Reviving Iterative Refinement in Diffusion-based NER with an Initializer-Restorer Approach

Long Hai Trieu[1], Hieu Minh Phi[2], Makoto Miwa[3, 2]
[1]Cancer Research UK Cambridge Institute, University of Cambridge
[2]National Institute of Advanced Industrial Science and Technology
[3]Toyota Technological Institute
hai-long.trieu@cruk.cam.ac.uk, hieu.phi@aist.go.jp, makoto-miwa@toyota-ti.ac.jp

Abstract

Diffusion models have introduced a generative paradigm for Named Entity Recognition (NER), formulating the task as refining entity spans from noise. While promising, our analysis on the ACE2004 dataset reveals a limitation when training with Exponential Moving Average (EMA): the model performance often peaks at a single inference step ($\gamma = 1$) and plateaus or degrades with additional steps. This suggests that under standard stable training configurations, the model may function primarily as a one-step generator rather than leveraging the iterative refinement capability characteristic of diffusion models. To address this, we propose an **Initializer-Restorer** approach. Instead of initializing the reverse process from random Gaussian noise, we utilize a preliminary set of candidate spans generated by a standard NER model (e.g., BERT or GLiNER). This allows the diffusion model to start from an informed, diverse prior, enabling effective iterative restoration. We investigate different training strategies for the restorer and find that a hybrid strategy mixing ground truth and noisy predictions is essential. Experiments on ACE2004, GENIA, and CleanCoNLL show that our approach improves performance over the baseline, particularly when multiple restoration steps are employed. For instance, on CleanCoNLL, our method achieves an F1 score of 94.70%, compared to 93.79% for the baseline. Our code is available at https://github.com/longtrieu-ai/Initializer-Restorer-NER.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in Information Extraction, traditionally addressed via sequence labeling (Devlin et al., 2019) or span classification (Sohrab and Miwa, 2018). Recently, generative approaches have gained traction, utilizing Sequence-to-Sequence models (Yan et al., 2021) or Large Language Models (LLMs) to generate entity sequences directly. Among these, DiffusionNER (Shen et al., 2023) proposed a novel paradigm by modeling NER as a boundary diffusion process. By treating entity spans as continuous coordinates generated from a Gaussian distribution, diffusion models offer the theoretical advantage of iteratively refining predictions, capturing complex inter-entity dependencies through a multi-step denoising process.

In this work, we revisit the inference dynamics of diffusion-based NER. Our empirical analysis indicates that when trained with Exponential Moving Average (EMA), which is a standard practice to stabilize diffusion model training, the baseline model achieves optimal performance with a single reverse step ($\gamma = 1$). While EMA is beneficial for overall training stability, this behavior suggests that the iterative denoising process, which is central to the diffusion framework, may not be fully utilized when starting from pure noise. The model effectively learns a direct mapping from noise to entities, rendering additional computational steps redundant or even detrimental.

To better leverage the refinement capability of diffusion models, we propose an **Initializer-Restorer** architecture. Our method consists of two stages: (1) An **Initializer**, which can be a standard NER model, provides an initial set of noisy candidate spans. (2) A **Restorer**, implemented as a conditional diffusion model, starts the reverse process from these candidates rather than random noise. By initializing with a structured prior, the Restorer can focus on local adjustments and error correction. We further explore training strategies for the Restorer, comparing training on ground truth spans versus training on the Initializer's predictions.

