# Overcoming Data Scarcity in Named Entity Recognition: Synthetic Data Generation with Large Language Models

**Tuan An Dao**[1,2] **Hiroki Teranishi**[2] **Yuji Matsumoto**[2]
**Florian Boudin**[3] **Akiko Aizawa**[4,2]
[1]The University of Tokyo, Tokyo, Japan
[2]RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan
[3]JFLI, Nantes Université, France
[4]National Institute of Informatics, Tokyo, Japan
dtan@g.ecc.u-tokyo.ac.jp, {hiroki.teranishi, yuji.matsumoto}@riken.jp,
florian.boudin@univ-nantes.fr, aizawa@nii.ac.jp

## Abstract

Named Entity Recognition (NER) is crucial for extracting domain-specific entities from text, particularly in biomedical and chemical fields. Developing high-quality NER models in specialized domains is challenging due to the limited availability of annotated data, with manual annotation being a key method of data construction. However, manual annotation is time-consuming and requires domain expertise, making it difficult in specialized domains. Traditional data augmentation (DA) techniques also rely on annotated data to some extent, further limiting their effectiveness. In this paper, we propose a novel approach to synthetic data generation for NER using large language models (LLMs) to generate sentences based solely on a set of example entities. This method simplifies the augmentation process and is effective even with a limited set of entities. We evaluate our approach using BERT-based models on the BC4CHEMD, BC5CDR, and TDMSci datasets, demonstrating that synthetic data significantly improves model performance and robustness, particularly in low-resource settings. This work provides a scalable solution for enhancing NER in specialized domains, overcoming the limitations of manual annotation and traditional augmentation methods.

## 1 Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) aiming at identifying and classifying named entities in text. The primary goal of NER is to extract specific entities such as people, organizations, locations, and specialized terms (e.g., chemicals, diseases) from unstructured text. Effective NER is vital in many fields, particularly in the biomedical and chemical domains, where accurate entity recognition supports applications such as drug discovery, literature mining, and patent analysis.

One significant challenge in developing high-quality NER models is the scarcity of annotated data, particularly in specialized domains and low-resource scenarios. Recent advancements in data augmentation for NER have explored diverse strategies to tackle data scarcity, especially in low-resource settings. Techniques range from simple methods like synonym replacement (Dai and Adel, 2020; Sabty et al., 2021; Chen et al., 2021; Yaseen and Langer, 2021; Phan and Nguyen, 2022; Sutiono and Hahn-Powell, 2022) and random noise introduction (Issifu and Ganiz, 2021; Liu et al., 2023) to more complex approaches such as cross-domain transformation and leveraging large language models (LLMs) (Liu et al., 2022; Ye et al., 2024). These methods help to generate additional training examples but often still rely on existing labeled data, which can limit their effectiveness in highly specialized domains where labeled examples are scarce or non-existent.

To overcome these limitations, we propose an approach for synthetic data generation using LLMs that generates sentences based solely on a set of example entities, without relying on pre-existing annotated data. Our method (GenLLM) simplifies the augmentation process by directly generating domain-relevant sentences while ensuring entity correctness and contextual consistency. Unlike traditional techniques, our approach does not depend on manually annotated examples, making it especially valuable for low-resource or highly specialized domains where obtaining labeled data is challenging. By leveraging LLMs, we can produce diverse and contextually appropriate sentences that reflect real-world entity occurrences and relationships. We evaluate the effectiveness of our approach by applying it to NER tasks using BERT-based models on three datasets: the widely used BC4CHEMD (Krallinger et al., 2015) and BC5CDR (Li et al., 2016) datasets, along with the TDMSci (Hou et al., 2021) dataset for task, dataset, and metric entities. Our results show that pretraining on synthetic data generated by LLMs



*Proceedings of the 24th Workshop on Biomedical Language Processing (BioNLP 2025)*, pages 328–340

consistently improves model performance, outperforming previous data augmentation methods that combine synthetic data with the original training data in both low and high-resource settings. We explored using only synthetic data generated by LLMs for training, which proved effective in low-resource scenarios. However, human-annotated data yielded better results as the dataset size increased, emphasizing the value of expert annotations in high-resource settings. GenLLM offers a promising data augmentation solution for low-resource domains, particularly when annotated data is limited. The code, generated data, and trained models used in this work are publicly available at https://github.com/daotuanan/GenLLM_NER.

## 2 Related Work

NER relies heavily on high-quality annotated datasets, but in many specialized domains, such as the biomedical and scientific domain, manually labeled data are scarce. To address this issue, synthetic data generation has emerged as an alternative to enhance model performance (Xu et al., 2024). Generating synthetic data for the NER task is challenging because it requires more than just producing natural-sounding sentences; it must also ensure entity correctness, contextual consistency, and domain relevance. Unlike general text generation, NER data must contain entities that are correctly labeled and naturally embedded within the context, reflecting real-world sentence structures.

### 2.1 Traditional Data Augmentation Methods for NER

Traditional augmentation methods such as synonym replacement, backtranslation, and cross-domain adaptation have been used to enhance NER performance, particularly in low-resource settings (Dai and Adel, 2020; Sabty et al., 2021; Issifu and Ganiz, 2021; Chen et al., 2021; Yaseen and Langer, 2021; Phan and Nguyen, 2022). While these techniques have proven effective, they often struggle to generate highly contextualized and domain-specific entity mentions. For instance, basic methods like synonym replacement and random insertion have shown improvements in biomedical NER (Issifu and Ganiz, 2021), and backtranslation has been particularly effective in low-resource biomedical and materials science domains (Yaseen and Langer, 2021). However, these methods typically fail to capture complex entity structures and contextual dependencies required for domain-specific tasks.

