A Scalable Framework for Automated NER Annotation Correction in Low-Resource Languages

Toqeer Ehsan[1] and Thamar Solorio[2]
[1]Quantitative Science and Technology Studies (QSTS),
VTT Technical Research Centre of Finland Ltd., Espoo, Finland
[2]Mohamed bin Zayed University of Artificial Intelligence (MBZUAI),
Masdar City, Abu Dhabi, United Arab Emirates
Correspondence: toqeer.ehsan@vtt.fi

Abstract

Poor quality or noisy annotations in Named Entity Recognition (NER), as in any other NLP task, make it challenging to achieve state-of-the-art performance. In this paper, we present a multi-step framework to enhance the annotation quality of NER datasets by employing automated techniques. We propose a frequency-based iterative approach that leverages self-training and a dual-threshold mechanism to enhance inference confidence. Experimental evaluations on different NER datasets demonstrate significant improvements in NER performance with respect to the original datasets. This work further explores the potential of generative Large Language Models (LLMs) to perform NER for low-resource languages.

1 Introduction

With advancements in generative LLMs, multilingual models are now capable of performing several NLP tasks including NER. However, their performance on existing NER benchmarks, such as CoNLL-03 (Sang and De Meulder, 2003) and OntoNotes 5.0 (Weischedel et al., 2013), remains moderate compared to supervised learning approaches (Chen et al., 2023; Guo et al., 2024; Ye et al., 2023). The performance of supervised NER approaches is heavily dependent on the quality of the datasets, underscoring the need for well-annotated and improved datasets (Mayhew et al., 2023).

Table 1 presents the statistics of the training sets of three NER datasets; MK-PUCIT (Kanwal et al., 2019), Shahmukhi (Ahmad et al., 2020; Tehseen et al., 2023), and SiNER (Ali et al., 2020) for Urdu, Shahmukhi (Western Punjabi), and Sindhi languages. Despite the large number of sentences, these datasets exhibit a significant amount of missing annotations, leading to reduced model performance on the NER task. To gain a preliminary understanding of the extent of missing annotations in the datasets, we conducted an initial analysis by filling in potential missing entity mentions (see the estimates in Table 1). This was achieved through context-free mapping of named entities extracted from the MK-PUCIT dataset onto all other training sets, including MK-PUCIT itself. This approach provided an estimate of the annotation gaps and highlighted the scale of the problem. Manual revision and re-annotation, which are labor-intensive tasks requiring expertise and resources, are not feasible for datasets with such a large number of samples. These challenges highlight the need for automated approaches that can detect and fill missing annotations, improving the overall quality and completeness of these datasets.

NE Type	Original NEs	Updated NEs	Increase (%)
MK-PUCIT			
PER	10,486	11,965	12.40%
LOC	19,868	23,880	16.80%
ORG	5,277	8,665	39.10%
Total NEs	35,631	44,510	19.90%
# Sents.	24,080	–	–
Shahmukhi			
PER	4,609	4,732	2.60%
LOC	1,852	2,160	14.30%
ORG	526	644	18.30%
Total NEs	6,987	7,536	7.30%
# Sents.	13,412	–	–
SiNER			
PER	12,720	13,131	3.13%
LOC	14,136	15,029	5.94%
ORG	1,342	3,380	60.29%
Total NEs	28,198	31,540	10.60%
# Sents.	31,612	–	–

Table 1: Estimated number of named entities (PER, LOC, ORG) in the original training sets, the counts after filling missing annotations, and the percentage increase in named entities after insertion.

We propose a frequency-based iterative tech-

Findings of the Association for Computational Linguistics: EACL 2026, pages 4138–4151
March 24-29, 2026 ©2026 Association for Computational Linguistics

nique to correct missing annotation errors using a self-training mechanism. To ensure reliable predictions, a dual confidence threshold is imposed on logits and self-attention scores to reduce the risk of propagating erroneous annotation. Multiple annotation candidates are generated for each sentence using gold and pseudo-labels, and a multilingual NER model is employed as a validator to select the most plausible candidate based on the F_1 score. Before employing the self-training model, all three datasets were enhanced automatically to ensure correct word segmentation and named entity augmentation for frequent entity mentions.

The performance of NER on the corrected datasets is evaluated against baseline (original) datasets by fine-tuning the multilingual XLM-RoBERTa-large (Conneau et al., 2019; Liu et al., 2019) model. The corrected datasets demonstrate performance improvements of 3.96, 1.4 and 1.44 micro F_1 points for MK-PUCIT, Shahmukhi and SiNER, respectively. Our approach effectively integrates the robustness of multilingual NER models with iterative self-training to enhance annotation quality of low-resource NER datasets. The main contributions of this work are as follows:

- We propose a frequency-based iterative technique that effectively corrects missing annotation errors in NER datasets

- We demonstrate the capability of causal LLMs to improve the quality of non-English datasets.

- We provide insights into the potential of causal LLMs for NER in low-resource languages by employing an in-line NER labeling method.

- We prepare accurate validation and test sets for all three datasets, ensuring unbiased and reliable performance assessments.

2 Related Work

2.1 High-Resource NER Correction

Significant efforts have been made to develop advanced NER systems, while comparatively less attention has been paid to improving the quality of the annotation of the available datasets on which these models are based (Bernier-Colborne and Vajjala, 2024). The quality of annotated data has a strong impact on NER performance. Recent research has focused on cleaning and correcting high-resource corpora such as OntoNotes 5.0 (Weischedel et al., 2013), ANERcorp (Benajiba et al., 2007), and CoNLL-03 (Sang and De Meulder, 2003). Bernier-Colborne and Vajjala (2024) addressed annotation errors in OntoNotes 5.0, a major English NER dataset, by identifying errors and applying automated correction rules. Their re-annotated dataset improved overall F-scores but required substantial manual effort. Similarly, Al-Duwais et al. (2024) introduced CLEANANERCorp, a refined version of ANERcorp for Arabic NER. Using CLEANLAB (Wang and Mueller, 2022), they detected mislabeled entities and manually re-annotated them with refined guidelines. Rueda et al. (2024) introduced CoNLL#, a revised version of the CoNLL-03 English test set to support fine-grained error analysis. In contrast, Rücker and Akbik (2023) released CleanCoNLL, a semi-automatic relabeling of CoNLL-03 with updated and more consistent NER labels across train, dev, and test, assisted by entity linking and automatic consistency checks, followed by iterative cross-checking with manual inspection and correction. These approaches demonstrate that manual, semi-automatic, consistency-checked corrections can improve datasets, but they remain applicable mainly to well-resourced languages. *Limitations:* High-resource correction approaches generally assume availability of external resources like Wikipedia and expert annotators. Consequently, they are difficult to apply to low-resource languages, where annotated data and supporting resources are scarce.

