Treating Decoder-Only LLMs as Encoders: A Simple and Effective Fine-tuning Approach for Named Entity Recognition

Ken Yano and Hiroya Takamura
National Institute of Advanced Industrial Science and Technology
Artificial Intelligence Research Center
{yano.ken, takamura.hiroya}@aist.go.jp

Abstract

NER requires token-level classification using both left and right context, which makes encoder-only models like BERT naturally well-suited for the task. Decoder-only LLMs, by contrast, use causal masking during training, so their token representations lack right-side context, limiting their effectiveness on structured prediction tasks like NER despite their strong general capabilities. To address this, the authors propose fine-tuning decoder-only LLMs with causal attention replaced by full attention, combined with label-supervised discriminative training. While similar ideas exist in prior work, those studies were limited in scope. This work evaluates seven LLMs across four model families (Gemma, Qwen2.5, Llama3.1, Llama3.2) and compares full fine-tuning against LoRA. Results show that the proposed approach with an appropriate LoRA configuration outperforms encoder baselines (BERT, RoBERTa, DeBERTa), and achieves strong NER performance without auxiliary data or architectural modifications, though it does not reach SOTA on BC5CDR and CoNLL2003.

1 Introduction

Named Entity Recognition (NER) in the biomedical domain is a foundational task for extracting structured knowledge from clinical and scientific text, enabling downstream applications such as drug-disease relationship extraction, clinical trial mining, and pharmacovigilance. Biomedical NER is traditionally treated as a sequence-labeling task using a tagging scheme such as BIO, enabling detection and classification of entity spans — such as diseases, chemicals, genes, and clinical procedures — via token classification. It is therefore crucial to correctly classify each token using the context before and after it, a particular challenge in biomedical text where entity boundaries are often ambiguous, and terminology is highly specialized.

Encoder-only models such as BioBERT, PubMedBERT, and DeBERTa are trained to predict the masked token given the surrounding context, making their learned token representations well-suited to token classification tasks such as biomedical NER. On the other hand, decoder-only LLMs are trained to generate text in an autoregressive manner using a causal mask. As a result, the learned token representation contains no information about the context to the right of each token, which is a structural disadvantage for span-level extraction tasks.

Nonetheless, decoder-only LLMs trained on massive amounts of text — including biomedical literature, clinical notes, and scientific corpora — can solve various tasks using an instruction prompt, with or without in-context examples, and demonstrate strong capabilities across a wide range of biomedical understanding tasks. However, this generative solution method has not shown sufficient strength for structure prediction tasks such as NER in both general and biomedical domains. In particular, in the biomedical domain, it requires precise extraction of entity spans — including nested and discontinuous mentions extraction common in clinical text — and accurate classification of fine-grained biomedical entity types such as chemical compounds, disease mentions, and genomic variants.

To address these issues, we propose a label-supervised discriminative LLM fine-tuning method that replaces causal attention with a full-attention mask during both training and inference, adapting decoder-only LLMs for NER. Similar methods have been proposed in previous work (Li et al., 2023; Dukić and Šnajder, 2024); however, they examined only a limited number of LLM model families of a few different sizes. Our method also partially overlaps with LLM2Vec (BehnamGhader et al., 2024), where LLMs were effectively transformed into encoders. Our NER method cannot

Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026), pages 312–325

handle nested and discontinuous entity mentions, which are common in the biomedical domain and require span-based approaches (Tan et al., 2020; Su et al., 2022; Wang et al., 2022), which are beyond the scope of this work.

In this work, we evaluated seven decoder LLMs with varying sizes (from 0.5B to 8B) from four model families — Gemma, Qwen2.5, Llama3.1, and Llama3.2 — on general and biomedical NER benchmarks. For the baseline encoders, we evaluated BERT, RoBERTa, and DeBERTa. We also examined how two different training approaches, full fine-tuning and LoRA, affect the performance of the proposed method in both general and biomedical settings, where labeled data are often scarce and costly to annotate. Our results show that label-supervised discriminative LLM fine-tuning with full self-attention outperforms all baseline encoders when fine-tuned with an appropriate LoRA configuration. Although the results of our proposed method did not reach state-of-the-art on two NER benchmarks, BC5CDR and CoNLL2003, our models still demonstrate strong performance even when fine-tuned solely on each benchmark's training set, without relying on auxiliary corpora, domain-adaptive pretraining, or additional model architecture modifications.

2 Related Work

2.1 Named Entity Recognition

NER is traditionally tackled as a sequence labeling task, and numerous methods have been proposed (Lample et al., 2016; Ma and Hovy, 2016). With the emergence of large-scale transformer-based pretrained models, fine-tuning has become the standard for achieving better performance (Peters et al., 2018; Devlin et al., 2019).

The major approaches using transformer-based models can be categorized as follows: (1) supervised fine-tuning using encoder-based models such as BERT (Devlin et al., 2019) and its variants (Liu et al., 2019), (2) supervised fine-tuning using encoder-decoder models such as T5 (Wang et al., 2023; Paolini et al., 2021), (3) few-shots in-context learning using LLMs (Ding et al., 2021; Monajatipoor et al., 2024; Wang et al., 2025), and (4) supervised fine-tuning using LLMs (Xu et al., 2025; Dukić and Šnajder, 2024; Li et al., 2023).

Another line of effort focuses on the structure prediction task for longer contexts, such as multi-document relation extraction (Tan et al., 2022; Xue et al., 2024; Chen et al., 2020; Amalvy et al., 2023).

2.2 Generative methods by LLMs

Even though decoder-only LLMs have achieved SOTA performances on a variety of NLP tasks, their performance on NER still falls short of the supervised encoder baselines. This is because NER is a sequence-labeling task, whereas decoder-only LLMs are text-generation models.