### 2.2 LLM-Based Approaches to Data Generation

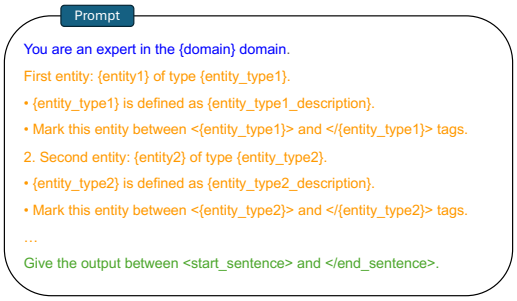
Recent advances in LLM-based synthetic data generation offer a more flexible and scalable alternative. LLMs can generate diverse, contextually rich sentences while preserving entity correctness and domain relevance. For example, prompting strategies have been shown to significantly enhance NER performance in low-resource scenarios, improving F1 scores by over 40% (Liu et al., 2022). Techniques like context similarity-based augmentation (e.g., COSINER) and transformer-based data generation have demonstrated effectiveness in improving NER in both general and specialized domains, such as biomedical texts (Bartolini et al., 2022, 2023; Yili and Haonan, 2023). Moreover, methods like TarGEN employ multi-step prompting and self-correction to generate high-quality synthetic datasets (Gupta et al., 2023). Despite the promise of these approaches, challenges remain in ensuring the scalability and quality of synthetic data, particularly in highly specialized domains like clinical NER (Hiebel et al., 2023). However, a key limitation of these studies is their focus on general rather than specialized domains. The effectiveness of synthetic data and pretraining methods might not translate well to domain-specific applications, such as biomedical or clinical NER.

## 3 Entity-Based Synthetic Data Generation

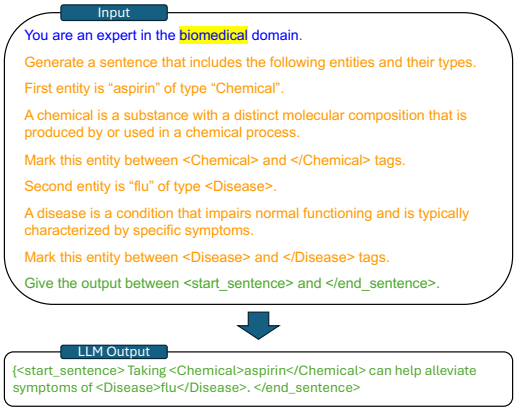
Our approach leverages the LLMs to generate synthetic sentences that incorporate specified entities while maintaining contextual consistency. The process consists of three main steps: entity selection 3.1, prompt construction 3.2, and sentence generation 3.3.

### 3.1 Entity Selection

We begin by selecting a set of seed entities, which serve as the foundation for sentence generation. These entities can be obtained from existing datasets, knowledge bases, or domain-specific lexicons with relatively low effort compared to manually creating fully annotated sentences. The selection ensures that the generated data covers a diverse set of entity mentions necessary for effective NER training. Entities are randomly combined




(a) Prompt structure.




(b) Example prompt instance.

Figure 1: Illustration of the example prompt used for generating synthetic sentences with specified entities and their types. The prompt includes the model's role, task instructions, and output formatting guidelines.

from different categories, with each sentence containing one to three entities. For example, from the categories CHEM (aspirin, lithium) and DISEASE (lung carcinoma, flu), a possible combination could be: aspirin, flu.

### 3.2 Prompt Construction

To generate high-quality synthetic sentences for NER, we design a structured prompt that ensures the inclusion of specific entities while maintaining contextual coherence. Our prompt explicitly defines the domain, entity types, and entity annotations to improve generation accuracy and reduce annotation errors.

The prompt follows a template-based format that guides the language model to generate a sentence containing specified entities with correct annotations. It consists of the following key components:

- Domain Specification: The model is instructed to act as an expert in a specific domain (e.g., biomedical sciences) to ensure domain-relevant sentence generation.
- Entity Introduction and Definition: Each target entity is explicitly listed along with its type and a brief description of that type. This helps the model understand the contextual role of the entity.
- Entity Annotation Instructions: The prompt explicitly instructs the model to enclose entities within predefined tags, ensuring clear entity labeling in the generated sentence.
- Output Formatting: The generated sentence is enclosed within `<start_sentence>` and `</end_sentence>` tags to facilitate automatic extraction and processing.

This prompt serves as the foundation for generating synthetic NER training data, ensuring both entity correctness and contextual consistency in the generated sentences. The prompt template is found in Figure 1a. An example of this prompt format is illustrated in Figure 1b, demonstrating how contextual cues and entity definitions improve generation accuracy. This prompt format can be used with any popular LLM for generating synthetic data.

### 3.3 Sentence Generation

We use the **LLaMA-3.2-3B-Instruct**[1] model to generate synthetic sentences containing specified named entities. This model was selected for its balance between generation quality and computational efficiency. Unlike prior work that relies on proprietary and resource-intensive models such as GPT-4 or GPT-4o (Ye et al., 2024), our approach uses an *open-source*, lightweight model that is more accessible and cost-effective, making it better suited for reproducible research and large-scale generation in constrained environments. Once the LLM processes the prompt, it generates a synthetic sentence where the specified entities are correctly embedded within a natural linguistic context. To maintain consistency and avoid introducing unintended entities, we post-process the output by verifying entity correctness and ensuring compliance with the annotation format. To ensure that the generated output adheres to the required annotation format, we apply the following post-processing steps:

[1] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

- **Tag Validation**: We verify that all entity tags are correctly opened and closed. Each entity must be enclosed within its respective <entity_type> and </entity_type> tags to maintain proper annotation structure.

