OXFORD UNIVERSITY PRESS | JAMIA: Journal of the American Medical Informatics Associat

Semi-Supervised Learning from Small Annotated Data and Large Unlabeled Data for Fine-grained PICO Entity Recognition

Journal:	*Journal of the American Medical Informatics Association*
Manuscript ID	amiajnl-2024-016427
Article Type:	Research and Applications

JAMIA Research and Applications

Semi-Supervised Learning from Small Annotated Data and Large Unlabeled Data for Fine-grained PICO Entity Recognition

Fangyi Chen, MS[1,*], Gongbo Zhang, PhD[1,*], Yilu Fang, MA[1], Yifan Peng, PhD[2, #], Chunhua Weng, PhD[1,#]

Affiliation

[1]Department of Biomedical Informatics, Columbia University, New York, NY, USA

[2]Department of Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

[*]Equal contribution first authors

[#]Equal contribution senior corresponding authors

Corresponding Authors:

Chunhua Weng, 622 W 168th Street, PH20 room 407, chunhua@columbia.edu
Yifan Peng, 425 E 61st DIV 305, New York, NY 10065, yip4002@med.cornell.edu

Word counts: 4,252

Keywords: Fine-grained PICO extraction, PICO extraction, semi-supervised learning, named entity recognition

Abstract

Objective: Extracting PICO elements—Participants, Intervention, Comparison, and Outcomes—from clinical trial literature is essential for clinical evidence retrieval, appraisal, and synthesis. Existing approaches identify PICO entities at a coarse granularity. This study aims to develop a named entity recognition (NER) model to extract PICO entities at a fine-grained level.

Materials and Methods: Using a corpus of 2,511 abstracts with PICO mentions from 4 public datasets, we developed a semi-supervised method to facilitate the training of a NER model, FinePICO, by combining limited annotated data of PICO entities and abundant unlabeled data. For evaluation, we divided the entire dataset into two subsets: a smaller group with annotations and a larger group without annotations. We then established the theoretical lower and upper performance bounds based on the performance of supervised learning models trained solely on the small, annotated subset and on the entire set with complete annotations, respectively. Finally, we evaluated FinePICO on both the smaller annotated subset and the larger, initially unannotated subset. We measured the performance of FinePICO using precision, recall, and F1.

Results: Our method achieved precision/recall/F1 of 0.567/0.636/0.60, respectively, using a small set of annotated samples, outperforming the baseline model (F1: 0.437) by more than 16%. The model demonstrates generalizability to a different PICO framework and to another corpus, which consistently outperforms the benchmark in diverse experimental settings (p-value <0.001).

Conclusion: This study contributes a generalizable and effective semi-supervised approach leveraging large unlabeled data together with small, annotated data for fine-grained PICO extraction.

1. Introduction

Evidence-based medicine (EBM) has gained increasing popularity over the past decades and has become the guiding principle of medical practice[1–5]. The process of aggregating, synthesizing, and understanding the best available clinical evidence is essential to enhancing decision-making in medical practices and optimizing treatment outcomes[6]. The PICO (Participants, Intervention, Comparison, and Outcomes) framework serves as the basis for formulating clinical questions and effectively retrieving, selecting, and categorizing evidence from clinical studies.

Automated PICO entity extraction is a named entity recognition (NER) task, wherein each token is tagged with a pre-defined label. Early methods relied on rule-based approaches, Conditional Random Fields (CRF) models, or a combination of basic classifiers[7–9]. These approaches necessitate exhaustive feature engineering, which can be labor-intensive and time-consuming. More recently, the adoption of deep learning algorithms, such as bidirectional long short-term memory (BiLSTM) networks[10–12] and BiLSTM models augmented with a CRF module[13,14], have demonstrated superior performance without laborious feature extraction. Later, transformer-based models (e.g., BERT and its variants)[15–18] have further advanced the field.

Despite these advancements, several widely acknowledged challenges persist. One primary challenge is the lack of large, high-quality annotated datasets since annotation is a labor-intensive and time-consuming task that often requires domain experts. Furthermore, the absence of standardized PICO annotation guidelines, which becomes impractical due to variations in study purposes and domains, has further complicated the annotation process. The largest publicly available corpus, EBM-NLP[19], was reported to exhibit significant inconsistency in annotated results[20–22]. These inconsistencies are mainly attributed to unclear definitions of text span boundaries and complex annotation guidelines, resulting in suboptimal model performances[20–22]. To address these limitations, manual corrections or heuristic rule-based approaches have been leveraged to relabel entities[20,22,23]. Notably, Hu et al. proposed a two-step NLP pipeline that first classifies sections of sentences and then extracts PICO from sentences in Title and Method sections using BiomedBERT trained on re-annotated abstract[20]. Although their proposed method reduced annotation time for sentences rich in PICO information and achieved high inter-annotator agreement, the overall number of annotated abstracts remained considerably limited.