Although several LLM-based NERs have been proposed, they still perform subpar with supervised encoder models. For example, GPT-NER (Wang et al., 2025) claims to perform on par with the encoder model. However, their method has some limitations, such as the ability to extract only one entity type at a time and the need for many in-context examples to match the performance of supervised encoder models.

There are also issues with converting structured information, such as NER, to a serialized form generated by LLMs. Without guardrails, parsing errors are a significant concern.

To overcome these issues, several approaches exist for generating structured output from LLMs, including prompt engineering, constrained decoding, grammar-based generation, and tool-augmented generation. Even though structured output from LLMs helps mitigate parsing errors, it significantly impacts inference time.

2.3 Discriminative methods by LLMs

This paper is closely related to the work by Li et al. (2023) and Dukić and Šnajder (2024), where the causal attention of LLMs is replaced by full attention to solve structure prediction tasks. Dukić and Šnajder (2024) proposed selectively removing a causal attention mask by identifying some of the decoder layers, instead of all decoder layers, to yield the best performance. However, this search space is enormous if the number of decoder layers n is large.

To reduce the search space to a manageable size, Dukić and Šnajder (2024) proposed dividing the entire set of decoder layers into four layer groups, each with eight consecutive decoder layers, where n is 32. This leaves them $2^4 = 16$ possible search patterns to evaluate.

However, it remains unclear whether this consecutive grouping of an equal number of decoder layers is optimal for achieving the best NER performance. Moreover, their ablation studies did

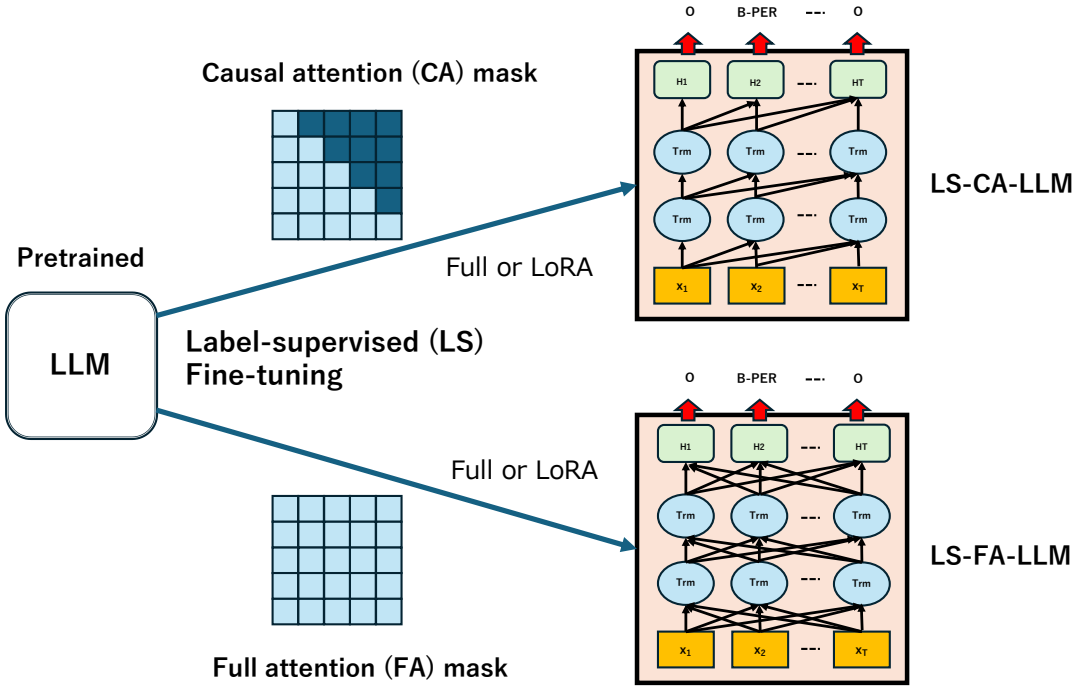

Figure 1: Label supervised (LS) LLM fine-tuning for NER task. The upper right figure shows the default LS fine-tuning with a causal attention mask. The lower-right figure shows the proposed LS fine-tuning, which replaces causal attention with a full attention mask across all decoder blocks. We denote the LS fine-tuned models using causal attention (CA) and full attention (FA) masks as LS-CA-LLM and LS-FA-LLM, respectively.

not identify consistent replacement patterns for decoder layers that yielded significantly better results across different decoder models and NER tasks. Hence, we replaced the causal attention mask with a full attention mask across all decoder layers.

3 Token Classification by Decoder-only LLM

We use LLMs not as generative models but as token classifiers to solve NER tasks, similar to encoder models, in which NER is performed by classifying each token into predefined BIO tags.

Token classification LLM models are defined by replacing the language modeling head with a token classification head initialized with pretrained LLM weights. The token classification head is initialized to random values before fine-tuning.

During fine-tuning, token classification LLMs are trained to classify each token using supervised labels. Here, the self-attention of original LLMs is defined using a causal attention mask, so the learned token representation contains no information about the context to the right of each token. This is a disadvantage, as both past and future token information is essential for classifying each token's label.

So our proposed method replaces the causal attention (CA) mask with a full attention (FA) mask across all decoder transformer blocks, ensuring that learned token representations capture context not only from past but also from future tokens, as shown in Fig. 1. We define label supervised (LS) fine-tuning of LLMs using CA and FA mask by LS-CA-LLM and LS-FA-LLM, respectively, and, for a specific LLM, we append **LS-CA-** and **LS-FA-** as prefixes to the name of LLMs in the following discussion to differentiate them.