- **Sentence Formatting**: We confirm that the entire sentence is enclosed within <start_sentence> and </end_sentence> tags. This ensures the output remains structured and easily extractable for further processing.

By enforcing these constraints, we ensure consistency in the synthetic data before it is used for training. After validating the output format, we convert the annotated entities into the BIO (Begin-Inside-Outside) tagging scheme. Each token in the sentence is assigned a label based on its entity type. This transformation ensures compatibility with standard NER training pipelines.

The synthetic data is then incorporated into the training set through pretraining, where the model is first trained on the synthetic data before being fine-tuned on gold-standard annotated data. This approach helps the model learn general entity patterns from the generated data, improving performance, especially in low-resource or specialized domains.

## 4 Experiment with Low-resource Setting

In this experiment, we explore the performance of our method (GenLLM) in a low-resource setting, where only a limited amount of manually annotated data is available. The aim is to evaluate whether our method can outperform or complement other state-of-the-art systems, such as LSMS(Dai and Adel, 2020), LLM-DA(Ye et al., 2024), and NuNER(Bogdanov et al., 2024), when trained with a small amount of labeled data. We also investigate how different data augmentation strategies and pretraining methods impact the model's ability to generalize to unseen examples.

### 4.1 Experimental Setup

#### 4.1.1 Dataset Construction

We conduct experiments using 3 datasets: BC4CHEMD, BC5CDR, and TDMSci. Since we use seed entities as the main input for the augmentation process, it is important to note that obtaining a large set of seed entities in real-world applications can be difficult, particularly in specialized domains where annotated data is scarce. As a result, working with a smaller, more manageable set of seed entities is often necessary. Our method, which only uses seed entities for the augmentation process, is designed to be effective even with this limitation. In contrast, other methods like LSMS and LLM-DA rely on gold-label data as input for augmentation. We create a "Limited Dictionary" setting to compare our method with these alternative approaches.

To construct the seed sets used for augmentation, we select the most frequent entities from the training data for each entity type. For each dataset, we define multiple settings with different values of $N$ (e.g., $N = 5, 10, 15, 20, 50$), where $N$ denotes the number of unique entities per type. The selection process involves counting and ranking entities by frequency, then selecting the top $N$ for each type. We also ensure type balance by including an equal number of sentences for each entity type (e.g., equal numbers for CHEMICAL and DISEASE in BC5CDR).

This choice of using frequent entities—rather than randomly sampling or relying on external lexicons—is motivated by both practical and methodological reasons. First, frequent entities are more likely to appear in natural, contextually appropriate sentences, resulting in higher-quality and more realistic generated data. Second, using a fixed set of frequent entities leads to a more stable and reproducible experimental setup. In contrast, random sampling introduces variability and typically requires multiple runs to obtain robust estimates. Similarly, depending on external lexicons may introduce domain mismatches or lead to unnatural entity combinations. By relying on the internal statistics of the training corpus, we ensure that the selected entities are representative of the target domain and the actual model training distribution.

#### 4.1.2 Comparison Methods

We consider the following baseline methods for comparison purposes:

- **Original (org)**: Training directly on the full dataset without any augmentation.

- **LSMS**: Applying lexical-based sampling and substitution strategies, including replace-mention (RM), replace-token (RT), shuffle-within-segments (SWS), and synonym-replacement (SR).

| Dictionary Size | 5 | 10 | 15 | 20 | 50 |
|---|---|---|---|---|---|
| org | 4.68 | 12.35 | 15.62 | 31.79 | 47.18 |
| LSMS | 32.85 | **46.03** | 44.35 | **48.34** | 57.76 |
| LLM-DA | **39.89** | 43.17 | **45.75** | 46.29 | 49.68 |
| NuNER | 21.24 | 28.29 | 40.11 | 44.31 | 52.49 |
| Ours (GenLLM) | 26.06 | 37.18 | 41.85 | 43.67 | **58.80** |

Table 1: Performance comparison on the BC4CHEMD dataset across different dictionary sizes ($N$).

| Dictionary Size | 5 | 10 | 15 | 20 | 50 |
|---|---|---|---|---|---|
| org | 45.62 | 51.87 | 51.81 | 51.73 | 54.83 |
| LSMS | 51.28 | 57.19 | 60.66 | 60.97 | 68.42 |
| LLM-DA | 52.29 | 57.72 | 60.94 | 64.12 | 66.79 |
| NuNER | 40.70 | 43.45 | 50.17 | 50.86 | 46.87 |
| Ours (GenLLM) | **53.67** | **60.14** | **63.65** | **65.34** | **72.85** |

Table 2: Performance comparison on the BC5CDR dataset across different dictionary sizes ($N$).

| Dictionary Size | 5 | 10 | 15 | 20 | 50 |
|---|---|---|---|---|---|
| org | 17.02 | 23.21 | 26.59 | 26.90 | 42.77 |
| LSMS | **28.18** | 32.81 | 39.71 | 41.82 | 48.28 |
| LLM-DA | 17.28 | 27.12 | 33.92 | 37.21 | 39.90 |
| NuNER | 10.37 | 17.88 | 11.56 | 11.76 | 22.78 |
| Ours (GenLLM) | 25.64 | **35.05** | **41.20** | **45.60** | **51.23** |

Table 3: Performance comparison on the TDMSci dataset across different dictionary sizes ($N$).

