei Liu, and Shu Zhao. 2021. [PDALN: Progressive domain adaptation over a pre-trained model for low-resource cross-domain named entity recognition](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5441–5451, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. [Siren’s song in the ai ocean: a survey on hallucination in large language models](#). *arXiv preprint arXiv:2309.01219*.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. [ERNIE: enhanced language representation with informative entities](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.
- Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. [Verify-and-edit: A knowledge-enhanced chain-of-thought framework](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.
- Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. [MELM: Data augmentation with masked entity language modeling for low-resource NER](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.



## A Implementation Details

**GENIA Preprocessing.** Although GENIA is a nested NER benchmark, we choose GENIA to investigate the performance of the flat NER models because 1) we wanted to validate the robustness of our framework on datasets with more labels than BC5CDR, and 2) there are a limited number of available biomedical NER datasets. To fine-tune GENIA on models that are trained for flat NER, we transform GENIA into a flat NER task by separating a single instance with nested labels into multiple individually labeled instances.

**Dataset Statistics and Sampling.** Due to the budget constraints, we use randomly sampled test set from both GENIA and BC5CDR for all experiments. We randomly sample 500 and 100 documents from GENIA and BC5CDR, respectively. Then, to verify each entity, we reconstruct context documents into entity-level input context. Specifically, we divide document per entity and verify each. We provide the detailed dataset statistics in Table 6, and detailed error ratio of the models in 7.

Dataset	Domain	Types	Split	#Document	#Entities
GENIA	biomedical	protein, RNA, DNA, cell_type, cell_line	train test	16615 500	44488 1472
BC5CDR	biomedical	Chemical, Disease	train test	500 100	9366 1831

Table 6: Dataset Statistics

Model	Type	Span	Type&Span	Spurious	FP	FN
ConNER	10.85	32.77	8.51	38.94	91.1	8.9
BioBERT	2.39	33.61	2.9	22.2	61.1	38.9
GPT-NER	4.22	11.71	4.92	15.14	64	36

Table 7: Ratio of error types of NER models.

**Baselines.** To apply VERIFINER, we consider two groups of NER models as baselines: the fine-tune based NER model (*i.e.* ConNER and BioBERT), and prompting-based LLM (*i.e.* GPT-NER). In case of ConNER and GPT-NER, we follow the official implementation of (Jeong and Kang, 2023) and (Wang et al., 2023a), respectively. In case of BioBERT (Lee et al., 2019), we use the checkpoint of pre-trained model from huggingface<sup>4</sup>. We train BioBERT on both datasets (*i.e.* GENIA and BC5CDR) for 20 epochs with learning rate of 3e-5.

<sup>4</sup><https://huggingface.co/dmis-lab/biobert-v1.1>

**KB Augmentation.** Inspired from the training process of language models that learn reliable knowledge from the train set, we add entities from the train set to KB as reliable knowledge. It is a fair setting because the entities are not directly assigned as the answer because not only the exact span, but also other multiple spans are given as candidates, and LLM has to choose the answer among them.

In order to set a scenario where all gold entities exist completely in the knowledge source, we augment the knowledge base (KB) using entities present in the dataset. First, we search all possible candidate spans extracted from initial prediction to the UMLS database. We then add these results to the dictionary if the results were found from the database. Subsequently, for entities that could not be found in the search, we additionally search within the annotated entity set present in each dataset (GENIA, BC5CDR) and add them to the dictionary. Through this process, we create a complete external knowledge source containing over 90% of the knowledge from the gold entities.

It is worth noting that our framework is applicable even when not all knowledge is in the KB or when new entities emerge, because as long as there exists a KB that contains knowledge of the query entity (no matter if the KB is complete or not) and we can retrieve knowledge, VerifNER is able to utilize knowledge and post-edit NER errors.

**Manual Mapping.** We provide the statistics of the number of semantic types assigned to each label.

Chemical	Disease	Total
13	15	28

Table 8: Distribution of semantic types mapped into BC5CDR labels.

Protein	Cell Type	DNA	Cell Line	RNA	Total
27	10	18	8	5	68

Table 9: Distribution of semantic types mapped into GENIA labels.