Implementation of LS-FA-LLMs

We use HuggingFace Transformers to implement the proposed NER models. For a particular LLM model family, the corresponding token classification model is defined separately in the library. For instance, for Llama, the corresponding token classification model is defined as *LlamaForTokenClassification* in the library.

To replace the causal attention of LLMs, originally implemented for their corresponding token classifier models, such as *LlamaForTokenClassification*, with a full attention mask, we manually modified the source code. Currently, the source modification is ad hoc, meaning we need to manu-

ally change the source code. We leave, making the change systematic for future work.

4 Experimental Setup

4.1 NER datasets

We used two NER datasets for our experiments. One is BC5CDR (Wei et al., 2016) and the other is CoNLL2003 (Tjong Kim Sang and De Meulder, 2003). BC5CDR defines two entity types, chemical and disease, and CoNLL2003 defines four entity types, person, organization, location, and miscellaneous. Details of each dataset are provided in Table 1.

We also constrained our experimental conditions to avoid using any auxiliary datasets that would boost LLM performance. Hence, we train each model using only the corresponding training and validation datasets and evaluate its performance on the corresponding testing dataset.

4.2 Examined baseline encoders and decoder-only LLMs

For the baselines, we used the following encoder models in our experiments: BERT (bert-base-cased) (Devlin et al., 2019), RoBERTa (roberta-base) (Liu et al., 2019), and DeBERTa (deberta-v3-large) (He et al., 2021).

For the decoder models, we selected four LLM model families: Gemma, Qwen2.5, Llama3.2, and Llama3.1. For each model family, we selected one or two models with different model sizes.

Specifically, we evaluated seven LLMs across various model sizes. Gemma-2B and Gemma-7B (Team et al., 2024), Qwen2.5-0.5B and Qwen2.5-7B, Llama3.2-1B and Llama3.2-3B and Llama3.1-8B (Grattafiori et al., 2024).

State-of-the-art models

For BC5CDR, we evaluated BioBERT_v1.1 (Lee et al., 2019), which is specifically fine-tuned for the biomedical domain, on the same experimental setting as other models. In addition, we compare OpenMED's SOTA results (Panahi, 2025) for reference.

For CoNLL2003, we compare the SOTA results of ACE (Automated Concatenation of Embeddings, proposed by Wang et al. (Wang et al., 2021), fine-tuned on CoNLL2003, for reference.

4.3 Training of label-supervised sequence labeling

LS-FA-LLMs, LS-CA-LLMs, and the baseline encoder models were trained for sequence labeling by adding a token classification head on top of the last hidden layer. The token classification head, initialized to random values, was trained alongside the underlying model parameters during fine-tuning.

In our experiments, all models are trained without parameter-efficient fine-tuning (PEFT) technique and with LoRA (Hu et al., 2021). By these experiments, we can analyze the results in four ways: either CA or FA attention mask with either full or LoRA training.

By using LoRA, we can reduce the trainable parameters of LS-CA-LLMs and LS-FA-LLMs to a much smaller size, even compared with the full parameter sizes of encoder models, as shown in Table 4.

Moreover, using LoRA, we were able to fit all LS-CA-LLMs and LS-FA-LLMs during fine-tuning with a consistent device batch size of 8 on a single H200 GPU with 141 GB of memory. To fully train LS-CA-LLMs and LS-FA-LLMs over 7B, we use multiple H200 GPUs to avoid out-of-memory errors. The details of training parameters and LoRA configuration are in Appendix A.

We used the same default LoRA configuration for all training. In the later result section (5.2), we investigate how different LoRA configurations affect the results. We conducted additional experiments with different LoRA configurations for LS-FA-Llama3.2-3B and LS-FA-Llama3.1-8B, which performed better among other models.

4.4 Comparison with generative NER by instruction fine-tuned LLMs

To verify the effectiveness of our proposed method, we also evaluate NER performance using the same LLMs with instruction-fine-tuning (IFT) to solve the same NER tasks in a generative manner.

To fine-tune IFT LLMs, we used the same LoRA configuration and hyperparameters as those used to train LS-FA-LLMs. However, to prevent overfitting to the training dataset, we use early_stopping and stop training when the last 10 consecutive updates do not improve the evaluation loss.

For NER instruction, we adopt a format similar to Alpaca (Taori et al., 2023). The detailed instruction samples are provided in Appendix B. We use **IFT-** as the prefix to denote instruction-fine-tuned

Dataset	Train	Validation	Test	Entity Types
BC5CDR	5,228	5,330	5,865	Chemical, Disease
CoNLL2003	14,041	3,250	3,453	PER,ORG,LOC,MISC

Table 1: The statistics for the number of samples for BC5CDR (Wei et al., 2016) and CoNLL2003 (Tjong Kim Sang and De Meulder, 2003). PER, ORG, LOC, MISC specify person, organization, location, and miscellaneous, respectively.

LLMs such as IFT-Gemma-2B in the results.

5 Results

We first present the NER results by comparing baseline encoders with LS-CA-LLMs and LS-FA-LLMs, all of which are trained with full or default LoRA settings. Then we show performance variability across different LoRA configurations.

In the last section, we show comparative results between generative methods by IFT-LLMs and LS-FA-LLMS with baselines.