2.2 LLM-Assisted Annotation

Yao et al. (2024) compared human and LLM-generated annotations on a Chinese address NER dataset, finding that LLMs performed comparable to humans for building-level entities but underperformed for other entities. Naraki et al. (2024) combined manual annotation with ChatGPT few-shot prompting in subsets of CoNLL03 and WikiGold (Balasuriya et al., 2009), showing that careful prompt design can reduce human effort, but still requires manual verification. Tan et al. (2024) surveyed LLM-aided annotation and synthesis strategies, and Zhang et al. (2023) proposed treating LLMs as “active annotators”. These studies highlight the promise of LLMs to accelerate annotation, but also underline the need for careful calibration and human oversight. *Limitations:* Existing

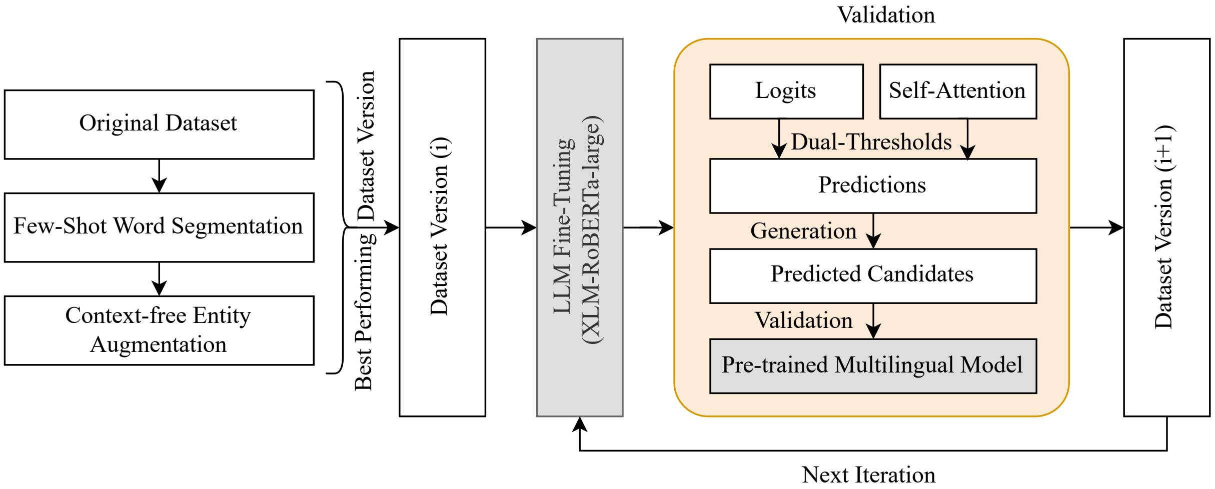

Figure 1: Framework for NER Dataset Correction: An iterative process leveraging word segmentation correction, context-free entity augmentation, and iterative self-training with dual-threshold validation to enhance the annotation quality.

LLM-assisted methods require prompt engineering and manual annotation verification. Moreover, the computational cost and less exposure to low-resource regional languages limit their scalability.

2.3 Low-Resource NER Correction

For low-resource languages, most work focuses on cross-lingual transfer or data augmentation. Cross-lingual and multilingual transfer techniques exploit character-level models or multilingual embeddings to improve low-resource NER (Cotterell and Duh, 2024; Torres et al., 2024). Cross-lingual data augmentation helps to improve NER performance for low-resource languages (Ehsan and Solorio, 2025). Liu et al. (2025) performed low-resource relation extraction and introduced logical rules to guide LLMs in producing high-confidence pseudo-labels achieving improved results. *Limitations:* These methods improve the robustness of the model but do not fill the annotation gaps. Distant supervision yields noisy labels, synthetic data can be culturally implausible, and cross-lingual augmentation introduces new samples without resolving missing annotation.

3 Multi-Step Framework

The languages selected in this work are topologically related and culturally similar. In terms of named entities, they share similar names, locations and organizations. Given these similarities, cross-lingual representation could be helpful in improving the performance of NER for the regional languages. However, all three datasets contain a substantial amount of missing annotations, making it challenging for supervised models to achieve state-of-the-art performance. To address this issue, we propose a dataset correction framework that incorporates word segmentation correction, context-free named entity augmentation, and a frequency-based iterative self-training mechanism shown in Figure 1. Following sections describe the steps of our proposed approach.

3.1 Word Segmentation

Urdu, Shahmukhi, and Sindhi are written in Perso-Arabic script, which is a cursive script (Consortium, 2024). A character may have one of four positions within a token; initial, middle, final, or independent (Zia et al., 2018; Ali et al., 2024). Characters are further categorized into joiners and non-joiners. Joiner characters are joined with the preceding character, while non-joiners are not. Importantly, the space character is not mandatory; instead, it is used to keep the correct shape of tokens.

Different writers, sources, and keyboards generate varying character sequences for the same content. LLM performance depends on tokenization, and inconsistencies can significantly impact model accuracy (Ovalle et al., 2024; Kudo, 2018; Schmidt et al., 2024). All three datasets exhibit word segmentation errors, further complicating supervised NER model performance. Figure 2 illustrates an MK-PUCIT example with incorrect segmentation and its XLM-RoBERTa-large tokenization.

There are three space inconsistencies between original and corrected sentences in Figure 2, that

Original Sentence (14 words)

وہ میوہ جات کے خاندانی کارو بار کے سلسلے میں کرا چی جایا کرتےتھے

(he, fruits, of, family, business, of, for, in, Karachi, go, used, to)

(He used to go to Karachi in connection with the family fruit business)

Corrected Sentence (13 words)

وہ میوہ جات کے خاندانی کاروبار کے سلسلے میں کراچی جایا کرتے تھے

Word Segmentation Issues

Karachi (LOC): کرا چی vs کراچی, Business: کارو بار vs کاروبار,
Used to: کرتےتھے vs کرتے تھے

SentencePiece Tokenization from XLM-RoBERTa-large

Original Sentence (21 tokens)

['_وه', '_می', 'وه', '_جا', 'ت', '_کے', '_خاندان', 'ی', '_کار', 'و',
'_بار', '_کے', '_سلسلے', '_میں', '_کرا', '_چی', '_جای', 'ا',
'_کرتے', 'ت', 'ھے']

Corrected Sentence (17 tokens)

['_وه', '_می', 'وه', '_جا', 'ت', '_کے', '_خاندان', 'ی', '_کاروبار',
'_کے', '_سلسلے', '_میں', '_کراچی', '_جای', 'ا', '_کرتے', '_تھے']

Figure 2: An example from the MK-PUCIT dataset demonstrating the effect of inconsistent word segmentation on tokenization for transformer-based models.

produce different tokenization from the SentencePiece tokenizer (Kudo and Richardson, 2018). The original sentence from the MK-PUICT produces 21 tokens, whereas the corrected sentence has 17 tokens. Along with others, there is a notable tokenization inconsistency in the named entity Karachi (LOC) due to incorrect word segmentation in the original dataset. This highlights that accurate word segmentation is crucial for consistent tokenization during the training and fine-tuning of LLMs.