## B Experimental Settings

**Unseen Distribution Settings.** We fine-tune CONNER and BIOBERT using the training sets of the GENIA and BC5CDR datasets, respectively. Then using the checkpoints from each trained

Source → Target	GENIA' → GENIA			BC5CDR' → BC5CDR		
	P	R	F	P	R	F
GPT-NER	56.44	42.15	48.26	79.84	47.48	59.55
CONNER	74.38	59.57	66.16	75.73	84.85	80.03
+ VERIFINER	<b>78.86</b>	<b>59.8</b>	<b>68.02</b>	<b>74.16</b>	90.03	<b>81.48</b>
BIOBERT	15.16	16.12	15.62	46.45	55.49	50.57
+ VERIFINER	<b>78.27</b>	<b>38.77</b>	<b>51.85</b>	<b>93.33</b>	<b>61.92</b>	<b>74.45</b>

Table 10: Experimental results on distribution shift settings.

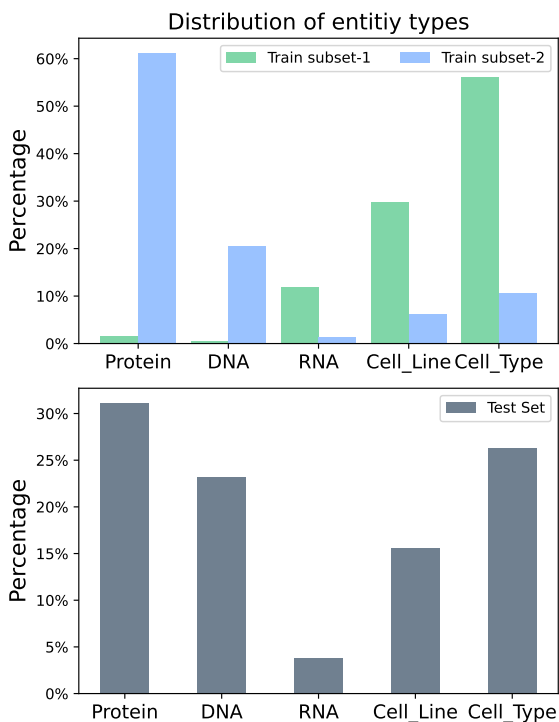


Figure 9: Label distribution of splitted GENIA train subsets and test set for shifted distribution experiments.

model, we cross-infer on the test set of remaining dataset, which is not used during train phase.

When applying VERIFINER to the inferred predictions, we adjust the initial predictions by a threshold to the logits of the model output. This step is implemented to mitigate the occurrence of an excessive number of "O" tags in the generated output, which could otherwise lead to a scarcity of entity input to the verification. Specifically, we compute the output logit percentile for the "O" tags across all instances. Then, for the top 10% of "O" tokens, we retain the original logits, while for the rest, we multiply the logits by -9999. This adjustment ensure that only predictions confidently classified as "O" tags remained, while the rest are assigned as entities for the input of verification.

**Shifted Distribution Settings.** To assume the scenarios where the target distributions differ from the source distributions, we manually split both datasets into two distinct subsets. We then fine-tune CONNER and BIOBERT using each subset as a source and inferred on each test set. The distribution of each subset and the experimental results for subset 2 are presented in Figure 9 and Table 10, respectively.

**Computational Resources and API Cost.** We run ConNER and BioBERT on eight NVIDIA RTX A5000 GPUs. For ChatGPT API usage, we use \$ 470 in total.

## C Case Study

We select representative examples of verification result by VERIFINER from span error case in GENIA and t&s error case in BC5CDR.

Table 11 presents a span error verification from GENIA. The two candidates are very similar with each other, yet VERIFINER manages to distinguish the answer between the two. Specifically, VERIFINER selects the gold annotation "*human lymphocytes*" as the final answer, based on the evidence that "*human lymphocytes*" narrows down the scope to human cells specifically.

Table 12 shows a type & span error verification result from BC5CDR. It is evident that the initial prediction, "*bupivacaine arrhythmogenicity*" is not in the candidate list. This indicates that the NER model has predicted an invalid entity that is not registered in the KB. Luckily, VERIFINER is able to filter out such invalid entities by searching the candidates in KB and verify their correctness with using knowledge as evidence.

## D Prompts

We provide type factuality verification prompts in Table 13, 14, and contextual relevance verification prompts in Table 15, 16. Note that we use individual prompt for GENIA and BC5CDR respectively.