5.1 Comparison of baseline encoders and LS-CA-LLMs and LS-FA-LLMs

Table 2 shows precision, recall, and test micro F1 scores of BC5CDR by baseline encoders, LS-CA-LLMs, and LS-FA-LLMs. At the top, we show the SOTA results from OpenMed for reference, along with the results of domain-adapted BioBERT_v1.1. To evaluate sequence labeling results, we used seqeval. The first three columns are results of full training, and the last three columns are results of LoRA training. For each column, the best score is indicated in **bold** font. The scores with a yellow background specify the best score across full and LoRA training. In both full and LoRA training, LS-FA-LLM models significantly outperform the corresponding LS-CA-LLM models by 5-15 points, indicating that the causal attention mask is the main cause of low performance on NER tasks.

When trained with LoRA, LS-CA-LLMs and LS-FA-LLMs consistently outperform fully trained models, indicating that LoRA training is highly effective for fine-tuning LLMs with relatively small training datasets such as BC5CDR.

On the other hand, we cannot confirm a similar performance gain of LoRA over full training for baseline encoders, especially for BERT and RoBERTa. We hypothesize that LoRA training is effective only when the training dataset is not large enough relative to the model's capacity, in which case full training may lead to overfitting.

We confirm that the results of LS-FA-Llama3.2-3B and LS-FA-Llama3.1-8B, trained with LoRA, outperform all baseline encoders and the domain-adapted BioBERT_v1.1, excluding SOTA f1 results by OpenMed (Panahi, 2025).

Similarly, Table 3 shows precision, recall, and test micro F1 scores of CoNLL2003 by baseline encoders, LS-CA-LLMs, and LS-FA-LLMs. At the top, we present the SOTA f1 result by ACE+Fine-tune (Wang et al., 2021) for reference.

We see similar trends to those seen in the previous BC5CDR results in Table 2. Excluding SOTA f1 by ACE+Fine-tune, LS-FA-Llama3.2-3B and LS-FA-Llama3.1-8B, trained with LoRA, outperform all the baseline encoders in all metrics.

5.2 Variability of performance on different LoRA configurations

We evaluate how different LoRA configurations affect performance. We used LS-FA-Llama3.2-3B and LS-FA-Llama3.1-8B, which outperformed the baselines and other LS-FA-LLM results. Table 4 shows the results of five different LoRA configurations for these two models. The row in lightgrey is the result of LoRA configuration used in Tables 2 and 3.

In the table, the columns of "r", "alpha", "dropout", and "modules" specify the LoRA configuration items: attention dimension (rank), LoRA scaling, dropout probability, and target modules, respectively. Column "#trainable params" specified the number of trainable params, and its percentage of the model size. By adopting LoRA, the number of trainable parameters in LS-FA-LLMs is much lower than in encoder models trained without LoRA.

The results of these two LS-FA-LLMs vary across different LoRA configurations. However, we confirm that LoRA configurations with relatively large $rank \geq 32$ and $alpha = 2 \times rank$ targeted for all linear modules achieve higher performance.

Training	Full			LoRA		
BC5CDR	Precision	Recall	F1	Precision	Recall	F1
SOTA						
OpenMed (Panahi, 2025)	F1 = 0.961 (Chemical), F1 = 0.912 (Disease)					
BioBERT_v1.1 (Lee et al., 2019)	0.874	**0.918**	**0.895**	0.860	0.908	0.884
Baseline Encoders						
BERT	0.845	0.872	0.858	0.828	0.875	0.851
RoBERTa	0.871	0.890	0.880	0.842	0.889	0.865
DeBERTa	**0.876**	0.889	0.882	0.877	0.909	0.893
LS-CA-LLM						
LS-CA-Gemma-2B	0.716	0.701	0.708	0.747	0.776	0.761
LS-CA-Gemma-7B	0.622	0.628	0.625	0.768	0.801	0.784
LS-CA-Qwen2.5-0.5B	0.628	0.612	0.620	0.666	0.681	0.674
LS-CA-Qwen2.5-7B	0.669	0.696	0.682	0.710	0.730	0.720
LS-CA-Llama3.2-1B	0.630	0.602	0.616	0.692	0.704	0.698
LS-CA-Llama3.2-3B	0.634	0.621	0.627	0.710	0.732	0.721
LS-CA-Llama3.1-8B	0.660	0.655	0.657	0.719	0.739	0.729
LS-FA-LLM						
LS-FA-Gemma-2B	0.848	0.853	0.851	0.873	0.899	0.886
LS-FA-Gemma-7B	0.737	0.730	0.734	0.876	0.904	0.890
LS-FA-Qwen2.5-0.5B	0.849	0.825	0.837	0.858	0.868	0.863
LS-FA-Qwen2.5-7B	0.765	0.735	0.750	0.875	0.902	0.888
LS-FA-Llama3.2-1B	0.799	0.799	0.799	0.876	0.901	0.889
LS-FA-Llama3.2-3B	0.843	0.839	0.841	**0.892**	0.906	0.899
LS-FA-Llama3.1-8B	0.810	0.798	0.804	0.887	**0.920**	**0.904**

Table 2: BC5CDR: Precision, recall and micro F1 scores for the baseline encoders, LS-CA-LLMs and LS-FA-LLMs. At the top, we include SOTA results by OpenMed (Panahi, 2025) for reference and BioBERT_v1.1 (Lee et al., 2019). The first three columns are results of full training, and the last three columns are results of LoRA training. For each column, the best score is indicated in **bold** font. The scores with a yellow background specify the best score across full and LoRA training.

5.3 Comparison with instruction fine-tuned LLMs approach

Table 5 shows the comparison with instruction fine-tuned (IFT) counterparts with label-supervised full-attention LLMs and baseline encoders. We also add zero-shot results by OpenAI GPT-4.1 as a baseline (see in Appendix C for details). The best scores are indicated in bold font.