Another issue is the lack of fine-grained annotation. Most public datasets only provide coarse-level PICO annotations[24], which do not always meet the requirements for many downstream tasks, such as meta-analysis or evidence appraisal. Although the EBM-NLP dataset was unusually annotated with fine-grained PICO entities, these annotations are unsuitable for meta-analysis because they do not capture numeric values associated with outcome measures for different study arms (e.g., intervention and control). The ability to extract numerical data is critical for conducting a statistical analysis to evaluate the efficacy of the intervention[25]. Nevertheless, limited effort has been dedicated to extracting detailed outcome information, *e.g.,* the number of subjects experiencing specific outcome events. Mutinda et al. introduced a fully annotated dataset comprising 1,011 randomized controlled trials (RCTs) on breast cancer[26]. While their PICO annotation framework was suitable for conducting meta-analysis, it did not include annotations for key population characteristics (e.g., sex) because their selected RCTs focused mainly on the female population. Therefore, the generalizability of NER models built using this dataset was significantly compromised.

To improve on these areas, we proposed FinePICO, a semi-supervised learning (SSL) algorithm to enhance the extraction of fine-grained PICO entities. Our main objective was to demonstrate that combining limited labeled data and a substantial volume of unlabeled data can achieve performance comparable to that of models trained using fully annotated data. Our findings demonstrate that SSL techniques can optimize fine-grained PICO extraction by greatly expanding the training sample size while minimizing reliance on extensive manual annotation efforts.

2. Materials and Methods

2.1 Workflow Overview

FinePICO employs an iterative SSL process to adjust model weights and generate pseudo-labels for unlabeled data. **Figure 1** depicts the overview design of the model. Specifically, we first develop a NER model using the available annotated data via the traditional supervised learning approach. Once the initial model is trained, it is deployed to make inferences on the unlabeled data, referred to as pseudo-labels. We enrich the original labeled data with the high-confidence pseudo-labeled data for fine-tuning the model. We iteratively repeat the cycle of generating pseudo-labels and updating model weights until the model's performance converges on the validation dataset or a predefined maximum number of iterations has been reached. To ensure the quality of the pseudo-labels and minimize the risk of error propagation, we incorporate a self-cleaning module. It performs a quality check and selects the generated labels with high confidence.

[Figure 1]

2.1.1 Backbone Model

To leverage the power of pre-trained language models, we select a BERT-based model as our foundation model[27]. We define S as the entire collection of sentences of interest, where S_{label} refers to the sentences with pre-annotated named entity tags associated with their tokens. For each sentence $s_i^l \in S_{label}$, we have a sequence of tokens $\{t_{i1}^l,\ t_{i2}^l, \ldots, t_{im}^l\}$, where each token t_{ij}^l is associated with a label y_{ij}^l, and m is the length of the sentence s_i^l.

We also define $S_{unlabel}$ as the set of sentences without annotated named entity tags. We leverage the BERT-based model that was previously trained on S_{label} to make inferences on $S_{unlabel}$ and generate the set of pseudo-labels ($\hat{y}_{ij}^u$) for each token in the unlabeled sentence $s_i^u \in S_{unlabel}$.

The training and fine-tuning process involves applying the softmax function $\sigma(.)$ on the last layer of the neural network to compute the probability p_{ij}^k for the kth entity class associated with the token t_{ij}. The predicted entity class $\hat{y}_{ij}$ is then determined as follows:

$$p_{ij}^{k} = \sigma(\boldsymbol{z})_{ij}^{k} = \frac{\exp(z_{ij}^{k})}{\sum_{v=1}^{C} \exp\,(z_{ij}^{v})} \tag{1}$$

$$\hat{y}_{ij} = \arg max\left(\sigma(\boldsymbol{z})_{ij}\right), \quad p_{ij}^{k} \subseteq \sigma(\boldsymbol{z})_{ij} \tag{2}$$

where $\boldsymbol{z}$ is an embedding-based representation of each token, and C is the total number of entity class. $\sigma(\boldsymbol{z})_{ij}$ represents probabilities across entity tags for token t_{ij}. The target function is to minimize the cross-entropy loss function. The loss function at token t_{ij} is defined below:

$$\mathcal{L}_{CE_{ij}} = -\sum_{k=1}^{C} \mathbb{1}\left(y_{ij} = k\right)\ \log p_{ij}^{k} \tag{3}$$

The binary indicator $\mathbb{1}(*) \in \{0, 1\}$ equals to 1 if a token belongs to the k^{th} class and 0 otherwise. The overall loss function comprises of two parts: the supervised loss ($\mathcal{L}_s$) and unsupervised loss ($\mathcal{L}_u$).