We evaluate our approach on three standard benchmarks: ACE2004, GENIA, and CleanCoNLL (Rücker and Akbik, 2023). We include the re-implementation of the original DiffusionNER as a strong baseline to ensure a fair comparison

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 877–882
July 2-7, 2026

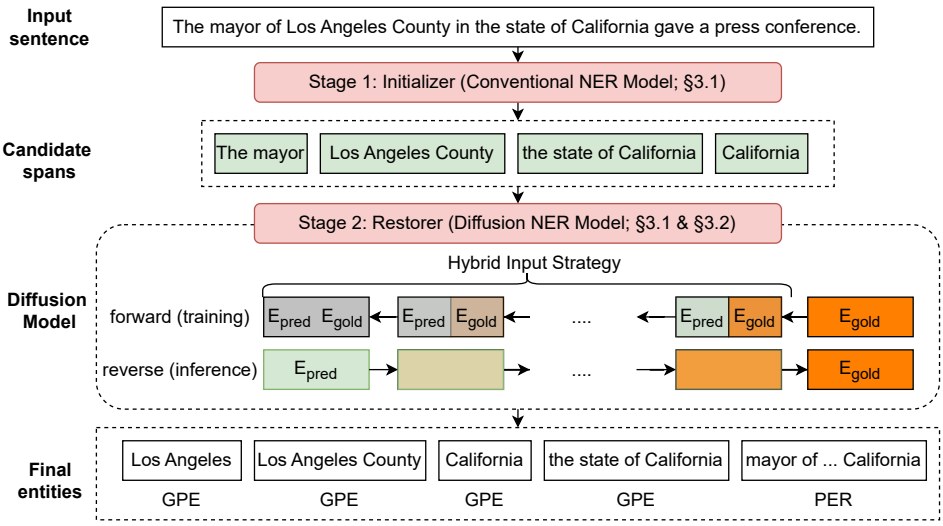

Figure 1: Overview of the proposed initializer-restorer framework. An input sentence is first processed by the Initializer (a conventional NER model), which produces a set of candidate entity spans. The Restorer, a diffusion model, takes these candidate spans as input and iteratively denoises and corrects them over γ steps to produce the final, more accurate set of entities. E_{gold} (orange) denotes ground truth spans, which become increasingly noisy (gray) during the forward process. E_{pred} (green) represents the Initializer's predictions.

under identical experimental conditions. Our results demonstrate that initializing with candidates from BERT or GLiNER consistently improves F1 scores. Furthermore, unlike the baseline, our Restorer exhibits performance gains as the number of inference steps increases, confirming that the proposed initialization strategy revives the effectiveness of iterative refinement.

2 Related Work

2.1 Generative NER

Generative approaches to NER have diversified beyond traditional sequence labeling. Seq2Seq models like BART and T5 have been adapted to generate entity sequences or pointers (Yan et al., 2021). These models model the joint probability of entities directly but can suffer from error propagation during decoding. DiffusionNER (Shen et al., 2023) introduced a continuous diffusion process to NER, treating entity identification as a coordinate generation task. This allows for parallel generation of entities and iterative refinement. Our work builds directly on this diffusion framework, identifying and addressing limitations in its refinement behavior under practical training setups.

2.2 Diffusion Models in NLP

Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020) have revolutionized image generation and are increasingly applied to NLP. However, applying diffusion to discrete text is challenging. Approaches include embedding continuous diffusion (Gong et al., 2023) or discrete diffusion processes. DiffusionNER bypasses the discreteness issue by diffusing on span coordinates rather than token embeddings. While successful, the efficiency and necessity of the multi-step process in this specific domain have not been fully scrutinized. Our work investigates the trade-off between one-shot generation and iterative refinement in this context.

3 Method

Our proposed method is a two-stage, Initializer-Restorer framework, as depicted in Figure 1. This modular design separates the task into an initial proposal stage and a subsequent restoration stage.

3.1 Initializer-Restorer framework

Stage 1: Initializer We employ a non-diffusion NER model to estimate the entities. Let the input sentence be S. The Initializer outputs a set of predicted spans E_{init}. We consider two Initializers:

- **BERT-Tagger:** A BERT-large model (Devlin et al., 2019) fine-tuned for sequence labeling or span classification[1]. This provides a strong, standard baseline initialization.