- **LLM-DA**: Utilizing large language model-based data augmentation at both the context and entity levels, with noise injection.
- **NuNER**: Fine-tuning the pretrained NuNER-v2.0 model on the gold annotations of the datasets.[2]

For all methods except NuNER, we use BERT-base-uncased as the base model. LSMS and LLM-DA are trained for 10 epochs on the combination of original training data and augmented data. NuNER is fine-tuned for 10 epochs on gold data.

#### 4.1.3 Proposed Method: GenLLM and Training Setup

Our proposed method, **GenLLM**, generates synthetic training data using LLM-based augmentation techniques. It employs prompt engineering with constraints to ensure data quality and entity control. Training follows a two-stage approach: we first pretrain the model on synthetic data for 3 epochs, then fine-tune on the gold-annotated data for 10 epochs. All experiments are conducted under reduced labeled data settings ($N = 5, 10, 15, 20, 50$ entities per type), simulating low-resource environments. We compare GenLLM's performance against the baselines introduced in Section 4.1.2.

Additional implementation details, including training hyperparameters and hardware specifications, are provided in Appendix A.1.

### 4.2 Results and Analysis

The performance comparison across different methods on the BC4CHEMD, BC5CDR, and TDMSci datasets is shown in Tables 1, 2, and 3, respectively.

Our method (GenLLM) consistently outperforms the org and NuNER, with significant improvements. On BC5CDR, GenLLM achieves the highest performance at all dictionary sizes, outperforming both LSMS and LLM-DA. On TDMSci, GenLLM shows strong performance, competing well with LSMS and LLM-DA, only losing to LSMS when N = 5. The low performance of GenLLM at smaller dictionary sizes on the BC4CHEMD dataset is likely due to the limited diversity and insufficient augmentation with only a few seed entities, which restricts the model's ability to generalize effectively. As the dictionary size increases, the synthetic data improves, leading to better performance. Overall, our method outperforms previous methods like LSMS and LLM-DA, offering a robust solution for low-resource settings by leveraging synthetic data generation for better generalization.

## 5 Experiment with High-Resource Setting

In this section, we evaluate our proposed method in a high-resource setting, where we utilize the full training data from three benchmark datasets: BC4CHEMD, BC5CDR, and TDMSci. This setting allows us to assess the performance of our approach when abundant annotated data is available, providing a direct comparison with conventional methods that rely on manually annotated corpora.

### 5.1 Experimental Setup

We conduct experiments using the full training datasets of BC4CHEMD, BC5CDR, and TDMSci. The models are trained using the standard dataset splits provided in prior studies to ensure comparability. All models are trained for 3 epochs. LSMS and LLM-DA also use the combination of original training data and augmented data generated by these methods. For our method (GenLLM), we

[2] https://huggingface.co/numind/NuNER-v2.0

| Dataset | BC4CHEMD | BC5CDR | TDMSci |
|---|---|---|---|
| Org | **87.19** | 83.27 | 55.19 |
| LSMS | 86.58 | **84.04** | 58.32 |
| LLM-DA | 86.58 | 82.40 | 52.79 |
| NuNER | 85.88 | 81.10 | 48.19 |
| Ours (GenLLM) | 86.85 | 83.74 | **58.70** |

Table 4: Performance (F1-score) comparison across different methods on the BC4CHEMD, BC5CDR, and TDMSci datasets on high-resource setting.

use the "pretraining" approach, first fine-tuning the model on synthetic data for 1 epoch, followed by fine-tuning on gold data for 3 epochs. For synthetic data, due to the cost of generating additional data, we reuse the data generated in the low-resource setting and combine the generated data of all sizes from that setting. Additional implementation details, including training hyperparameters and hardware specifications, are provided in Appendix A.1.

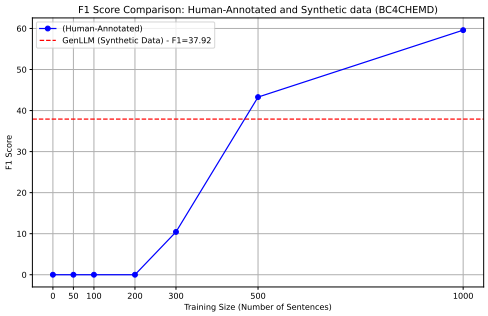
### 5.2 Results and Analysis

Table 4 presents the F1-score performance of various methods on the BC4CHEMD, BC5CDR, and TDMSci datasets in a high-resource setting. The experimental results in a high-resource setting show that different methods exhibit varying effectiveness across datasets. On BC4CHEMD, without augmentation (org) outperforms all other methods with an F1-score of 87.19, followed closely by GenLLM (86.85). LSMS and LLM-DA show similar performance, while NuNER lags slightly behind. On BC5CDR, LSMS achieves the highest F1-score (84.04), with GenLLM coming second (83.74), slightly outperforming LLM-DA and NuNER. GenLLM generally performs competitively or better than other methods in high-resource settings, with the best performance on TDMSci and close results on BC4CHEMD and BC5CDR. It becomes much harder to significantly improve performance with augmentation when the training data size is large, as seen in the BC4CHEMD and BC5CDR datasets.