To address word segmentation errors across all three datasets, we employed ChatGPT-4o (May 2024 version) using few-shot learning approach. We used the OpenAI API[1] to prompt the model to correct word segmentation while preserving the NER annotations. For this purpose, the CoNLL format was transformed into an in-line annotation format, inspired by Paolini et al. (2021). A transformed sequence is represented as:

```
token-0 [token-1]NERTAG-1 token-2
[token-3 token-4]NERTAG-2 token-5.
```

The entity text span is enclosed in square brackets, followed by its corresponding NER tag (e.g., PER, LOC, or ORG). All three datasets were updated with correct word segmentation for further processing. The prompt used for in-context word segmentation is shown below.

Prompt: "You are an expert in identifying correct word boundaries in {language name}. Your task is to perform word segmentation by inserting missing blank spaces between words and by removing extra blank spaces. The text also contains named entities enclosed in square brackets followed by an entity label.
Instructions:
- Insert missing blank spaces between two or more words.
- Remove wrong blank spaces that separate characters of a single word.
- Perform word boundary correction within square brackets as well as outside of brackets.
- Keep the named entity annotation intact along with their labels.
- Punctuations should remain as space-separated tokens.
- Within square brackets, update boundaries only when the produced word is valid.
- Spellings and number of words must remain the same. Do not add any extra symbol, character, or punctuation.
Three examples are given for your reference.
Examples:
INPUT: {*sentence with few word seg. errors*}
OUTPUT: {*sentence with correct word seg.*}"

3.2 Context-Free Entity Augmentation

The datasets are further enhanced by filling the missing annotations. For this purpose, entity lexicons are extracted for each entity type from each dataset, with minimum frequency threshold of three. These lexicons were then used to annotate unannotated sequences for each entity type without considering surrounding tokens/context. We compared the performance of models fine-tuned on the original, word segmented, and entity augmented datasets. The dataset version achieving the highest F_1 score on the validation set was selected for further processing.

3.3 Iterative Self-Training

We propose a frequency-based iterative approach to address annotation errors in low-resource NER datasets. The approach focuses on refining the missing annotations by self-supervised learning inspired by El Mekki et al. (2022), enabling iterative improvements in annotations. The iterative approach operates on a cleaner version of the datasets produced from the previous steps. At each cycle,

[1] platform.openai.com

we fine-tune the model on the current labeled subset of the training split and then run inference on the full training split to propose missing entity labels.

To ensure reliable predictions, we employ a dual-thresholding mechanism that applies a confidence threshold on logits and self-attention scores. The refined versions of datasets are evaluated against validation sets by fine-tuning a multilingual XLM-RoBERTa-large model after each iteration.

3.4 Thresholds

The approach involves a probability threshold (τ_p) and an attention-based dynamic threshold (τ_a), which together reduce the possibility of over-propagating erroneous annotations.

3.4.1 Probability Thresholding

For a given token i, let $\mathbf{z}_i \in \mathbb{R}^T$ represent the logits for T possible NER tags. The probability $p_{i,t}$ of the token i being assigned tag t is computed as:

$$p_{i,t} = \frac{\exp(z_{i,t})}{\sum_{t' \in \mathcal{T}} \exp(z_{i,t'})}, \tag{1}$$

where $\mathcal{T}$ is the set of possible NER labels in the training set. The predicted tag for token i is achieved by the highest probability as shown in Eq 2.

$$\hat{t}_i = \arg\max_{t \in \mathcal{T}} p_{i,t}. \tag{2}$$

$$p_i = \max_{t \in \mathcal{T}} p_{i,t} \geq \tau_p, \tag{3}$$

For predictions with high confidence, only tokens that have a probability higher than or equal to the threshold τ_p are considered (Eq 3). Tokens that do not fulfill this criterion are assigned the non-entity tag (O). This step is helpful in reducing false positives predicted from low-confidence probabilities.

3.4.2 Attention-Based Filtering

To find the contextually significant predictions, we incorporate attention-based filtering by obtaining self-attention scores from the model's final layer. High self-attention scores indicate labels that have inherent meaning which are useful for NER, especially for noisy datasets as named entities often rely on their inherent meaning. This filtering approach selects semantically relevant NER labels by minimizing noise and incorrect annotations. Let $\mathbf{A} \in \mathbb{R}^{L \times L}$ denote the self-attention matrix for a sequence of length L, where $A_{i,j}$ represents the attention score from token i to token j. The self-attention score for token i is represented by $A_{i,i}$, corresponding to the diagonal elements of the attention matrix. We compute a dynamic threshold τ_a based on the top 10% of self-attention scores for a sentence. The τ_a is defined as the 90th percentile of self-attention scores along the diagonal of $\mathbf{A}$.

$$\tau_a = \text{Percentile}_{90}(\{A_{j,j} \mid j = 1, \ldots, L\}). \tag{4}$$

Tokens with self-attention scores greater than or equal to τ_a are considered for prediction shown as $A_{i,i} \geq \tau_a$. This ensures that only the top 10% of tokens with the highest contextual relevance are considered for further predictions. Tokens that do not meet this threshold criterion are filtered out, retaining only predictions with significant contextual importance.

To enhance the reliability of the final predictions, we combine the probability and attention-based filtering mechanisms. The combined rule for assigning the final tag $\hat{t}_i^{\mathrm{F}}$ to token i is shown in Eq 5.

$$\hat{t}_i^{\mathrm{F}} = \begin{cases} \hat{t}_i, & \text{if } p_i \geq \tau_p \text{ and } A_{i,i} \geq \tau_a, \\ O, & \text{otherwise.} \end{cases} \tag{5}$$

The proposed thresholding balances precision and recall in the iterative self-training approach. The threshold on the NER labels, ensures the statistical confidence by discarding low-confidence predictions.