[1] `dslim/bert-base-NER`

- **GLiNER**: A generalist NER model (Zaratiana et al., 2024)[2] capable of identifying diverse entity types. GLiNER is particularly effective at recalling varied entities, potentially providing a richer set of candidates. This provides a task-agnostic initialization, demonstrating that a single generalized model can serve as an effective prior across diverse datasets.

Stage 2: Restorer The Restorer is a conditional diffusion model that takes E_{init} as a structured prior, rather than starting from pure noise ($E_T \sim \mathcal{N}(0, I)$), where T is the number of timesteps, to generate entities. In the forward (training) process, we progressively add Gaussian noise to the target spans. In the reverse (inference) process, the Restorer iteratively predicts the denoised coordinates at each step, starting from E_{init}, and improves them over γ restoration steps.

3.2 Restorer Training Strategies

A critical design choice is how to train the Restorer to handle the Initializer's outputs. We explored three strategies:

1. **Ground Truth Only (GT-Only):** The Restorer is trained purely as a standard diffusion model using ground truth spans $\mathbf{x}_0$ with added Gaussian noise. During inference, it is applied to the Initializer's output. This creates a mismatch between training (Gaussian noise) and inference (structural noise from Initializer).

2. **Prediction Only (Pred-Only):** The Restorer is trained using only the noisy predictions from the Initializer as the starting point. This aligns the training and inference distributions but risks overfitting to the specific error patterns of the Initializer, potentially limiting generalization.

3. **Hybrid Input Strategy (Ours)**: To align training with inference, we construct the input by mixing: (a) **Ground Truth Spans** to encourage the retention of correct entities, (b) **Initializer Predictions** to learn boundary and classification corrections specific to the Initializer's error patterns, and (c) **Random Gaussian Noise** to learn how to filter out false positives. This mixture ensures the **Restorer** can robustly handle various error patterns.

[2] `gliner-community/gliner_small-v2.5`

Steps (γ)	**1**	**2**	**5**	**10**
Baseline (w/ EMA)	**88.51**	88.12	87.92	87.92
Ours (BERT Init)	85.74	88.34	88.59	**88.65**

Table 1: F1 Score (%) of the Baseline DiffusionNER and the proposed method on the ACE2004 development set with varying inference steps.

Training Strategy	**F1 Score (%)**
GT-Only	85.5
Pred-Only	87.8
Hybrid (Ours)	**88.62**

Table 2: Comparison of training strategies on ACE2004 (BERT+GliNER Init). The Hybrid strategy yields the best performance.

3.3 Implementation Details

For the diffusion backbone, we use BERT-large for ACE2004 and CleanCoNLL, and BioBERT-large for the biomedical GENIA dataset. We set the number of candidate entity boxes to $k = 180$ to ensure sufficient coverage for sentences with dense entities. We utilize Exponential Moving Average (EMA) with a decay rate of 0.9999 for model parameters to stabilize training. We use a cosine noise schedule and train on an NVIDIA H200 GPU.

4 Experiments

4.1 Setup

Datasets: We evaluate on **ACE2004** (Doddington et al., 2004) (Nested NER), **GENIA** (Kim et al., 2003) (Biomedical, Nested), and **CleanCoNLL** (Rücker and Akbik, 2023) (Flat NER). CleanCoNLL is a corrected version of CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), providing a more reliable evaluation benchmark by fixing annotation errors in the original dataset. Statistics for these datasets are provided in Appendix B. We report the micro-averaged entity-level F1 score.

Baselines:

- **Baseline (Re-run):** We re-implemented DiffusionNER with the same backbone (BERT-large) and EMA settings (decay=0.9999). This serves as a controlled comparison to isolate the effect of our framework. Following the default setting of Shen et al. (2023), we report baseline performance at $\gamma = 5$ to ensure a consistent comparison.