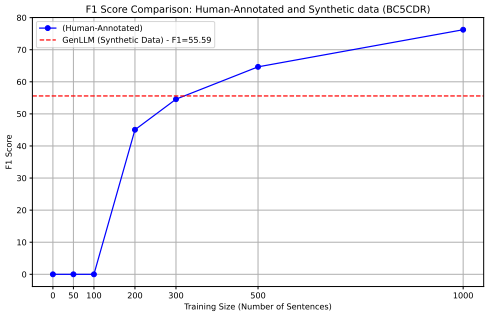
## 6 Analysis and Discussion

### 6.1 Quality of Synthetic Data

In this experiment, we investigate whether a model can be effectively trained using only synthetic data generated by LLMs, without any manually annotated data. The primary objective is to assess the feasibility of LLM-generated sentences as a standalone training resource in specialized domains.




(a) BC4CHEMD dataset.




(b) BC5CDR dataset.

Figure 2: F1 Score Comparison between BERT-base-uncased trained on human-annotated data and trained on synthetic data varying training sizes for the BC4CHEMD and BC5CDR datasets.

While synthetic data provides diversity, it may introduce hallucinated entities, ambiguous contexts, or annotation errors, leading to noisy supervision. Additionally, LLMs, trained on general-domain corpora, may struggle with domain-specific terminology, impacting performance.

#### 6.1.1 Data Sampling

**Human-Annotated** Setting: In this setting, we randomly select gold sentences from the training data of each dataset (BC4CHEMD and BC5CDR). These sentences are manually annotated and serve as the ground truth for model training.

**GenLLM** Setting: For the GenLLM approach, we provide the model with a list of 10 entities from each type in the dataset (e.g., chemicals, diseases) and instruct it to generate 1000 synthetic sentences. The goal is to use these synthetic sentences to train the model in the absence of human-annotated data. The models are then evaluated on the full test data of each dataset to assess their performance.

**Findings** Figures 2a and 2b help to understand the effectiveness of using LLM-generated synthetic data for training NER models and compare its per-

| Dictionary Size | 5 | 10 | 15 | 20 | 50 |
|---|---|---|---|---|---|
| Simple Prompt | 24.66 | **35.05** | 40.45 | **45.60** | 51.23 |
| + filter | 13.00 | 22.00 | 27.52 | 24.97 | 30.45 |
| + no-new-entity | **25.64** | 32.92 | **41.20** | 43.65 | 50.37 |
| + COT | 21.68 | 26.98 | 38.34 | 41.01 | **51.59** |

Table 5: F1 Score Comparison of Different Prompting Methods Across Training Sizes on TDMSci Test set.

| Error Type | Count |
|---|---|
| False Negative (Missing Entity) | 54 |
| False Positive (Spurious Entity) | 2 |
| Boundary Misalignment | 12 |

Table 6: Error analysis of 100 manually checked TDMSci samples.

formance to using human-annotated data. When fewer than 300-400 sentences are annotated, the synthetic data approach (from GenLLM) yields better performance. This suggests that synthetic data might be more effective in low-resource scenarios where manual annotation is costly or time-consuming, and small training sets are available. However, as the annotated data size grows beyond this point, human-annotated data consistently provides better results.

### 6.2 Different Prompting Methods

In this ablation study, we examine how different prompting strategies influence the quality of the generated synthetic data and the performance of the trained NER model. We evaluate the following four prompting methods:

- **Simple Prompt**: The model is provided with a plain list of entity names and their types, without any additional constraints or filtering (Figure 1a).
- **Simple Prompt + filter**: In this approach, we filter out generated sentences that introduce new entities not present in the seed list. This aims to ensure that only relevant entities appear in the synthetic data, reducing entity drift.
- **Simple Prompt + no-new-entity**: The prompt explicitly instructs the model to avoid introducing new entities beyond the provided list (Figure 3).
- **Simple Prompt + COT (Chain-of-Thought)**: The model is guided to generate sentences step-by-step, ensuring logical coherence and correct entity usage (Figure 4).

Table 5 presents the F1 scores for different prompting methods across various training sizes. The Simple Prompt baseline demonstrates strong performance, particularly at 10 and 20 training examples, where it achieves the highest scores (35.05 and 45.60, respectively). However, adding a filtering mechanism to remove sentences introducing new entities significantly reduces performance across all training sizes. This suggests that while filtering ensures strict entity control, it may also remove valuable diverse contexts that contribute to learning. The no-new-entity constraint, which instructs the LLM not to introduce unseen entities during generation, performs well in low-resource settings (5 and 15 examples), surpassing the Simple Prompt in these cases. The Chain-of-Thought (COT) prompting does not outperform the Simple Prompt in all training scenarios. It achieves its highest score (51.59) at 50 examples, which is slightly higher than the Simple Prompt's 51.23. These results highlight the trade-offs between entity control, data diversity, and reasoning-driven generation in synthetic data creation for NER.

### 6.3 Error Analysis

In the process of using LLMs for tasks such as NER and data generation, three common types of errors may arise: **False Negatives**, **False Positives**, and **Boundary Misalignment**. Understanding these errors is crucial for improving the accuracy and reliability of the generated sentences.

- **False Negatives (Missing Entities)** These occur when valid entities present in the sentence are not recognized or labeled by the model, resulting in under-annotation and potential loss of critical information.
- **False Positives (Spurious Entities)**
  In these cases, the model incorrectly identifies and labels non-entity spans as entities, introducing noise into the training data.
- **Boundary Misalignment**
  This error arises when the model detects the correct entity type but assigns incorrect boundaries—either extending beyond or falling short of the true entity span.