Token	Original	Pred.	Cand. 1	Cand. 2	Cand. 3
سپریم	B-ORG	B-ORG	B-ORG	B-ORG	B-ORG
کورٹ	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG
کے	O	O	O	O	O
ججز	O	O	O	O	O
میں	O	O	O	O	O
جسٹس	O	O	O	O	O
جواد	O	B-PER	B-PER	O	B-PER
ایس	O	I-PER	I-PER	O	I-PER
خواجہ	O	I-PER	I-PER	O	I-PER
اور	O	O	O	O	O
جسٹس	O	O	O	O	O
عظمت	O	B-PER	O	B-PER	B-PER
سعید	O	I-PER	O	I-PER	I-PER
شیخ	O	I-PER	O	I-PER	I-PER
شامل	O	O	O	O	O
ہیں	O	O	O	O	O

Figure 3: An example of candidate generation from predicted pseudo labels. Translation: The judges of the Supreme Court include Justice **Jawad S. Khawaja** and Justice **Azmat Saeed Sheikh**.

3.4.3 Validation of Pseudo-Labels

A multilingual XLM-RoBERTa-large NER model fine-tuned on all three datasets is employed to validate and refine the pseudo-label sequences predicted during the self-training process. For each predicted sentence, the pseudo-label sequence $\hat{Y}_p$ is used to generate multiple candidate sequences $\{C_1, C_2, \ldots, C_n\}$, along with the gold labels Y_{orig}. These candidates are generated using a permutation-based approach if more than one named entity is identified, as shown in Figure 3. For example, if two named entities are introduced by the self-training iterative model, three candidates are generated: one with the first named entity, the second with the second named entity, and the third with both named entities. Each candidate is evaluated against the predictions of the multilingual NER model, and the sequence with the highest F_1-score is selected for integration into the dataset. The model's raw predictions could be noisy; the candidate selection refines the results and filters out lower-confidence predictions. By using this validation, we effectively leverage the predictions while ensuring consistency and robustness in the final labels. The candidate with the highest F_1-score is selected as:

$$C_{\text{best}} = \arg\max_{C_i} F_1(C_i). \tag{6}$$

The validation step ensures robust evaluation of pseudo-labels using the multilingual NER model. It incorporates accurate label sequences into the enhanced datasets that iteratively improve annotation quality.

4 Datasets and Languages

Pakistani languages pose several challenges for the task of NER, such as absence of capitalization, contextual ambiguity, flexible word-order, and agglutinating nature (Khalid et al., 2023; Ehsan and Hussain, 2021; Ahmed et al., 2024). Despite the larger sample sizes in MK-PUCIT, Shahmukhi, SiNER datasets, they face limited domain coverage, incomplete NER labels, low sentence-to-entity ratio, and noisy annotations, highlighting their low-resource status.

In this paper, we enhance three existing datasets, MK-PUCIT, Shahmukhi-NER and SiNER. For evaluation purposes, we manually reviewed NER annotations for the validation and test splits of all three datasets using the annotation guidelines provided by Kanwal et al. (2019). We maintained the integrity of the validation and test sets by keeping them independent of the propagation process to ensure unbiased benchmarks for evaluation. The evaluation sets were manually curated, and therefore, not affected by any label propagation. We used 100 reference sentences from each evaluation set to compute the inter-annotator agreement (IAA) between an expert linguist and the annotator (first author). The statistics of the evaluation sets are given in Table 2. Appendix A.3 (Table 9) reports and analyzes label-count changes by comparing the original vs. revised named-entity counts (PER/LOC/ORG) for both validation and test splits of all three datasets. We used validation sets to evaluate the quality of each dataset after each processing step and final results are computed on test sets.

Eval. Sets	**MK-PUCIT**		**Shahmukhi**		**SiNER**	
NE Type	**Val**	**Test**	**Val**	**Test**	**Val**	**Test**
PER	760	1,330	412	1,210	917	985
LOC	892	2558	146	432	884	1,106
ORG	190	808	54	177	211	185
TOTAL	1,842	4,696	612	1,819	2,012	2,276
#Sents.	1,146	2,201	1,000	1,000	1,000	1,000
IAA	0.9713	0.9765	0.9604	0.9551	0.9440	0.9615

Table 2: Statistics of manually revised evaluation sets for all three datasets and inter-annotator agreement using Cohen's Kappa scores on 100 reference sentences.

Urdu: Urdu is relatively resource-rich compared to other regional languages and is in the *Vital* category (Eberhard and Fennig, 2024). Several NER datasets are available for Urdu with different data annotations and sizes (Khana et al., 2016; Hussain, 2008; Jahangir et al., 2012; Malik, 2017). However, we experimented with MK-PUCIT (Kanwal et al., 2019), which is annotated with coarse-grained named entities: person, location, and organization and is a larger dataset.

Shahmukhi: We experimented with the only available dataset for Shahmukhi, annotated with person, location, and organization labels (Ahmad et al., 2020). The quality of the dataset was previously enhanced by using the BIO annotation scheme (Tehseen et al., 2023), and we use it as the Shahmukhi benchmark in our experiments.

Sindhi: The SiNER dataset is the first large NER dataset for Sindhi (Ali et al., 2020). Although it contains additional entity types, we experimented with three coarse-grained entity categories to ensure compatibility with the other datasets (MK-PUCIT and Shahmukhi NER).

5 Experimental Setup

We conduct experiments that are designed to improve the annotation quality of NER datasets for low-resource languages, where generative LLMs often struggle due to limited language representation (Naguib et al., 2024; Villena et al., 2024; Lu et al., 2024; Chen et al., 2023). This research addresses two key questions; 1) How can supervised models be used to improve the annotation errors of noisy datasets? 2) How effective are generative LLMs to perform NER for low-resource languages? We hypothesize that supervised self-training mechanism can enhance NER annotations without requiring manual re-annotation or evaluation. Furthermore, if generative LLMs can perform NER comparably well, they can be employed as alternative annotation tools.

5.1 Supervised NER Model

For our experiments, we employed the pre-trained multilingual model, XLM-RoBERTa-large (Conneau et al., 2019), which is based on RoBERTa's architecture (Liu et al., 2019). XLM-RoBERTa-large is a transformer-based model that supports 100 languages, including Urdu, Sindhi, and Shahmukhi script. It is pre-trained on large multilingual text corpora using Masked Language Modeling (MLM) objective.

To fine-tune the model for NER, we added a token classification on top of the final transformer layer, which receives the hidden states from the last layer of the model, and computes the multi-class probability distribution over the entity classes for each token. Hyperparameters are described in A.1. Additionally, thresholding on label probabilities and self-attention scores is performed on the outputs from the final hidden layer, as described in Section 3.3.

5.2 Zero/Few-Shot NER

While the primary focus of this paper is on the enhancement of NER datasets utilizing supervised models, we also explore few-shot learning as an alternative approach. For this purpose, we transform the CoNLL format into an in-line format as described in Section 3.1. We employed the ChatGPT-4o model, a state-of-the-art multilingual generative model, using the following prompt for few-shot learning.