To train LLMs using IFT, we adopt an Alpaca-based instruction format (Taori et al., 2023) for fine-tuning, as described in the Appendix B. Since the named entities are extracted from the output text, the performance of IFT is evaluated using set-based precision, recall, and F1 scores between the estimated and true sets of entities, ignoring the spans of the extracted entities. We use the exact match when comparing estimated and true results.

To compare with span-based results by label-supervised LLMs and encoders, we convert them to set-based metrics by aggregating extracted entities.

To train IFT models, we used the same LoRA configuration as the label-supervised full-attention counterparts.

First, IFT-LLM results outperformed GPT-4.1 in zero-shot settings, indicating that correctly extracting entities remains challenging for high-performance closed-weight LLMs. Larger IFT-LLM results on BC5CDR show competitive performance relative to baseline encoders; however, they fall short of the baseline encoders on CoNLL2003.

Results on LS-FA-LLMs show superior performance on both BC5CDR and CoNLL2003, outperforming both IFT counterparts and baseline encoders. These results, combined with the results on Tables 2 and 3, underscore the superiority of the proposed LS-FA-LLMs.

6 Analysis

6.1 Training and Inferencing time

Table 6 shows a comparison of training time among all models for one epoch using a single H200 GPU. The column 'rel' indicates the relative time based on BERT's time. In these experiments, baseline encoder models are trained without LoRA, and LS-FA-LLMs are trained with LoRA. We confirm that, even with LoRA, much longer training times are required for LS-FA-LLMs. The training time proportionally increases with the size of the original LLMs. When fine-tuning on 7B LS-FA-LLMs, about $16-18\times$ and $15-16\times$ training times are re-

Training	Full			LoRA		
CoNLL2003	Precision	Recall	F1	Precision	Recall	F1
SOTA						
ACE+Fine-tune (Wang et al., 2021)			F1 = 0.946			
Baseline Encoders						
BERT	0.901	0.915	0.908	0.903	0.918	0.910
RoBERTa	**0.908**	**0.924**	**0.915**	0.912	0.928	0.920
DeBERTa	0.904	0.917	0.911	0.903	0.918	0.910
LS-CA-LLM						
LS-CA-Gemma-2B	0.729	0.780	0.754	0.762	0.814	0.787
LS-CA-Gemma-7B	0.680	0.739	0.708	0.766	0.815	0.790
LS-CA-Qwen2.5-0.5B	0.609	0.667	0.636	0.656	0.714	0.684
LS-CA-Qwen2.5-7B	0.656	0.714	0.684	0.696	0.751	0.722
LS-CA-Llama3.2-1B	0.626	0.678	0.651	0.684	0.740	0.711
LS-CA-Llama3.2-3B	0.618	0.675	0.645	0.700	0.757	0.727
LS-CA-Llama3.1-8B	0.643	0.700	0.671	0.699	0.754	0.725
LS-FA-LLM						
LS-FA-Gemma-2B	0.873	0.883	0.878	0.914	0.929	0.922
LS-FA-Gemma-7B	0.778	0.817	0.797	0.922	0.932	0.927
LS-FA-Qwen2.5-0.5B	0.806	0.818	0.812	0.887	0.894	0.891
LS-FA-Qwen2.5-7B	0.851	0.861	0.856	0.910	0.929	0.920
LS-FA-Llama3.2-1B	0.851	0.871	0.861	0.915	0.927	0.920
LS-FA-Llama3.2-3B	0.845	0.859	0.852	**0.930**	**0.941**	**0.935**
LS-FA-Llama3.1-8B	0.773	0.795	0.784	0.922	0.936	0.929

Table 3: CoNLL2003: Precision, recall and micro F1 scores for the baseline encoders, LS-CA-LLMs and LS-FA-LLMs. At the top, we include SOTA results by ACE+Fine-tune (Wang et al., 2021) for reference. The first three columns are results of full training, and the last three columns are results of LoRA training. For each column, the best score is indicated in **bold** font. The scores with a yellow background specify the best score across full and LoRA training.

quired for BC5CDR and CoNLL2003, respectively, compared with BERT.

These results show a negative effect of using larger LLMs for NER, despite achieving higher performance.

However, if multiple GPUs are available, the training time of LS-FA-LLMs can be further reduced using a distributed parallel training method

Table 7 shows the mean and the standard deviation of inference time in milliseconds to process one sample among 10 randomly selected test samples contained in BC5CDR and CoNLL2003 using a single H200 GPU. Larger LS-FA-LLMs such as LS-FA-Gemma-7B and LS-FA-Llama3.1-8B take $\sim$ 10$\times$ inference time compared with BERT, and smaller LS-FA-LLMs proportionally take less inference time as their parameter size gets smaller. LS-FA-Qwen2.5-0.5B shows exceptionally longer inference time compared with models with similar sizes, such as LS-FA-Gemma-2B and LS-FA-Llama3.2-1B. We have yet to analyze this phenomenon and leave it for future work.

We note again that if multiple GPUs are available, the inference time of LS-FA-LLMs can be further reduced using a distributed parallel decoding method.

6.2 Error Analysis

Tables 11 and 12 show the error analysis of DeBERTa, LS-FA-Llama3.1-8B, and LS-CA-Llama3.1-8B for BC5CDR and CoNLL2003, respectively. NER error was classified into three classes: SPAN, TYPE, and DETECTION errors, and each error class was further classified into more detailed error types.

In both analyses, we confirmed that the counts of errors for each error type were lower in LS-FA-Llama3.1-8B than in DeBERTa. This indicates that LS-FA-Llama3.1-8B has acquired more robust named-entity recognition knowledge than DeBERTa. LS-CA-Llama3.1-8B shows poor performance in both analyses, confirming that the causal attention mechanism is the bottleneck for NER.