To assess the quality of LLM-generated data, we manually analyzed 100 synthetic samples from the TDMSci dataset. The results of this error analysis are summarized in Table 6. The most frequent issue was **False Negatives**, which occurred in 54 cases. These typically happened because the LLM introduced new entities not in the input seed list, which distracted it from labeling the original seed entities. **Boundary Misalignment** was observed in 12 cases. This is often caused by inconsistencies in annotation guidelines, such as whether to include the word "dataset" or "task" in the entity span for DATASET and TASK types. **False Positives** occurred in only 2 cases. They often resulted from noisy or questionable input annotations. For instance, in the TDMSci dataset, a seed entity like "Arabic-English" is labeled as a TASK, although its status as a task is ambiguous, potentially confusing the model into tagging unrelated terms.

Table 7 shows example outputs from LLM-based generation using seed entities from the biomedical domain, specifically the BC5CDR dataset. We manually identified three main types of annotation errors in these generated sentences. First, **False Negatives** occur when entities from the seed list are present in the generated text but are not labeled—such as "cocaine" and "tachycardia" in the first example. This often happens when the LLM adds new, related entities (e.g., "arrhythmia"), which distracts the model from tagging the original seed entities. Second, **False Positives** arise when entities not included in the input seed list are introduced and labeled in the output—such as "phenytoin" and "temporal lobe epilepsy." These hallucinated entities may be contextually relevant but were not intended to appear or be annotated in the generated data. This is especially problematic when using the generated sentences as training data in a low-resource setting, where tight control over entity inclusion is critical. Lastly, **Boundary Misalignment** is illustrated by the example where only "insulin" is labeled instead of the correct full span "insulin resistance." This type of span-level error can be caused by the LLM misunderstanding multi-word entity boundaries or inconsistencies in annotation granularity during training. These examples highlight how generation errors can propagate into downstream tasks if not properly filtered, and emphasize the importance of post-processing and validation when using LLMs for data augmentation in biomedical NER.

| **Seed Entities (Types) and Generated Sentence with Error Type** |
|---|
| **Seed Entities:** tachycardia (Disease), cocaine (Chemical)<br>**Generated Sentence:** Chronic cocaine use has been associated with an increased risk of developing tachycardia, a type of <Disease>arrhythmia</Disease>, which can lead to a range of cardiovascular complications, including myocardial infarction and stroke.<br>*(**False Negatives**: "cocaine" and "tachycardia" are not labeled. **False Positive**: "arrhythmia" is introduced and labeled as Disease but not in the seed list)* |
| **Seed Entities:** lidocaine (Chemical)<br>**Generated Sentence:** The administration of <Chemical>lidocaine</Chemical> has been studied for its potential to reduce <Disease>seizure frequency</Disease> in patients with <Disease>epilepsy</Disease>, and its efficacy in this context has been demonstrated through various clinical trials, including a randomized controlled trial that compared the outcomes of patients receiving <Chemical>lidocaine</Chemical> versus those receiving <Chemical>phenytoin</Chemical> for the treatment of <Disease>partial seizures</Disease> in adults with <Disease>temporal lobe epilepsy</Disease>.<br>*(**False Positives**: "phenytoin", "seizure frequency", "epilepsy", "partial seizures", "temporal lobe epilepsy" are not in the seed list but labeled. **False Negative**: "lidocaine" is correctly labeled, others are missing from seed list)* |
| **Seed Entities:** DSM - IV bipolar I (Disease), cisplatin (Chemical), nicotine (Chemical)<br>**Generated Sentence:** Research on the efficacy and safety of <Disease>DSM-IV bipolar I</Disease> (<Disease>bipolar disorder</Disease>) has shown that patients with this condition often require <Disease>medication</Disease> management, including the use of <Chemical>cisplatin</Chemical> and <Chemical>nicotine</Chemical> as part of their treatment regimen, which can lead to potential <Disease>side effects</Disease> such as nausea, vomiting, and increased heart rate.<br>*(**False Positives**: "bipolar disorder", "medication", and "side effects" are not in the seed list but labeled. Also reflects questionable factual correctness)* |
| **Seed Entities:** insulin resistance (Disease)<br>**Generated Sentence:** Patients with <Disease>insulin</Disease> resistance often exhibit symptoms of metabolic syndrome, including obesity and hypertension.<br>*(**Boundary Misalignment**: Only "insulin" is labeled, while the full entity should be "insulin resistance")* |

Table 7: Examples of generated sentences with seed entities from BC5CDR (Chemical, Disease), along with identified error types: False Negatives (missing entities), False Positives (spurious entities), and Boundary Misalignment (incorrect span).

## 7 Conclusion

In this study, we proposed GenLLM, a method for data augmentation in low-resource settings using large language models to generate synthetic sentences containing specific named entities. Our experiments demonstrated that GenLLM effectively enhances performance compared to baseline models like LSMS, LLM-DA, and NuNER when limited labeled data is available. By leveraging synthetic data generation with only seed entities, Gen-

LLM outperforms or complements state-of-the-art systems, especially in scenarios with constrained resources Furthermore, we explored the feasibility of training models using only synthetic data generated by LLMs, which proved to be effective in low-resource scenarios. However, human-annotated data still provided better results once the dataset size grew large enough, highlighting the importance of expert-annotated data in high-resource settings. GenLLM offers a promising solution for data augmentation in low-resource domains, particularly when manually annotated data is scarce. Future work can focus on further improving synthetic data quality and exploring additional augmentation strategies to enhance model generalization in diverse domains.

## Limitations

One potential limitation of this paper is that the quality of synthetic data generated by large language models (LLMs) may be inconsistent, potentially impacting model performance. To mitigate this, we ran each experiment three times and report the averaged results to ensure the robustness and generalizability of our findings. Additionally, this study focus on scienctific domains such as biomedical, chemical, and computer science, which may not generalize to other fields.