Prompt: "You are an expert in identifying and annotating Named Entities in {language name}. Your task is to annotate three named entities, PERSON (PER), LOCATION (LOC), and ORGANIZATION (ORG) as per the following guidelines:
PER: Names of persons without job titles and designations such as, {title examples}. Include titles that represent caste, tribe or religious titles such as, {title examples}.
LOC: Names of places, cities, towns, countries etc.
ORG: Names of organizations, public or private.
Instructions:
- Number of words must remain the same in the INPUT and OUTPUT sentences.
- Respond with only annotated output without any additional description.
Annotate sentences according to the format in the following five examples.
Examples:
INPUT: {Unannotated sentence}
OUTPUT: {Annotated sentence}"

5.3 Iterative Self-training Protocol

Our iterative model is run for 10 cycles, and macro-F_1 is computed after each iteration on the validation set. The first iteration is fine-tuned on training sentences containing named entities with a minimum frequency of three, which is gradually reduced to one for later iterations. In each cycle, we fine-tune on the current non-empty subset of the training split (sentences with at least one named entity) and then predict over the full training split to propose additional labels (no cross-fold partitioning); proposed labels are filtered by our thresholds (Section 3.4) before being merged into the next dataset version. We exclude fully empty sentences because they may contain unannotated entities that could introduce noise and degrade performance, and we select the final dataset version using the best validation macro-F_1 across cycles.

6 Results and Discussion

We report results on the manually revised validation/test splits described in Section 4 (Table 2), first analyzing LLM-based zero/few-shot NER and then the supervised correction pipeline.

6.1 Zero/Few-Shot NER Analysis

Our NER experiments begin with in-context learning, employing zero-shot and few-shot learning using ChatGPT-4o. We evaluated generated outputs for both validation and test sets to demonstrate model's capability to perform NER and annotation correction. Table 3 presents results for all three datasets.

	Zero-shot NER			**Few-shot NER**		
NE Type	**Pre.**	**Rec.**	F_1	**Pre.**	**Rec.**	F_1
			Validation Sets			
MK-PUCIT	70.06	68.74	69.40	75.14	73.23	74.19
Shahmukhi	59.09	70.70	64.38	67.32	78.56	72.51
SiNER	62.76	73.88	67.87	67.22	79.86	73.54
			Test Sets			
MK-PUCIT	73.47	71.73	72.58	72.12	80.64	76.14
Shahmukhi	63.84	67.71	65.72	65.49	76.66	70.63
SiNER	64.92	75.68	69.89	71.21	83.57	76.90

Table 3: Zero-shot and few-shot learning based micro-F_1 scores on validation and test sets for all three datasets using ChatGPT-4o.

Datasets	**Pre.**	**Rec.**	F_1
MK-PUCIT Validation			
Original	80.59	72.45	76.31
Original+WS	81.09	73.05	77.08
Original+WS+AUG	78.23	77.70	**77.96**
Shahmukhi Validation			
Original	81.74	77.19	79.40
Original+WS	84.89	77.71	**81.14**
Original+WS+AUG.	77.36	79.36	78.34
SiNER Validation			
Original	93.52	80.80	86.70
Original+WS	94.31	80.93	87.11
Original+WS+AUG	89.46	87.90	**88.67**

Table 4: Micro-F_1 scores evaluated on the validation sets for each dataset using the original, word-segmented, and augmented versions.

6.2 Automated Annotation Correction

Table 4 presents baseline results obtained by fine-tuning the XLM-RoBERTa-large model on the original versions of all three datasets. While the performance of ChatGPT-4o is quite impressive on these low-resource languages, the supervised multilingual model still outperforms it, even when fine-tuned on the original datasets containing missing annotations.

The performance of the supervised model is further improved with word segmentation correction and entity augmentation. All three datasets demonstrate performance enhancement with correct word segmentation that leads to more accurate tokenization. Unannotated named entities were filled using frequent named entities extracted from all three datasets. The MK-PUCIT and SiNER demonstrate improvements with named entity augmentation, whereas the Shahmukhi dataset shows a decrease.

			MK-PUCIT		
Iter.	**Pre.**	**Rec.**	**F-Score**	**#Sents.**	**#NEs**
0	76.50	39.96	49.66	12,412	25,479
1	75.05	49.46	56.66	14,301	30,740
2	78.18	70.96	74.28	18,629	44,621
3	78.47	70.91	74.00	18,658	44,655
4	78.17	77.21	**77.68**	18,677	44,677
5	75.63	78.31	76.84	18,701	44,712
6	78.08	75.90	76.94	18,736	44,749
7	76.19	71.40	73.34	18,745	44,759
8	78.79	67.79	72.45	18,750	44,768
9	80.80	69.81	74.64	18,763	44,792
			Shahmukhi		
Iter.	**Pre.**	**Rec.**	**F-Score**	**#Sents.**	**#NEs**
0	87.66	65.05	73.93	2,502	3,403
1	81.66	71.97	76.47	3,149	4,505
2	84.62	77.79	81.03	4,439	7,105
3	81.28	78.28	79.50	4,471	7,137
4	79.06	76.26	77.49	4,476	7,149
5	82.62	77.02	79.44	4,782	7,449
6	81.44	76.86	78.73	5,167	7,834
7	76.99	80.18	78.40	5,189	7,856
8	84.33	79.07	**81.52**	6,010	8,679
9	83.45	73.10	77.52	7,009	9,680
			SiNER		
Iter.	**Pre.**	**Rec.**	**F-Score**	**#Sents.**	**#NEs**
0	77.94	70.71	74.11	9,838	17,275
1	81.33	69.65	75.01	11,149	20,280
2	80.55	79.50	80.02	13,913	28,997
3	79.29	80.21	79.72	14,023	29,085
4	83.03	87.38	**84.83**	14,077	29,139
5	82.76	81.14	81.81	14,242	29,305
6	80.19	75.80	77.89	14,335	29,398
7	81.40	76.87	79.03	14,372	29,435
8	79.51	84.49	81.16	14,457	29,523
9	83.12	82.87	82.96	14,516	29,582

Table 5: Macro-F_1 scores for all three datasets from our iterative self-training alongside number of sentences and named entities after each iteration. Best F_1 scores are highlighted in bold.

Table 5 presents F_1 scores on the validation set along with the number of sentences and the count of named entities. The sentence count keeps increasing as new named entities are introduced after each iteration, and the training set includes only sentences that contain at least one named entity. Based on these results, we selected the best-performing version of each dataset.