In the analysis of LS-FA-Llama3.1-8B and DeBERTa, TYPE error counts are much lower in BC5CDR than in CoNLL2003, indicating greater ambiguity among entity types in CoNLL2003.

The above results and analysis confirm that the decoder LLMs' weakness in structured precision tasks, such as NER, can be mitigated by our proposed method.

Model	BC5CDR F1	CoNLL2003 F1	r	alpha	dropout %	modules	#trainable params
LS-FA-Llama3.2-3B	0.883	0.914	8	16	0.05	q,k,v,o	4.60M (0.143%)
	0.889	0.918	16	32	0.05	q,k,v,o	9.19M (0.285%)
	0.899	**0.935**	32	64	0.05	all linear	48.60M (1.490%)
	0.898	0.928	64	128	0.05	all linear	97.30M (2.940%)
	0.900	0.927	128	256	0	all linear	195.00M (5.710%)
LS-FA-Llama3.1-8B	0.888	0.919	8	16	0.05	q,k,v,o	6.84M (0.091%)
	0.893	0.922	16	32	0.05	q,k,v,o	13.70M (0.182%)
	0.904	0.929	32	64	0.05	all linear	83.90M (1.110%)
	0.904	0.929	64	128	0.05	all linear	168.00M (2.190%)
	0.900	**0.932**	128	256	0	all linear	336.00M (4.280%)

Table 4: Test micro F1 scores for LS-FA-Llama3.2-3B and LS-FA-Llama3.1-8B with five different LoRA configurations, including the one used for previous experiments shown in light gray background. Columns "r" and "alpha" specify rank and alpha values of LoRA parameters. The "modules" specifies the targeted modules and "all linear" specifies all linear modules: q_proj,k_proj,v_proj,o_proj,down_proj,gate_proj and up_proj.

7 Conclusion

We address weaknesses in decoder LLMs for structure prediction tasks such as NER by focusing on the attention mechanism and adopting a discriminative approach with supervised labels. Specifically, we demonstrate that this weakness can be mitigated by replacing causal attention with a full-attention mask during both training and inference.

However, our results show that this change to the self-attention mechanism alone does not yield superior NER performance compared with strong baselines such as RoBERTa and DeBERTa. Specifically, we verified that instead of full fine-tuning, using LoRA with a suitable configuration is key to boosting the performance. This is especially true for large LLMs because full fine-tuning on a relatively small training dataset, such as BC5CDR or CoNLL2003, can easily lead to overfitting, hindering performance on the test set.

Our results also suggested that linguistic knowledge acquired during pre-training of LLMs with causal attention is transferable even after the attention mechanism is switched to full-attention during fine-tuning. In our experiments, we used only the training subsets of the NER benchmarks for fine-tuning, so additional training with an auxiliary dataset could increase the performance even more.

Although our proposed best model outperformed all baseline encoders on BC5CDR and CoNLL2003, it requires much longer training and inference times than the baselines. Nonetheless, we verified that our proposed method is highly effective and can serve as an alternative encoder model to BERT-like models in downstream applications.

Since the LoRA configuration significantly affects the performance of the proposed methods, future work should investigate automated methods to obtain an optimal LoRA configuration.

Acknowledgments

This paper is based on results obtained from AIST policy-based budget project "R&D on Generative AI Foundation Models for the Physical Domain".

Limitations

In our experiments, we conducted a single fine-tuning step for each model, using a fixed learning rate and other hyperparameters. With more computing resources, multiple runs with different training parameters might yield more reliable results.

Our experiments were conducted on two NER datasets, one in the biomedical domain and the other in the general domain, so further experiments on datasets in other domains and languages could yield additional insights.

References

Arthur Amalvy, Vincent Labatut, and Richard Dufour. 2023. The role of global and local context in named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 714–722, Toronto, Canada. Association for Computational Linguistics.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *Preprint*, arXiv:2404.05961.

Yubo Chen, Chuhan Wu, Tao Qi, Zhigang Yuan, and Yongfeng Huang. 2020. Named entity recognition

		BC5CDR			CoNLL2003		
Model	**Training**	**Precision (set)**	**Recall (set)**	**F1 (set)**	**Precision (set)**	**Recall (set)**	**F1 (set)**
Baseline Encoders							
BERT	Full	0.861	0.877	0.869	0.914	0.914	0.914
RoBERTa	Full	0.882	0.896	0.889	0.917	0.921	0.919
DeBERTa	Full	0.887	0.896	0.892	0.915	0.916	0.915
IFT-LLM							
GPT-4.1	Zero-shot	0.670	0.505	0.576	0.388	0.669	0.491
IFT-Gemma-2B	LoRA	0.870	0.858	0.864	0.870	0.859	0.864
IFT-Gemma-7B	LoRA	0.878	0.876	0.877	0.876	0.873	0.874
IFT-Qwen2.5-0.5B	LoRA	0.803	0.804	0.804	0.807	0.808	0.808
IFT-Qwen2.5-7B	LoRA	0.881	0.878	0.880	0.882	0.874	0.878
IFT-Llama3.2-1B	LoRA	0.866	0.861	0.864	0.865	0.860	0.862
IFT-Llama3.2-3B	LoRA	0.879	0.862	0.870	0.879	0.861	0.869
IFT-Llama3.1-8B	LoRA	0.901	0.887	0.894	0.902	0.888	0.895
LS-FA-LLM							
LS-FA-Gemma-2B	LoRA	0.889	0.900	0.894	0.929	0.930	0.930
LS-FA-Gemma-7B	LoRA	0.887	0.906	0.896	0.932	0.933	0.933
LS-FA-Qwen2.5-0.5B	LoRA	0.874	0.874	0.874	0.899	0.897	0.898
LS-FA-Qwen2.5-7B	LoRA	0.893	0.902	0.898	0.920	0.925	0.923
LS-FA-Llama3.2-1B	LoRA	0.893	0.902	0.898	0.928	0.928	0.928
LS-FA-Llama3.2-3B	LoRA	**0.907**	0.909	0.908	**0.937**	**0.940**	**0.938**
LS-FA-Llama3.1-8B	LoRA	0.904	**0.920**	**0.912**	0.933	0.936	0.935