<b>Example of Span Error Correction from GENIA</b>	
Initial prediction	('lymphocytes', 'cell_type')
Context	At 10 microM, both compounds inhibited IL-2 mRNA and protein levels in the NFAT-1-linked lac-Z transfectants, and in human lymphocytes.
Candidate pairs	<p><b>[Pair 1]</b>  Candidate: 'lymphocytes'  Knowledge:  Evidence: 'Based on the given knowledge, the semantic type of lymphocytes is 'cell_type'. Lymphocytes are a type of white blood cell that plays a crucial role in the immune system. They are responsible for recognizing and attacking foreign substances in the body. In the given paragraph, lymphocytes are mentioned in the context of inhibiting IL-2 mRNA and protein levels. IL-2 is an important cytokine involved in immune responses, and its inhibition suggests a potential role of lymphocytes in regulating immune activity. Therefore, based on the semantic type and the context, the class of the entity lymphocytes is cell_type.'</p> <p><b>[Pair 2]</b>  Candidate: 'human lymphocytes'  Knowledge:  Evidence: 'Based on the given knowledge, the semantic type of human lymphocytes is 'cell_type'. Lymphocytes are a type of white blood cell that plays a crucial role in the immune system. In the given paragraph, human lymphocytes are mentioned in the context of inhibiting IL-2 mRNA and protein levels. This aligns with the role of lymphocytes in immune responses and their involvement in cytokine production. Therefore, the class of the entity 'human lymphocytes' is cell_type.'</p>
Reasoning paths	<p><b>[Reasoning Path 1]</b>  Based on the provided explanations, both candidate entities 'lymphocytes' and 'human lymphocytes' are described as types of white blood cells that play a crucial role in the immune system. They are both mentioned in the context of inhibiting IL-2 mRNA and protein levels, which aligns with their involvement in immune responses and cytokine production.  However, the specific mention of "human lymphocytes" in the paragraph suggests a focus on human cells specifically. While lymphocytes in general can include various types of cells, <b>the mention of "human lymphocytes" narrows down the scope to human cells specifically.</b> This specificity makes "human lymphocytes" the more appropriate and specific entity in the given biomedical context.  Therefore, the final answer is ('human lymphocytes', cell_type).</p> <p><b>[Reasoning Path 2]</b>  Based on the provided explanations, both candidate entities 'lymphocytes' and 'human lymphocytes' are described as a type of white blood cell that plays a crucial role in the immune system. They are both mentioned in the context of inhibiting IL-2 mRNA and protein levels. Although the term 'human lymphocytes' provides more specificity, it is important to note that the given knowledge does not provide any additional information distinguishing between lymphocytes and human lymphocytes. Therefore, it is not possible to definitively determine which term is the most reliable entity in this context. Therefore, the final answer is (None, None).</p> <p><b>[Reasoning Path 3]</b>  ...</p>
Votes	('human lymphocytes', 'cell_type') 8 ('lymphocytes', 'cell_type') 1 (None, None) 1
Corrected prediction	('human lymphocytes', 'cell_type')
Gold annotation	('human lymphocytes', 'cell_type')

Table 11: Example from GENIA, span error is corrected.