	Original Dataset		**Improved Dataset**			
Datasets	**Val-F$_1$**	**Test-F$_1$**	**Val-F$_1$**	Δ	**Test-F$_1$**	Δ
MK-PUCIT	76.31	84.59	80.68	+4.37	88.55	+3.96
Shahmukhi	79.40	81.34	80.21	+0.81	82.74	+1.40
SiNER	86.70	87.92	89.14	+2.44	89.36	+1.44

Table 6: Micro-F_1 scores of the best-performing datasets on the test sets alongside the original versions of each dataset.

MK-PUCIT achieves the best macro-F_1 score after the fifth iteration, Shahmukhi after the ninth iteration, and SiNER after the fifth iteration. The final results on the test sets are presented in Table 6, using the dataset versions with highest scores from Table 5, alongside a comparison with the micro-F_1 scores from the original datasets. Table 7 presents the statistics of named entities for different types in the original and updated datasets. All entity types show increased counts in the updated datasets, suggesting that our iterative self-training approach effectively recovers missing named entities. However, counts alone do not confirm the correctness of each added label; improved F1 scores on clean test sets provide supporting evidence of better annotation quality. Additional qualitative examples of annotation corrections and their iterative updates are provided in Appendix B.

Dataset/Type	Version	PER	LOC	ORG	Total
MK-PUCIT	Original	10,486	19,868	5,277	35,631
	Updated	12,216	23,592	8,869	44,677
Shahmukhi	Original	4,609	1,852	526	6,987
	Updated	6,226	1,893	560	8,679
SiNER	Original	12,720	14,136	1,342	28,198
	Updated	12,753	13,958	2,428	29,139

Table 7: Statistics of type-wise named entities before and after entity propagation process.

Experimental results strongly support the first part of our hypothesis. Supervised self-training approach, combined with validation, achieved significant improvements in annotation and NER performance without any manual effort. These findings highlight the effectiveness of our approach which is suitable for the enhancement of low-resource sequence labeling datasets. However, the results from generative LLMs, such as ChatGPT-4o are impressive, but reveal limitations of its capabilities to perform NER for low-resource languages.

7 Conclusion

Our proposed iterative self-training approach with dual-threshold mechanism and multilingual validation, demonstrates significant improvements in annotation quality and NER performance for three datasets, MK-PUCIT, Shahmukhi, and SiNER. Our approach effectively addresses the challenge of missing annotations for low-resource languages. Experimental results highlight robustness of the approach with consistent F_1 improvements. The methodology eliminates the need of labor-intensive manual re-annotation that makes it suitable for large-scale datasets in under-resourced languages. The work not only provides a scalable solution for noisy datasets, but also highlights the importance of high-quality datasets for robust NER systems.

Limitations

The quality of automated annotation depends on the performance of the self-trained model and the validation process. While the iterative approach demonstrates progressive refinement of empty annotations, errors in the initial pseudo-labels can propagate through iterations, leading to the reinforcement of incorrect annotations. Additionally, the approach is totally automated without any intervention from human-annotators. Inherently, it lacks the contextual understanding that human annotators can provide. Despite significant improvements, enhanced datasets may still contain some annotation inconsistencies.

Acknowledgments

We acknowledge the use of ChatGPT (OpenAI) for writing assistance, grammar polishing, and text improvement.

References

Muhammad Tayyab Ahmad, Muhammad Kamran Malik, Khurram Shahzad, Faisal Aslam, Asif Iqbal, Zubair Nawaz, and Faisal Bukhari. 2020. Named Entity Recognition and Classification for Punjabi Shahmukhi. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(4):1–13.

Anil Ahmed, Degen Huang, Syed Yasser Arafat, and Imran Hameed. 2024. Enriching Urdu NER with BERT Embedding, Data Augmentation, and Hybrid Encoder-CNN Architecture. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(4):1–38.

Mashael Al-Duwais, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2024. CLEANANERCorp: Identifying and Correcting Incorrect Labels in the ANERcorp Dataset. *arXiv preprint arXiv:2408.12362*.

Wazir Ali, Jay Kumar, Saifullah Tumani, Redhwan Nour, Adeeb Noor, and Zenglin Xu. 2024. Enhancing Sindhi Word Segmentation using Subword Representation Learning and Position-aware Self-attention. *IEEE Access*, pages 1–10.

Wazir Ali, Junyu Lu, and Zenglin Xu. 2020. SiNER: A Large Dataset for Sindhi Named Entity Recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2953–2961.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named

Entity Recognition in Wikipedia. In *Proceedings of the 2009 workshop on the people's web meets NLP: Collaboratively constructed semantic resources (People's Web)*, pages 10–18.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings 8*, pages 143–153. Springer.

Gabriel Bernier-Colborne and Sowmya Vajjala. 2024. Annotation Errors and NER: A Study with OntoNotes 5.0. *arXiv preprint arXiv:2406.19172*.

Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Rui Zheng, Minlong Peng, Jie Zhou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. How Robust is GPT-3.5 to Predecessors? A Comprehensive Study on Language Understanding Tasks. *arXiv preprint arXiv:2303.00293*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *CoRR*, abs/1911.02116.

The Unicode Consortium. 2024. Arabic (unicode block). Retrieved from the Unicode Character Database.

Ryan Cotterell and Kevin Duh. 2024. Low-Resource Named Entity Recognition with Cross-Lingual, Character-level Neural Conditional Random Fields. *arXiv preprint arXiv:2404.09383*.

Gary F. Simons Eberhard, David M. and Charles D. Fennig. 2024. Ethnologue: Languages of the world. *SIL International*, 27.

Toqeer Ehsan and Sarmad Hussain. 2021. Development and Evaluation of an Urdu Treebank (CLE-UTB) and a Statistical Parser. *Language Resources and Evaluation*, 55(2):287–326.

Toqeer Ehsan and Thamar Solorio. 2025. Enhancing NER Performance in Low-Resource Pakistani Languages using Cross-Lingual Data Augmentation. In *Proceedings of the Tenth Workshop on Noisy and User-generated Text*, pages 117–132, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Abdellah El Mekki, Abdelkader El Mahdaouy, Ismail Berrada, and Ahmed Khoumsi. 2022. AdaSL: An Unsupervised Domain Adaptation Framework for Arabic Multi-dialectal Sequence Labeling. *Information Processing & Management*, 59(4):102964.

Qian Guo, Yi Guo, and Jin Zhao. 2024. Diluie: Constructing Diverse Demonstrations of In-Context Learning with Large Language Model for Unified Information Extraction. *Neural Computing and Applications*, pages 1–22.

Sarmad Hussain. 2008. Resources for Urdu language processing. In *Proceedings of the 6th workshop on Asian Language Resources*.