Table 5: Comparison with instruction fine-tuned (IFT) counterparts with label supervised full-attention LLMs and baseline encoders. Performance of IFT is evaluated by set-based precision, recall, and f1 between estimated entities and true entities using exact match. Token-based results by label-supervised LLMs and encoders are converted to set-based metrics. Best scores are in bold face.

	BC5CDR		CoNLL2003	
	[mm:ss]	rel.	[mm:ss]	rel.
BERT	00:39	1.0	01:08	1.0
RoBERTa	00:36	0.9	01:09	1.0
DeBERTa	01:47	2.8	03:27	3.1
LS-FA-Gemma-2B	03:30	5.4	05:49	5.2
LS-FA-Gemma-7B	11:47	18.3	18:29	16.4
LS-FA-Qwen2.5-0.5B	02:08	3.3	03:46	3.3
LS-FA-Qwen2.5-7B	11:25	17.8	17:21	15.4
LS-FA-Llama3.2-1B	02:13	3.5	03:58	3.5
LS-FA-Llama3.2-3B	05:20	8.3	08:23	7.4
LS-FA-Llama3.1-8B	11:48	18.4	17:54	15.9

Table 6: Elapsed time to train encoders and LS-FA-LLMs for BC5CDR and CoNLL2003, each for one epoch, using one H200 GPU. The column 'rel' indicates relative time based on BERT's time. Encoder models are trained without LoRA and LS-FA-LLMs are trained with LoRA.

	BC5CDR	CoNLL2003
	Time [msec]	Time [msec]
BERT	2.15 ± 0.02	2.82 ± 0.10
RoBERTa	3.03 ± 0.08	3.54 ± 0.03
DeBERTa	17.88 ± 0.46	18.23 ± 0.34
LS-FA-Gemma-2B	8.29 ± 0.08	7.91 ± 0.06
LS-FA-Gemma-7B	23.93 ± 0.59	26.48 ± 0.37
LS-FA-Qwen2.5-0.5B	15.55 ± 3.77	11.32 ± 0.26
LS-FA-Qwen2.5-7B	21.31 ± 0.31	24.42 ± 0.68
LS-FA-Llama3.2-1B	6.23 ± 0.21	12.85 ± 0.29
LS-FA-Llama3.2-3B	13.20 ± 0.20	11.57 ± 0.69
LS-FA-Llama3.1-8B	23.99 ± 0.45	26.39 ± 0.81

Table 7: Elapsed inference time in milliseconds to process an example out of 10 randomly selected test samples for BC5CDR and CoNLL2003

in multi-level contexts. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 181–190, Suzhou, China. Association for Computational Linguistics.

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

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. 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 3198–3213, Online. Association for Computational Linguistics.

David Dukić and Jan Šnajder. 2024. Looking right is sometimes right: Investigating the capabilities of

decoder-only llms for sequence labeling. *Preprint*, arXiv:2401.14556.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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, San Diego, California. Association for Computational Linguistics.

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.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu lee Wang, Qing Li, and Xiaoqin Zhong. 2023. Label supervised llama finetuning. *Preprint*, arXiv:2310.01208.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Preprint*, arXiv:1907.11692.

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

Masoud Monajatipoor, Jiaxin Yang, Joel Stremmel, Melika Emami, Fazlolah Mohaghegh, Mozhdeh Rouhsedaghat, and Kai-Wei Chang. 2024. Llms in biomedicine: A study on clinical named entity recognition. *Preprint*, arXiv:2404.07376.

Maziyar Panahi. 2025. Openmed ner: Open-source, domain-adapted state-of-the-art transformers for biomedical ner across 12 public datasets. *Preprint*, arXiv:2508.01630.

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. *ArXiv*, abs/2101.05779.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *Preprint*, arXiv:1802.05365.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global pointer: Novel efficient span-based approach for named entity recognition. *Preprint*, arXiv:2208.03054.

Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020. Boundary enhanced neural span classification for nested named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9016–9023.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting DocRED - addressing the false negative problem in relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8472–8487, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

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

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022. SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3476, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

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

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. 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 2643–2660, Online. Association for Computational Linguistics.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database: The Journal of Biological Databases and Curation*, 2016.

Weilu Xu, Renfei Dang, and Shujian Huang. 2025. LLM's weakness in NER doesn't stop it from enhancing a stronger SLM. In *Proceedings of the Second Workshop on Ancient Language Processing*, pages 170–175, The Albuquerque Convention Center, Laguna. Association for Computational Linguistics.

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. AutoRE: Document-level relation extraction with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 211–220, Bangkok, Thailand. Association for Computational Linguistics.

A Training parameters

Baseline encoders, label-supervised LLMs, and instruction fine-tuned LLMs are trained using the same hyperparameters shown in Table 8. Tables 9 and 10 show default LoRA configurations used for our experiments for LLMs and encoders.