<b>Example of Type&amp;Span Error Correction from BC5CDR</b>	
Initial prediction	('bupivacaine arrhythmogenicity', 'Disease')
Context	Progesterone potentiation of bupivacaine arrhythmogenicity in pentobarbital-anesthetized rats and beating rat heart cell cultures.
Candidate pairs	<p>('arrhythmogenicity', 'Arrhythmogenicity is a congenital cardiomyopathy characterized by the infiltration of adipose and fibrous tissue into the right ventricle wall and the loss of myocardial cells. It primarily affects the free wall of the right ventricle and right atria, resulting in ventricular and supraventricular arrhythmias. Based on this definition and the semantic type of 'arrhythmogenicity' is 'Disease or Syndrome'.')</p> <p>('bupivacaine', 'Bupivacaine is a widely used local anesthetic agent. It falls under the semantic type of 'Organic Chemical' and 'Pharmacologic Substance'. Based on this information, 'bupivacaine' can be classified as 'Chemical'.')</p>
Reasoning paths	<p><b>[Reasoning Path 1]</b> Based on the given knowledge, 'bupivacaine' is a widely used local anesthetic agent and is classified as a Chemical entity. On the other hand, 'arrhythmogenicity' is a congenital cardiomyopathy characterized by the infiltration of adipose and fibrous tissue into the right ventricle wall and the loss of myocardial cells. It primarily affects the free wall of the right ventricle and right atria, resulting in ventricular and supraventricular arrhythmias. Based on this definition and the semantic type of 'Disease or Syndrome', arrhythmogenicity can be classified as a disease. <b>Comparing these entities, 'bupivacaine' is directly related to the context as a local anesthetic agent</b>, and it satisfies the guideline of annotating the most specific chemical mentions. Therefore, the most appropriate answer entity is 'bupivacaine'.</p> <p><b>[Reasoning Path 2]</b> Based on the given knowledge, both 'arrhythmogenicity' and 'bupivacaine' are classifiable entities. 'Arrhythmogenicity' is a congenital cardiomyopathy characterized by the infiltration of adipose and fibrous tissue into the right ventricle wall and the loss of myocardial cells. It primarily affects the free wall of the right ventricle and right atria, resulting in ventricular and supraventricular arrhythmias. Based on this definition and the semantic type of 'Disease or Syndrome', arrhythmogenicity can be classified as a disease. On the other hand, 'bupivacaine' is a widely used local anesthetic agent and falls under the semantic type of 'Organic Chemical' and 'Pharmacologic Substance', indicating that it is a chemical. Comparing the two entities, they represent distinct classes of entities based on their definitions and semantic types. Therefore, the final answer is ('arrhythmogenicity', Disease).</p> <p><b>[Reasoning Path 3]</b> ...</p>
Votes	('bupivacaine', 'Chemical') 9 ('arrhythmogenicity', 'Disease') 1
Corrected prediction	('bupivacaine', 'Chemical')
Gold annotation	('bupivacaine', 'Chemical')

Table 12: Example from BC5CDR, type&span error is corrected.



---

### Type Factuality Verification Prompt (GENIA)

---

#### [Task Description]

You are solving named entity recognition task for biomedical articles. There is an entity to which you need to assign either one of the six classes : protein, DNA, RNA, cell\_line, cell\_type or None. Class None is assigned when an entity does not belong to any of the five classes: protein, DNA, RNA, cell\_line, and cell\_type. I will give you an entity and reliable knowledge about the entity for reference. There are two types of knowledge: definition and semantic type of entities you need to predict. Identify the class of the entity and explain why. When generating explanation, you need to quote the knowledge. If both definition and semantic type are not provided, use context information of entity in Paragraph or your own knowledge to explain. Remember, you should assign at least one class for each entity from above six (protein, DNA, RNA, cell\_line, cell\_type or None). Using Example 1 to 6 as reference, answer Example 7.

#### [Example 1]

**Entity:** GRalpha

**Knowledge about this entity:** The definition of GRalpha is not provided. The semantic type of GRalpha is ['Amino Acid, Peptide, or Protein', 'Receptor'].

**Explanation:** Based on the given knowledge, the semantic type of GRalpha is 'Amino Acid, Peptide, or Protein' and 'Receptor'. Proteins are large molecules that perform various functions in the body, including acting as receptors for hormones and other signaling molecules. In this case, GRalpha is described as a hormone-activated transcription factor, which aligns with its role as a protein receptor. Therefore, the class of the entity GRalpha is protein.

#### [Example 2]

**Entity:** neutrophils

**Knowledge about this entity:**

The definition of neutrophils is ['Granular leukocytes having a nucleus with three to five lobes connected by slender threads of chromatin, and cytoplasm containing fine inconspicuous granules and stainable by neutral dyes.']. The semantic type of neutrophils is ['Cell'].

**Explanation:**

Based on the given knowledge, the definition of neutrophils is 'Granular leukocytes having a nucleus with three to five lobes connected by slender threads of chromatin, and cytoplasm containing fine inconspicuous granules and stainable by neutral dyes.' This definition indicates that neutrophils are a type of cell, specifically a type of granular leukocyte. Therefore, the class of the entity neutrophils is cell\_type.