## Ethics Statement

This research adheres to ethical guidelines and practices throughout its execution. The datasets utilized in this study are publicly available and do not contain personally identifiable information. This paper was proofread with the assistance of OpenAI's GPT-4 language model to improve clarity and grammar. All substantive content and arguments are the author's own.

## Acknowledgement

This work was supported by the JSPS KAKENHI Grant Number 24K03231 and RIKEN AIP.

## References

Ilaria Bartolini, Vincenzo Moscato, Marco Postiglione, Giancarlo Sperlì, and Andrea Vignali. 2022. Cosiner: Context similarity data augmentation for named entity recognition. In *International Conference on Similarity Search and Applications*, pages 11–24. Springer.

Ilaria Bartolini, Vincenzo Moscato, Marco Postiglione, Giancarlo Sperlì, and Andrea Vignali. 2023. Data augmentation via context similarity: An application to biomedical named entity recognition. *Information Systems*, 119:102291.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11829–11841.

Shuguang Chen, Gustavo Aguilar, Leonardo Neves, and Thamar 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.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Himanshu Gupta, Kevin Scaria, Ujjwala Anantheswaran, Shreyas Verma, Mihir Parmar, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. 2023. Targen: Targeted data generation with large language models. *arXiv preprint arXiv:2310.17876*.

Nicolas Hiebel, Olivier Ferret, Karën Fort, and Aurélie Névéol. 2023. Can synthetic text help clinical named entity recognition? a study of electronic health records in french. In *The 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2021. Tdmsci: A specialized corpus for scientific literature entity tagging of tasks datasets and metrics. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 707–714.

Abdul Majeed Issifu and Murat Can Ganiz. 2021. A simple data augmentation method to improve the performance of named entity recognition models in medical domain. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 763–768. IEEE.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17.

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

Jian Liu, Yufeng Chen, and Jinan Xu. 2022. Low-resource ner by data augmentation with prompting. In *IJCAI*, pages 4252–4258.

Jiguo Liu, Chao Liu, Nan Li, Shihao Gao, Mingqi Liu, and Dali Zhu. 2023. Lada-trans-ner: adaptive efficient transformer for chinese named entity recognition using lexicon-attention and data-augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13236–13245.

Uyen Phan and Nhung Nguyen. 2022. Simple semantic-based data augmentation for named entity recognition in biomedical texts. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 123–129.

Caroline Sabty, Islam Omar, Fady Wasfalla, Mohamed Islam, and Slim Abdennadher. 2021. Data augmentation techniques on arabic data for named entity recognition. *Procedia Computer Science*, 189:292–299.

Arie Pratama Sutiono and Gus Hahn-Powell. 2022. Syntax-driven data augmentation for named entity recognition. *Proceedings of Pattern-based Approaches to NLP in the Age of Deep Learning*, page 56.

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

Usama Yaseen and Stefan Langer. 2021. Data augmentation for low-resource named entity recognition using backtranslation. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 352–358.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *arXiv preprint arXiv:2402.14568*.

Qian Yili and Xu Haonan. 2023. Datg: Data augmentation with transformer-based generation for low-resource named entity recognition. In *2023 China Automation Congress (CAC)*, pages 6188–6193. IEEE.

# A Appendix

## A.1 Experimental Setting Details

**Base NER model** We employ a fine-tuned BERT model for NER. The input sequences are first tokenized and then passed through BERT to obtain contextualized embeddings. These embeddings are fed into a linear classification layer followed by a softmax activation to predict the entity type of each token. For words that are split into multiple subwords during tokenization, only the embedding of the first subword is used for classification.

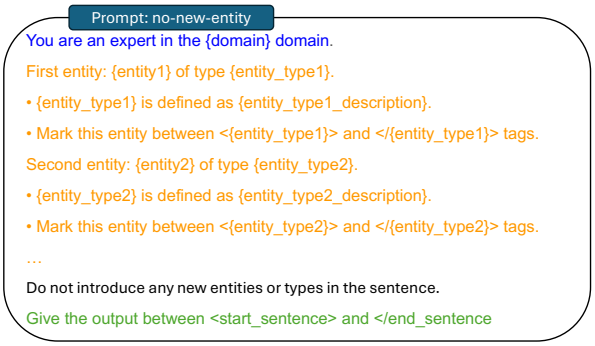



Figure 3: Example of the **Simple Prompt + no-new-entity** setup, where the prompt explicitly instructs the model to generate a sentence using only the provided entities and avoid introducing any new entities.

**Hyperparameters.** For all experiments, we use the following settings unless otherwise specified:

- **Learning rate:** 1e-4
- **Batch size:** 32
- **Optimizer:** AdamW
- **Max sequence length:** 256
- **Dropout rate:** 0.1
- **Weight decay:** 0.01

All models are implemented using the HuggingFace Transformers library. To ensure reproducibility, we fix the random seed to 42 across all components including NumPy, PyTorch, and HuggingFace Transformers. Training is conducted on a single NVIDIA V100 GPU with 32 GB of memory. Each run (including pretraining and fine-tuning steps) takes approximately 30–90 minutes depending on the dataset and the size of the training set.