Faryal Jahangir, Waqas Anwar, Usama Ijaz Bajwa, and Xuan Wang. 2012. N-gram and Gazetteer List based Named Entity Recognition for Urdu: A Scarce Resourced Language. In *Proceedings of the 10th Workshop on Asian Language Resources*, pages 95–104.

Safia Kanwal, Kamran Malik, Khurram Shahzad, Faisal Aslam, and Zubair Nawaz. 2019. Urdu Named Entity Recognition: Corpus Generation and Deep Learning Applications. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):1–13.

Hamza Khalid, Ghulam Murtaza, and Qaiser Abbas. 2023. Using Data Augmentation and Bidirectional Encoder Representations from Transformers for Improving Punjabi Named Entity Recognition. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–13.

Wahab Khana, Ali Daudb, Jamal A Nasira, and Tehmina Amjada. 2016. Named Entity Dataset for Urdu Named Entity Recognition Task. *Language & Technology*, 51.

Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. *arXiv preprint arXiv:1804.10959*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Preprint*, arXiv:1808.06226.

Xiyang Liu, Chunming Hu, Richong Zhang, Junfan Chen, and Baowen Xu. 2025. Improving Data Annotation for Low-Resource Relation Extraction with Logical Rule-Augmented Collaborative Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1497–1510, Albuquerque, New Mexico. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1907.11692*.

Qiuhao Lu, Rui Li, Andrew Wen, Jinlian Wang, Liwei Wang, and Hongfang Liu. 2024. Large Language Models Struggle in Token-Level Clinical Named Entity Recognition. *arXiv preprint arXiv:2407.00731*.

Muhammad Kamran Malik. 2017. Urdu Named Entity Recognition and Classification System using Artificial Neural Network. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):1–13.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Šuppa, Hila Gonen, Joseph Marvin Imperial, Börje F Karlsson, Peiqin Lin, Nikola Ljubešić, Lester James Miranda, et al. 2023. Universal NER: A Gold-Standard Multilingual Named Entity Recognition Benchmark. *arXiv preprint arXiv:2311.09122*.

Marco Naguib, Xavier Tannier, and Aurélie Névéol. 2024. Few Shot Clinical Entity Recognition in Three Languages: Masked Language Models Outperform LLM Prompting. *arXiv preprint arXiv:2402.12801*.

Yuji Naraki, Ryosuke Yamaki, Yoshikazu Ikeda, Takafumi Horie, and Hiroki Naganuma. 2024. Augmenting NER Datasets with LLMs: Towards Automated and Refined Annotation. *arXiv preprint arXiv:2404.01334*.

Anaelia Ovalle, Ninareh Mehrabi, Palash Goyal, Jwala Dhamala, Kai-Wei Chang, Richard Zemel, Aram Galstyan, Yuval Pinter, and Rahul Gupta. 2024. Tokenization Matters: Navigating Data-Scarce Tokenization for Gender inclusive Language Technologies. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1739–1756.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured Prediction as Translation Between Augmented Natural Languages. *arXiv preprint arXiv:2101.05779*.

Susanna Rücker and Alan Akbik. 2023. CleanCoNLL: A Nearly Noise-Free Named Entity Recognition Dataset. *arXiv preprint arXiv:2310.16225*.

Andrew Rueda, Elena Álvarez Mellado, and Constantine Lignos. 2024. CoNLL#: Fine-grained Error Analysis and a Corrected Test Set for CoNLL-03 English. *arXiv preprint arXiv:2405.11865*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *arXiv preprint cs/0306050*.

Craig W. Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization Is More Than Compression. *Preprint*, arXiv:2402.18376.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large Language Models for Data Annotation and Synthesis: A Survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957.

Amina Tehseen, Toqeer Ehsan, Hannan Bin Liaqat, Xiangjie Kong, Amjad Ali, and Ala Al-Fuqaha. 2023. Shahmukhi Named Entity Recognition by using Contextualized Word Embeddings. *Expert Systems with Applications*, 229:120489.

Arthur Elwing Torres, Edleno Silva de Moura, Altigran Soares da Silva, Mario A Nascimento, and Filipe Mesquita. 2024. An Experimental Study on Data Augmentation Techniques for Named Entity Recognition on Low-Resource Domains.

Fabián Villena, Luis Miranda, and Claudio Aracena. 2024. llmNER:(Zero| Few)-Shot Named Entity Recognition, Exploiting the Power of Large Language Models. *arXiv preprint arXiv:2406.04528*.

Wei-Chen Wang and Jonas Mueller. 2022. Detecting Label Errors in Token Classification Data. *arXiv preprint arXiv:2210.03920*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. OntoNotes Release 5.0 LDC2013T19. *Linguistic Data Consortium, Philadelphia, PA*, 23(170):20.

Yuxuan Yao, Sichun Luo, Haohan Zhao, Guanzhi Deng, and Linqi Song. 2024. Can LLM Substitute Human Labeling? A Case Study of Fine-grained Chinese Address Entity Recognition Dataset for UAV Delivery. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1099–1102.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models. *arXiv preprint arXiv:2303.10420*.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. 2023. LLMaAA: Making Large Language Models as Active Annotators. *arXiv preprint arXiv:2310.19596*.

Haris Bin Zia, Agha Ali Raza, and Awais Athar. 2018. Urdu Word Segmentation using Conditional Random Fields (CRFs). *arXiv preprint arXiv:1806.05432*.

A Appendix

A.1 Hyperparameters

In the self-training process, the learning rate of 2e-5 was used along with the AdamW optimizer. The batch size was set to 8, which helped to maintain memory and training efficiency. The models were fine-tuned by using a varying number of epochs for low-resource datasets depending on the training samples. Early stopping was further implemented based on the micro F_1 score on the validation set. The maximum sequence length was set to 100 tokens. These hyperparameters ensured optimal performance of the models.

A.2 Attention Threshold Sensitivity Analysis

Table 8 reports the number of named entities extracted at various attention thresholds (percentiles 0.75–0.95). We conducted this analysis on a subset of the Shahmukhi dataset, trained for five epochs with the same settings as the first iteration (see Table 5). Because all three training sets contain many missing annotations, using a low attention threshold admits too many speculative candidates, which degrades the quality of the corrected dataset in later iterations. For this reason, we selected a 0.90 percentile threshold to retain only high-confidence named entities, and the results show that the threshold was helpful in improving the quality of the datasets.

Attention Threshold	PER	LOC	ORG	Total
Percentile = 0.95	250	103	6	359
Percentile = 0.90	293	121	10	424
Percentile = 0.85	306	127	11	444
Percentile = 0.80	308	129	11	448
Percentile = 0.75	308	132	11	451

Table 8: Number of detected named entities against different attention threshold percentiles.

A.3 Correction Statistics for Evaluation Sets

Table 9 shows statistics of corrected/filled named entities in the validation and test sets for all three datasets.