Model	ACE2004			GENIA			CleanCoNLL		
	P	R	F1	P	R	F1	P	R	F1
Initializer (BERT)	93.90	53.23	67.94	-	-	-	91.65	88.75	90.18
Initializer (GLiNER)	72.31	23.93	35.96	81.42	70.27	75.43	76.80	66.38	71.21
Reference: Original Paper†	88.11	88.66	88.39	82.10	80.97	81.53	-	-	-
Baseline (Re-run)	88.30	88.27	88.28	81.48	81.42	81.45	94.49	93.10	93.79
Ours (BERT Init)	**88.75**	88.48	**88.62**	-	-	-	94.39	94.34	94.37
Ours (GLiNER Init)	88.65	**88.49**	88.57	**88.65**	**81.96**	**81.69**	**95.04**	**94.36**	**94.70**

Table 3: Main results (Precision, Recall, F1 in %) on ACE2004, GENIA, and CleanCoNLL test sets. All metrics are micro-averaged. Baseline uses $\gamma = 5$ following Shen et al. (2023); their reported scores are also included for reference (†). Ours uses the Hybrid Input Strategy with the respective initializers.

- **Original DiffusionNER:** We refer to the scores reported in the original paper (Shen et al., 2023) for context, though direct comparison is complicated by differences in environment and dataset splits.

- **Initializer Baselines:** We also report the performance of the Initializer models (BERT-Tagger, GLiNER) alone to quantify the gain achieved by the Restorer.

Due to computational constraints, our reported results are based on a single random seed. While we acknowledge the potential for run-to-run variance, the consistent performance trends observed across multiple datasets support the validity of our approach.

4.2 Preliminary Study on EMA

We first analyze the behavior of the standard DiffusionNER model under stable training conditions. We evaluated the baseline (BERT-large backbone) on the ACE2004 dataset while varying the number of inference steps γ. As shown in Table 1, the F1 score peaks at $\gamma = 1$ (88.51%) and decreases as steps increase to 2 (88.12%) and 10 (87.92%). This trend suggests that with EMA, the model converges to a state where it performs "one-shot" generation from noise effectively, but the iterative trajectory does not contribute to further refinement. While this observation was primarily validated on ACE2004, it highlights a potential limitation in how multi-step refinement behaves under stable EMA training.

4.3 Comparison of Training Strategies

We evaluate the impact of the different training strategies described in Section 3.2 on the ACE2004 dataset. Table 2 shows the results. The **GT-Only** strategy performs poorly, likely due to the distribution shift between Gaussian noise (training) and Initializer output (inference). The **Pred-Only** strategy improves over GT-Only but is surpassed by the **Hybrid** strategy. The Hybrid strategy achieves the highest F1 score (88.62%), demonstrating the importance of exposing the model to a diverse range of input qualities during training.

4.4 Main Results

Table 3 summarizes the results on the test sets. We compare our method (Hybrid strategy) against the Baseline (Re-run) and the Initializer alone.

Comparison with Baseline DiffusionNER Our approach consistently outperforms the re-run baseline across all datasets. On CleanCoNLL, utilizing GLiNER as an initializer achieves 94.70%, an improvement of nearly 1.0 percentage point over the baseline (93.79%). For ACE2004, the BERT-initialized model reaches 88.62%, surpassing the baseline's 88.28%. These results suggest that a warm start effectively guides the diffusion process to a better solution.

Improvement over Initializer We evaluate the gain provided by the Restorer by comparing against the Initializer alone, using CleanCoNLL as the primary benchmark. For the GLiNER Initializer, the standalone F1 is 71.21%. Despite this moderate initial F1, the Restorer dramatically boosts the final F1 to 94.70%. This demonstrates that the Restorer is not merely filtering candidates; it effectively recovers missing entities and corrects span boundaries, compensating for the Initializer's limitations. Similarly, for the **BERT Initializer**, which is already a strong baseline with an F1 of 90.18%, the Restorer further improves performance to 94.37%.