#### [Example 3] ...

---

Table 13: The prompt for type factuality verification on GENIA (Examples 3 to 6 are omitted in this table).

---

**Type Factuality Verification Prompt (BC5CDR)**

---

**[Task Description]**

You are solving named entity recognition task for biomedical articles. There is an entity to which you need to assign either one of the three classes: Chemical or Disease or None. You should only use one of the three classes. No other class exists. I will give you an entity and reliable knowledge about the entity. There are two types of knowledge: definition and semantic type. Identify the class of the entity and explain why. When generating explanation, you need to use the given knowledge. If you are not certain with your decision, you may explain why you are not certain. Even if it does not fit to any of the classes, try to assign the most related one.

**[Example 1]****Paragraph:**

**Entity:** chest pain

**Knowledge about this entity:** The definition of chest pain is ['Pressure, burning, or numbness in the chest.']. The semantic type of chest pain is ['Sign or Symptom'].

**Explanation:** Chest pain is a symptom characterized by pressure, burning, or numbness in the chest. Based on the provided knowledge, chest pain falls under the semantic type of 'Sign or Symptom'. Therefore, the class of the entity chest pain is Disease.

**[Example 2]**

**Entity:** amoxicillin

**Knowledge about this entity:**

The definition of amoxicillin is ['A broad-spectrum semisynthetic antibiotic similar to AMPICILLIN except that its resistance to gastric acid permits higher serum levels with oral administration.']. The semantic type of amoxicillin is ['Organic Chemical', 'Antibiotic'].

**Explanation:**

amoxicillin is a broad-spectrum semisynthetic antibiotic that is similar to AMPICILLIN. It has the ability to resist gastric acid, allowing for higher serum levels with oral administration. Additionally, it is classified as an organic chemical and an antibiotic. Based on this information, it can be concluded that amoxicillin belongs to the class of Chemical. Therefore, the class of the entity amoxicillin is Chemical.

**[Example 3] ...**

---

Table 14: The prompt for type factuality verification on BC5CDR (Examples 3 to 6 are omitted in this table).

---

**Contextual Relevance Verification Prompt (GENIA)**

---

**[Task Description]**

You are solving named entity recognition task for biomedical articles. You will be provided pairs of candidate entities and explanations of why each entity is labeled such, along with a context in which the candidate entities appear. The entities are very similar, and your job is to choose a pair that has the most reliable explanation, and is most likely to be a biomedical entity considering the context. Think Step-by-step and explain why you think that entity is the answer and compare your answer entity to other candidate entities. Using Example 1 to 8 as reference, answer Example 9.

**[Example 1]****Pair of candidate entity and explanation:**

['proto-oncogene c-rel', 'Since the definition and semantic type of proto-oncogene c-rel are not provided, we can use our own knowledge to determine the class of the entity. Proto-oncogenes are genes that have the potential to become oncogenes, which are genes that can cause cancer when mutated or overexpressed. In this case, proto-oncogene c-rel is mentioned in the context of a DNA probe that spans a conserved domain among other genes, including the p50 DNA binding subunit of NF-kappa B. NF-kappa B is a transcription factor that plays a key role in regulating genes involved in immune responses and cell survival. The fact that proto-oncogene c-rel is mentioned in relation to DNA binding suggests that it may also be involved in gene regulation. Therefore, based on our knowledge, the class of the entity proto-oncogene c-rel is protein.']

['c-rel', 'In the given paragraph, c-rel is mentioned as a proto-oncogene that is conserved among other genes, including the p50 DNA binding subunit of NF-kappa B. Proto-oncogenes are genes that have the potential to cause cancer when mutated or overexpressed. In this case, c-rel is described as a proto-oncogene, indicating its role in cancer development. Therefore, the class of the entity c-rel is protein.']

**Your decision:**

Based on the provided explanation, 'proto-oncogene c-rel' links it to a conserved domain among other genes, including the p50 DNA binding subunit of NF-kappa B. The explanation goes on to connect proto-oncogene c-rel to the role of proto-oncogenes in cancer development, suggesting its involvement in gene regulation. This detailed explanation provides a broader understanding of the entity in the given biomedical context. In comparison, the other candidate entity 'c-rel' is less specific in its explanation, merely stating that c-rel is a proto-oncogene without providing the additional context about its conserved domain and its association with the p50 DNA binding subunit of NF-kappa B. The chosen entity, 'proto-oncogene c-rel,' is more comprehensive and aligns better with the information provided in the paragraph.