B Illustrative Annotation Corrections

Figure 4 presents annotation comparisons for some sample sentences before and after automatic label correction using the self-training label propagation method. Correctly identified labels are shown in green and incorrect labels in red. These examples were randomly selected from the corrected MK-PUCIT dataset. Although the dataset may still contain some incorrect annotations after automatic label correction and propagation, the overall quality of the dataset has improved compared to its original noisy annotations. Figure 5 further shows sample sentences with different annotations from all ten versions of the MK-PUCIT dataset. The correctly filled empty annotations are colored green.

		Validation set		Test Set	
Dataset Name	NE Type	Original	Revised	Original	Revised
MK-PUCIT	PER	657	760	1,275	1,330
	LOC	863	892	2,560	2,558
	ORG	187	190	853	808
	Total	1,707	1,842	4,688	4,696
Shahmukhi	PER	374	412	1229	1,210
	LOC	128	146	428	432
	ORG	49	54	128	177
	Total	551	612	1,785	1,819
SiNER	PER	809	917	1101	985
	LOC	908	884	1071	1,106
	ORG	249	211	140	185
	Total	1,966	2,012	2,312	2,276

Table 9: Statistics of named entities from validation and test sets after manual correction.

Sentence: ڈی جی رینجرز نے وفاقی وزیر[چوہدری نثار]👤 کو امن و امان سے متعلق تفصیلی بریفنگ دی
Translation: DG Rangers gave detailed briefing to Federal Minister **[Chaudhry Nisar]**PER regarding law and order.

Sentence: گلوکار [مہدی]👤 **حسن** کو ان کی زندگی میں 15 نگار ایوارڈز اور 25 گریجویٹ ایوارڈز بھی دیئے گئے
Translation: Singer **[Mehdi]**PER **Hassan** was also awarded 15 Nigar Awards and 25 Graduate Awards in his lifetime.

Sentence: [حیدرآباد]📍 : [نارا]📍جیل کے ڈپٹی سپرنٹنڈنٹ قاتلانہ حملے میں جاں بحق
Translation: **[Hyderabad]**LOC: Deputy Superintendent of **[Nara]**LOC Jail killed in assassination attempt.

Sentence: فخر الدین کے استعفے کے بعد چیف الیکشن کمشنر کا تقرر نہ ہونا تشویشناک ہے : [کنور دلشاد]👤
Translation: The lack of appointment of a Chief Election Commissioner after Fakhruddin's resignation is worrying: **[Kanwar Dilshad]**PER.

Sentence: شاہد خاقان کے مطابق چاروں صوبوں نے [مشترکہ مفادات کونسل]🏛 کے اجلاس میں اس پر اتفاق بھی کیا تھا
Translation: According to Shahid Khaqan, the four provinces had also agreed on this in the meeting of the **[Council of Common Interests]**ORG.

Sentence: [حیدرآباد]📍 **سینٹرل جیل** پر حملے کا خطرہ ، پولیس کی بھاری نفری تعینات
Translation: Threat of attack on **[Hyderabad]**LOC **Central Jail**, heavy police deployment.

Sentence: اسلام آباد … [مشترکہ مفادات کونسل]🏛 کا اجلاس آج ہو رہا ہے جس میں قومی توانائی پالیسی کی منظوری کا امکان ہے
Translation: Islamabad… A meeting of the **[Council of Common Interests]**ORG is being held today in which the approval of the National Energy Policy is likely.

Sentence: برطانوی وزیراعظم نے اس موقع پر [شام]📍اور دیگر ممالک کی صورتحال پر بھی بات کی
Translation: The British Prime Minister also discussed the situation in **[Syria]**LOC and other countries on this occasion.

Sentence: [چین]📍 ، تائیوان اور نیوزی لینڈ سے تعلق رکھنے والے طالب علموں نے الیکٹرک موٹر پہیوں والی جدید گاڑی ڈیزائن کر لی ہے
Translation: Students from **[China]**LOC, Taiwan and New Zealand have designed a modern vehicle with electric motor wheels.

Sentence: [عالمی ادارہ صحت]🏛 کا سیلاب سے متاثرہ علاقوں میں وبائی امراض کا خدشہ
Translation: **[World Health Organization]**ORG fears epidemics in flood-affected areas.

Sentence: [برازیلین]🏛 کمپنی نے شرارتی بچوں کو باندھ کر رکھنے والے کھلونے متعارف کروا دیئے
Translation: **[Brazilian]**ORG company introduces toys that tie up naughty children

Figure 4: Randomly selected sample sentences showing the annotations before and after label correction process. Corrected annotations are in green color and incorrect labels are in red color.

Sentence: [**حیدرآباد**]📍 : [**نارا**]📍 جیل کے ڈپٹی سپرنٹنڈنٹ قاتلانہ حملے میں جاں بحق
Translation: [**Hyderabad]**LOC: Deputy Superintendent of [**Nara**]LOC Jail killed in assassination attempt.

- Original: (no entities)
- V1: 📍[**حیدرآباد**] added
- V2 – V9: 📍[**نارا**] added - retained in later versions

Sentence: [**شاہد خاقان**]👤 کے مطابق چاروں صوبوں نے [**مشترکہ مفادات کونسل**]🏛 کے اجلاس میں اس پر اتفاق بھی کیا تھا
Translation: According to [**Shahid Khaqan**]PER, the four provinces had also agreed on this in the meeting of the [**Council of Common Interests**]ORG.

- Original: 👤[**شاہد خاقان**]
- V1: 🏛[**مشترکہ مفادات کونسل**] added
- V2 – V9: Entity set unchanged afterwards

Sentence: ترجمان [**ٹی ایم اے**]🏛 کے مطابق [**گوال منڈی**]📍میں فوج کو بلا لیا گیا
Translation: According to a [**TMA**]ORG spokesperson, the army was called to [**Gowal Mandi**]LOC.

- Original: 📍[**گوال منڈی**]
- V1: 🏛[**ٹی ایم اے**] added
- V2 – V9: No further changes

Sentence: ادھر پارٹی ذرائع کا کہنا ہے کہ [**حامد خان ایڈووکیٹ**]👤 ، پارٹی چیئرمین [**عمران خان**]👤کا دفاع کریں گے
Translation: Meanwhile, party sources say that [**Hamid Khan Advocate**]PER will defend party chairman [**Imran Khan**]PER.

- Original: 👤[**عمران خان**]
- V1: 👤[**حامد خان ایڈووکیٹ**]added
- V2 – V9: Entities stable after V1

Figure 5: Randomly selected sample sentences showing the annotations from different annotation versions. Correctly filled annotations are shown in green color.