4.5 Reviving Iterative Refinement

We examine whether our approach successfully utilizes the diffusion steps. Table 1 compares the F1 scores on ACE2004 across different γ. Unlike the Baseline which peaks at $\gamma = 1$, our model (BERT Init) starts at 85.74% and improves to 88.59% at $\gamma = 5$ and 88.65% at $\gamma = 10$. This confirms that initializing with candidate spans allows the model to perform actual restoration, correcting boundaries and labels over multiple steps.

Qualitative Error Analysis Manual inspection of the predictions reveals that the Restorer excels at two specific types of corrections. First, it effectively shifts imprecise entity boundaries proposed by the Initializer. Second, it successfully drops false positives generated by the Initializer by denoising them back into empty spans.

5 Conclusion

We investigated the behavior of diffusion-based NER models under EMA training and found that they often act as single-step generators. To address this, we proposed an Initializer-Restorer approach that starts the diffusion process from candidates generated by a standard NER model. Our method revives the iterative refinement capability of diffusion models and achieves improved performance across three benchmarks.

Limitations

While our framework successfully revives iterative refinement in diffusion-based NER, several limitations remain.

First, our study is limited to English datasets; the applicability to multilingual or low-resource languages requires further verification.

Second, the two-stage inference process (**Initializer-Restorer**) inevitably increases computational overhead. Empirically, the multi-step diffusion refinement incurs an inference latency noticeably higher than that of a single-step baseline, depending on the chosen number of reverse steps γ.

Finally, we observed that DeBERTa-v3-large underperformed BERT-large in this framework (see Appendix A), which we leave as an open question for future investigation, noting it here simply as an observed negative result. Additionally, systematically tuning the mixture ratio within the Hybrid strategy, exploring alternative noise schedules or stable training without EMA, and comparing against simpler non-diffusion refinement baselines remain promising directions for future work.

Ethics Statement

This work involves the development of information extraction models using publicly available datasets (ACE2004, GENIA, CleanCoNLL). We do not use any private or personally identifiable information not already present in these standard benchmarks. The proposed method aims to improve the efficiency and accuracy of NER systems, which can be beneficial for downstream applications such as knowledge graph construction and document analysis. We acknowledge the potential environmental impact of training large diffusion models and have employed techniques like EMA and efficient initialization to mitigate unnecessary computation.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback. We acknowledge the use of Google Gemini for language editing during manuscript preparation; all research design, experimentation, and analysis were conducted by the authors.

This work is based on results from project JPNP25006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). Experiments were conducted using ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use". This work is also supported by Cancer Research UK Cambridge Institute Core Award.

References

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.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, volume 33, pages 6840–6851.

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

Susanna Rücker and Alan Akbik. 2023. CleanCoNLL: A nearly noise-free named entity recognition dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8628–8645, Singapore. Association for Computational Linguistics.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

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

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

A Impact of Backbone Model

We compared DeBERTa-v3-large (He et al., 2023) and BERT-large as backbones. Table 4 shows that DeBERTa-v3-large consistently underperforms BERT-large (e.g., 77.05% vs 88.28% on ACE2004). We leave the exact cause of this architectural incompatibility as an open question for future investigation, noting it here simply as an observed negative result.

Dataset	DeBERTa-v3	(Bio)BERT-Large
ACE2004	77.05	**88.28**
GENIA	76.53	**81.45**
CleanCoNLL	89.35	**93.79**

Table 4: F1 score comparison (%) between DeBERTa-v3-large and BERT-large backbones (Baseline, $\gamma = 5$).

B Dataset Statistics

Table 5 provides statistics for the datasets used in our experiments.

Dataset	ACE04	GENIA	CoNLL
Domain	News	BioMed	News
Nested	Yes	Yes	No
# Types	7	5	4
# Train	6,202	15,023	14,041
# Dev	808	1,669	3,250
# Test	819	1,854	3,453

Table 5: Statistics of the datasets (counts refer to the number of sentences).