## A.2 Prompts

### A.2.1 Prompt: no-new-entity

This prompt is a controlled variation of the **simple prompt**, extended with an explicit instruction: "`Do not introduce any new entities or types in the sentence.`" This modification aims to address a common issue in LLM-based data generation—**false negatives (missing entities)**—where the model may omit entities from the provided seed list or introduce incorrect ones, resulting in incomplete or misaligned annotations. By enforcing this constraint, we improve the alignment between the

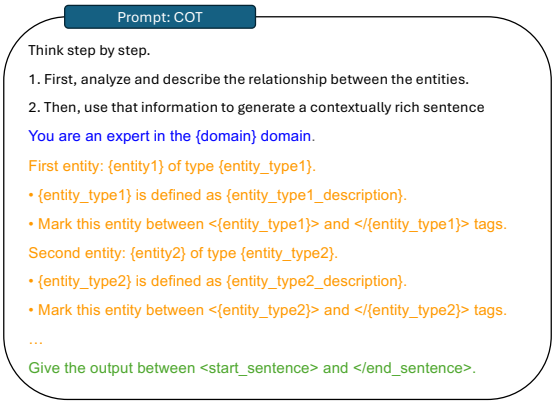



Figure 4: Example of the **Simple Prompt + COT (Chain-of-Thought)** setup, where the model is guided to generate the sentence in a step-by-step manner. This approach promotes logical coherence and helps ensure that the provided entities are used correctly in context.

prompt specification and the generated content, ensuring better coverage and fidelity to the intended entity set. An illustration of this prompt configuration is shown in Figure 3.

### A.2.2 Prompt: COT

Another variation is the **Simple Prompt + Chain-of-Thought (CoT)** setup, where the model is guided to reason step-by-step before producing the final sentence. This format encourages logical coherence and helps the model better understand and place the given entities in context. The intermediate reasoning steps can reduce annotation mistakes and improve entity boundary accuracy. An example of this prompt structure is shown in Figure 4.

## A.3 Datasets

We conduct experiments using three benchmark datasets for biomedical and scientific NER:

**BC4CHEMD** The BC4CHEMD dataset focuses on chemical entity recognition and is derived from biomedical abstracts. It contains over 30,000 sentences and nearly 900,000 tokens (see Table 11). The dataset features one entity type (CHEM), with 29,478 annotated chemical entities distributed across 14,529 sentences (see Table 12).

**BC5CDR** BC5CDR includes annotations for both chemical and disease entities, making it suitable for multi-type NER tasks. It comprises 4,560 sentences, with an average of 2.06 entities per sentence. Table 8 shows how entity coverage increases with larger subsets of annotated data, and general dataset statistics are shown in Table 11. Additional details on total entities and sentence coverage per type are listed in Table 12.

**TDMSci** TDMSci is a scientific NER dataset that includes three entity types: Task, Dataset, and Metric. It contains 1,523 sentences and is more diverse than the biomedical datasets in terms of entity types and structure (see Table 10 and Table 11). Table 12 further breaks down the number of entities and sentence distributions per type.

To simulate low-resource conditions, we create reduced versions of each dataset by limiting the number of unique entities used for training. These settings vary from 5 to 500 entities per type, as detailed in Tables 8, 9, and 10. These subsets are used in conjunction with our "Limited Dictionary" setup to test the effectiveness of data augmentation strategies.

| Dataset Size | Chemical | Disease |
|---|---|---|
| 5 | 19 | 20 |
| 10 | 36 | 41 |
| 15 | 54 | 60 |
| 20 | 73 | 72 |
| 50 | 192 | 178 |
| 100 | 364 | 340 |
| 200 | 736 | 666 |
| 300 | 1146 | 985 |
| 400 | 1516 | 1336 |
| 500 | 1868 | 1665 |

Table 8: Entity counts per entity type for BC5CDR dataset.

| Dataset Size | CHEM |
|---|---|
| 5 | 16 |
| 10 | 29 |
| 15 | 41 |
| 20 | 57 |
| 50 | 160 |
| 100 | 344 |
| 200 | 669 |
| 300 | 972 |
| 400 | 1285 |
| 500 | 1618 |

Table 9: Entity counts per entity type for CHEMDNER dataset.

| Dataset Size | DATASET | METRIC | TASK |
|---|---|---|---|
| 5 | 6 | 13 | 14 |
| 10 | 17 | 27 | 28 |
| 15 | 24 | 43 | 43 |
| 20 | 31 | 60 | 59 |
| 50 | 103 | 155 | 144 |
| 100 | 198 | 307 | 292 |
| 200 | 400 | 553 | 539 |
| 300 | 591 | 619 | 805 |
| 400 | 700 | 670 | 1056 |
| 500 | 732 | 681 | 1207 |

Table 10: Entity counts per entity type for TDMSci dataset.

Table 11: Dataset Statistics for NER Tasks

| Dataset | #Sentences | #Tokens | #Entity Types | Avg. Entities/Sent. | #Sent. w/o Entities |
|---|---|---|---|---|---|
| **BC4CHEMD** | 30,812 | 872,932 | 1 (CHEM) | 0.96 | 16,283 |
| **BC5CDR** | 4,560 | 118,170 | 2 (Chemical, Disease) | 2.06 | 753 |
| **TDMSci** | 1,523 | 49,460 | 3 (TASK, DATASET, METRIC) | 1.43 | 330 |

Table 12: Entity-Specific Statistics

| Dataset | Entity Type | #Entities | #Sentences w/ Entities |
|---|---|---|---|
| **BC4CHEMD** | CHEM | 29,478 | 14,529 |
| **BC5CDR** | Chemical | 5,203 | 2,951 |
| | Disease | 4,182 | 2,658 |
| **TDMSci** | TASK | 1,219 | 920 |
| | DATASET | 420 | 322 |
| | METRIC | 536 | 358 |