Therefore, the final answer is ('proto-oncogene c-rel', protein).

**[Example 2]****Pair of candidate entity and explanation:**

['CBF', 'Based on the given knowledge, the semantic type of CBF is protein. Proteins are large molecules that perform various functions in the body. In the given paragraph, CBF is described as consisting of two subunits, CBF alpha and CBF beta, with CBF alpha being a DNA binding subunit. This aligns with the role of proteins as molecules that can bind to DNA. Therefore, based on the semantic type information and the context provided, the class of the entity CBF is protein.']

['CBF beta', 'Based on the given knowledge, the semantic type of CBF beta is protein. In the given paragraph, CBF beta is described as a subunit that stimulates the DNA binding activity of CBF alpha. Therefore, the class of the entity CBF beta is protein.']

['beta', 'Based on the given knowledge, the semantic type of beta is 'protein'. Proteins are large molecules that perform various functions in the body. In this case, 'beta' is mentioned as a subunit that stimulates the DNA binding activity of CBF alpha, indicating its role as a protein. Therefore, the class of the entity 'beta' is protein.']

**Your decision:**

Based on the provided explanations, candidate 'CBF beta' is the subunit of CBF, but 'CBF beta' is specifically highlighted as a subunit that stimulates the DNA binding activity of CBF. The explanation for 'CBF beta' explicitly assigns it the semantic type of protein, aligning with the general role of proteins in the body. It is a specific subunit with a clear role in stimulating DNA binding activity, and its semantic type aligns with the expected classification of proteins.

Therefore, the final answer is ('CBF beta', protein).

**[Example 3] ...**

---

Table 15: The prompt for contextual relevance verification on GENIA (Examples 3 to 8 are omitted in this table).

---

**Contextual Relevance Verification Prompt (BC5CDR)**

---

**[Task Description]**

You are solving named entity recognition task for biomedical articles. You will be provided pairs of candidate entities and explanations of why each entity is labeled such, along with a context in which the candidate entities appear. The entities are very similar, and your job is to choose a pair that has the most reliable explanation, and is most likely to be a biomedical entity considering the context. Explain why you think that entity is the answer and compare your answer entity to other candidate entities. Use Example 1 to 20 as reference.

**[Example 1]****Pair of candidate entity and explanation:**

['hepatitis', 'hepatitis is defined as inflammation of the liver. Based on this definition, it can be classified as a disease. Additionally, the semantic type of hepatitis is also 'Disease or Syndrome'. Therefore, the entity hepatitis should be classified as a Disease.']

['fulminant hepatitis', 'Since there is no definition available for fulminant hepatitis, I will rely solely on the semantic type. The semantic type of fulminant hepatitis is 'Disease or Syndrome', indicating that it is a disease. Therefore, I would classify fulminant hepatitis as a Disease.']

**Your decision:**

Based on the given knowledge, hepatitis is defined as inflammation of the liver, which falls under the category of a disease. Additionally, the semantic type of hepatitis is also 'Disease or Syndrome'. Therefore, the entity hepatitis should be classified as a Disease. On the other hand, there is no specific definition available for fulminant hepatitis. However, the semantic type of fulminant hepatitis is 'Disease or Syndrome', indicating that it is a disease. Therefore, both entities can be classified as Disease. However, 'fulminant hepatitis' provides additional information about the severity of the hepatitis, indicating a rapid and severe progression. This makes it a more specific mention compared to 'hepatitis'.

Therefore, the final answer is ('fulminant hepatitis', Disease).

**[Example 2]****Pair of candidate entity and explanation:**

['haloperidol', 'The definition of haloperidol is 'A phenyl-piperidiny-1-butylpiperone that is used primarily to treat SCHIZOPHRENIA and other PSYCHOSES. It is also used in schizoaffective disorder, DELUSIONAL DISORDERS, ballism, and TOURETTE SYNDROME (a drug of choice) and occasionally as adjunctive therapy in INTELLECTUAL DISABILITY and the chorea of HUNTINGTON DISEASE. It is a potent antiemetic and is used in the treatment of intractable HICCUPS. (From AMA Drug Evaluations Annual, 1994, p279) The semantic type of haloperidol is 'Organic Chemical', 'Pharmacologic Substance'.']