Hyper-parameter	Value
per_device_batch_size	8
gradient_accumulation_steps	1
num_train_epochs	10
learning_rate	5E-05
weight_decay	0.001
warmup_ratio	0
lr_scheduler_type	cosine
gradient_checkpointing	False
model_max_length	512
bf16	false

Table 8: Training parameters

LoRA configuration item	Value
lora_r	32
lora_alpha	64
lora_dropout	0.05
lora_bias	none
lora_taget_modules	q_proj,k_proj,v_proj,o_proj down_proj,gate_proj,up_proj

Table 9: LoRA configuration for LS-LLMs and IFT-LLMs

LoRA configuration item	Value
lora_r	32
lora_alpha	64
lora_dropout	0.05
lora_bias	none
lora_taget_modules	query, key, value, dense

Table 10: LoRA configuration for encoders

B Instruction format for fine-tuning decoder-only LLMs

Sample instruction for BC5CDR

Instruction:
please extract named entities and their type from the input sentence, all entity types are in options.
Options:
chemical,disease
Sentence:
The hypotensive effect of 100 mg / kg alpha-methyldopa was also partially reversed by naloxone .
Response:
alpha-methyldopa:chemical;naloxone:chemical;hypotensive:disease;

Sample instruction for CoNLL2003

Instruction:
please extract named entities and their type from the input sentence, all entity types are in options.
Options:
organization,miscellaneous,person,location
Sentence:
The European Commission said on Thursday it disagreed with German advice to consumers to shun British lamb until scientists determine whether mad cow disease can be transmitted to sheep .
Response:
European Commission:organization;German:miscellaneous;British:miscellaneous;

C Zero-shot NER using GPT-4.1 by OpenAI

The following is a sample code snippet of zero-shot NER for BC5CDR and CoNLL2003 using OpenAI's structured output.

```python
from openai import OpenAI
from pydantic import BaseModel
from typing import Literal

client = OpenAI()

class Entity_BC5CDR(BaseModel):
    disease: list[str]
    chemical: list[str]

class Entity_CoNLL2003(BaseModel):
    organization: list[str]
    miscellaneous: list[str]
    person: list[str]
    location: list[str]

def extract_entity(content: str, task: Literal["BC5CDR","CoNLL2003"]):

    response = client.responses.parse(
        model="gpt-4.1",
        input=[
                {
                    "role": "system",
                    "content": "You are an excellent linguist. Extract all the mentions that can be considered diseases or chemicals." if task == 'BC5CDR' else "You are an excellent linguist. Extract all the mentions that can be considered organizations, people, locations, or miscellaneous.",
                },
                {
                    "role": "user",
```

```
                    "content": f"{content}",
                },
            ],
        text_format=Entity_BC5CDR if task == 'BC5CDR' else Entity_CoNLL2003,
    )
    entities = response.output_parsed
    return entities

if __name__ == "__main__":
    entities = extract_entity(content, task_name)
```

D Error analysis

NER errors are classified into SPAN, TYPE, and DETECTION error categories, and each error category is further classified into finer error types. Errors are computed by comparing gold and predicted labels, each represented as a list of entity labels of the form (start, end, etype), where 'start' and 'end' are the beginning and end of the entity span, and 'etype' is the entity type.

In SPAN errors, if either 'start' or 'end' position matches, but the span length is not matched, they are classified as too_short or too_long. Whereas in the case of both 'start' and 'end' positions not matching, but the difference of span length is within two words, it is classified as boundary_off. The others are classified as completely_wrong. If span position matches, but 'etype' does not match, they are classified as correct_span_wrong_type.

BC5CDR	**DeBERTa**	**LS-FA-Llama3.1-8B**	**LS-CA-Llama3.1-8B**
Total Prediction	10502	10442	11383
Exact Matches	8705	9007	7247
Total Errors	1797	1435	4136
SPAN ERRORS			
boudary_off	5 (0.3%)	4 (0.3%)	1 (0.0%)
too_short	198 (11.0%)	195 (13.6%)	284 (6.9%)
too_long	244 (13.6%)	169 (11.8%)	138 (3.3%)
completely_wrong	7 (0.4%)	6 (0.4%)	10 (0.2%)
TYPE ERRORS			
correct_span_wrong_type	39 (2.2%)	14 (1.0%)	195 (4.7%)
DETECTION ERRORS			
false_positives	693 (38.6%)	633 (44.1%)	1574 (38.1%)
false_negatives	611 (34.0%)	414 (28.9%)	1934 (46.8%)

Table 11: Error analysis of DeBERTa, LS-FA-Llama3.1-8B, and LS-CA-Llama3.1-8B for BC5CDR

CoNLL2003	**DeBERTa**	**LS-FA-Llama3.1-8B**	**LS-CA-Llama3.1-8B**
Total Prediction	5800	5775	5912
Exact Matches	5177	5283	4219
Total Errors	623	492	1693
SPAN ERRORS			
boudary_off	0 (0.0%)	1 (0.2%)	0 (0.0%)
too_short	52 (8.3%)	47 (9.6%)	49 (2.9%)
too_long	75 (12.0%)	44 (8.9%)	29 (1.7%)
completely_wrong	0 (0.0%)	0 (0.0%)	0 (0.0%)
TYPE ERRORS			
correct_span_wrong_type	236 (37.9%)	176 (35.8%)	734 (43.4%)
DETECTION ERRORS			
false_positives	152 (24.4%)	127 (25.8%)	264 (15.6%)
false_negatives	108 (17.3%)	97 (19.7%)	617 (36.4%)

Table 12: Error analysis of DeBERTa, LS-FA-Llama3.1-8B, and LS-CA-Llama3.1-8B for CoNLL2003