**Your decision:**

haloperidol is an organic chemical and pharmacologic substance that is primarily used to treat schizophrenia and other psychoses. It is also used in schizoaffective disorder, delusional disorders, ballism, Tourette syndrome, and occasionally as adjunctive therapy in intellectual disability and the chorea of Huntington disease. It is a potent antiemetic and is used in the treatment of intractable hiccups. Based on the provided knowledge, haloperidol is classified as a pharmacologic substance.

Therefore, the final answer is ('haloperidol', Chemical).

**[Example 3] ...**

---

Table 16: The prompt for contextual relevance verification on BC5CDR (Examples 3 to 20 are omitted in this table).



---

**LLM-revision Prompt (GENIA)**

---

**[Task Description]**

Within the GENIA dataset consisting of PubMed articles, the sentences are tagged with labels. The task is to verify whether the word is properly tagged with the correct label from the given sentence and revise it. If the entity is annotated incorrectly, you may revise tags. Here's what each label means: <P> is 'protein', <D> is 'DNA', <R> is 'RNA', <CL> is 'cell\_line', <CT> is 'cell\_type'. If the entity corresponds to Protein, <P> should be added before the entity and </P> after it. Follow the format of Example 1 to answer Example 2.

**[Example 1]****The given sentence:**

Whereas activation of the <D>HIV-1 enhancer</D> following T-cell stimulation is mediated largely through binding of the transcription factor <P>NF-kappa B</P> to two adjacent <D>kappa B sites</D> in the <R>HIV-1</R> long terminal repeat , activation of the <D>HIV-2 enhancer</D> in <CT>monocytes</CT> and <CT>T cells</CT> is dependent on four <D>cis-acting elements</D> : a single kappa B site , two purine-rich binding sites , <D>PuB1</D> and <D>PuB2</D> , and a pets site .

**The labeled sentence:**

Whereas activation of the <D>HIV-1 enhancer</D> following T-cell stimulation is mediated largely through binding of the <P>transcription factor</P> <P>NF-kappa B</P> to two adjacent <D>kappa B sites</D> in the <D>HIV-1 long terminal repeat</D> , activation of the <D>HIV-2 enhancer</D> in <CT>monocytes</> and </CT>T cells</> is dependent on four <D>cis-acting elements</D> : a single <D>kappa B site</D> , two purine-rich binding sites , <D>PuB1</D> and <D>PuB2</D> , and a <D>pets site</D> .

**[Example 2]****The given sentence:**

{context}

**The labeled sentence:**

---

Table 17: The prompt for LLM-revision on GENIA.

---

**LLM-revision Prompt (BC5CDR)**

---

**[Task Description]**

Within the BC5CDR dataset consisting of PubMed articles, the sentences are tagged with labels. The task is to verify whether the word is properly tagged with the correct label from the given sentence and revise it. If the entity is annotated incorrectly, you may revise tags. Here's what each label means: <C> is for 'Chemical', <D> is for 'Disease'. If the entity corresponds to Disease, <D> should be added before the entity and </D> after it. Follow the format of Example 1 to answer Example 2.

**[Example 1]****The given sentence:**

Recurarization in the recovery room . A case of recurarization in the recovery room is reported . Accumulation of <C>atracurium</C> in the intravenous line led to recurarization after flushing the line in the recovery room . A <D>respiratory arrest</D> with severe desaturation and bradycardia occurred . Circumstances leading to this event and the mechanisms enabling a neuromuscular blockade to occur , following the administration of a small dose of relaxant , are discussed .

**The labeled sentence:**

Recurarization in the recovery room . A case of recurarization in the recovery room is reported . Accumulation of atracurium in the intravenous line led to recurarization after flushing the line in the recovery room . A respiratory arrest with severe desaturation and bradycardia occurred . Circumstances leading to this event and the mechanisms enabling a <D>neuromuscular blockade</D> to occur , following the administration of a small dose of relaxant , are discussed .

**[Example 2]****The given sentence:**

{context}

**The labeled sentence:**

---

Table 18: The prompt for LLM-revision on BC5CDR.