$$\mathcal{L}_{total} = \mathcal{L}_s + \alpha\mathcal{L}_u \tag{4}$$

2.1.2 Supervised Learning Loss

We leveraged S_{label} as the main dataset for training and developing our initial baseline models M_0. The training process follows well-established supervised learning methods. In this stage, we aim to develop a model that can make reasonable inferences on unseen data. The baseline models were then iteratively refined using both S_{label} and $S_{unlabel}$ to minimize the learning loss. The total supervised learning loss $\mathcal{L}_s$ at t^{th} iteration is computed as follows:

$$\mathcal{L}_s = -\frac{1}{\sum_{q=1}^{n^l} m_q^l} \sum_i^{n^l} \sum_j^{m^l} \mathcal{L}_{CE_{ij}}^s \tag{5}$$

where n^l refers to the number of sentences with annotation and m^l is the number of tokens at i^{th} sentence. $\mathcal{L}_{CE_{ij}}^s$denotates as the supervised learning loss function at token t_{ij}.

2.1.3 The Self-cleaning Mechanism of Pseudo-label Generation

The baseline model M_0 infers labels for each token in the unlabeled sentences. We incorporated the sets of pseudo-labels $\{\hat{y}_{i1}^u,\ \hat{y}_{i2}^u, ...,\hat{y}_{im}^u\}$ with $\{t_{i1}^u,\ t_{i2}^u, ...,t_{im}^u\}$ of the sentence $s_i^u \in S_{unlabel}$ into the original training pool S_{label} to further improve M_0. For a token t_{ij}^u in the sentence s_i^u, its pseudo-label is formally defined as:

$$\hat{y}_{ij}^u = \arg max\left(\sigma(\mathbf{z}^u)_{ij}\right) \tag{6}$$

To maintain the quality and consistency of the generated pseudo-labels on a diverse set of training samples, we introduced a self-cleaning module to select the high-quality labels that would be used in subsequent training iterations. Specifically, we implemented three different self-cleaning approaches within the label selection process and evaluated their relative effectiveness in enhancing the overall model performances.

The selective unsupervised learning loss of a token is computed as follows:

$$\mathcal{L}^{u}{}_{CE_{ij}} = -\sum_{k=1}^{C} \mathbb{1}\left(\hat{y}_{ij}^{u}\right) \log p_{ij}^{k} \tag{7}$$

$$\mathbb{1}\left(\hat{y}_{ij}^{u}\right) = \mathbb{1}\left(\hat{y}_{ij}^{u} = k\right) \wedge \mathbb{1}\left(f\left(\hat{y}_{ij}^{u}, t_{ij}^{u}\right)\right) \tag{8}$$

where the binary indicator $\mathbb{1}\left(\hat{y}_{ij}^{u}\right) = 1$ when the two conditions are met simultaneously. The self-cleaning function f minimizes noises resulting from erroneous predictions by checking if the pseudo-label $\hat{y}_{ij}^{u}$ is accurate or has a high degree of certainty. In this study, we investigated three checking strategies.

1) **Confident-based masking**. This approach leverages prior studies that revealed the benefits of masking out low-confident examples from the training set[28,29]. It uses a predefined threshold to filter out pseudo-labels lower than this level. The threshold is empirically determined to balance between maintaining high label quality and retaining a sufficient volume of training samples.

2) **Class adaptive threshold-based masking**. A recognized limitation of confident-based masking is its potential bias toward classes with higher quality pseudo-labels[30]. To address this issue, we also implemented a class-wise threshold adjustment algorithm, where the threshold for entity class k is dynamically calculated per iteration:

$$\tau_k = \frac{\sum_{i=1}^{n^u} \max_j P(k|\, t_{ij}^u)}{\sum_{i=1}^{n^u} \sum_{j=1}^{m^u} \mathbb{1}\left(\hat{y}_{ij}^u = k\right)} \tag{9}$$

where n^u denotes the number of unlabeled sentences and m^u refers to the number of unlabeled tokens. We update the threshold for each class and filter the token and its label if the associated probability is less than the dynamic threshold τ_k.

3) **Label Selection via Model Distillation (GPT-based Selection)**. We leverage GPT-4o to evaluate the pseudo-label quality. With the tokenized sentences as input, we prompt GPT-4o to confirm whether the pseudo-labels are correct. Inspired by Hu et al.[31], we curate customized prompts for different entities. Each prompt includes annotation guideline, error-based instruction, as well as a few annotated examples (**Supplementary Table 1).** The labels confirmed as accurate by GPT-4o are then be incorporated into the new training dataset.

2.2 Data Source

We tested FinePICO with different data augmentation strategies, including the use of in-domain data, cross-domain data, and both. In-domain augmentation refers to the scenario where the labeled and unlabeled data are sampled from the same domain, while cross-domain augmentation refers to the scenario where the labeled and unlabeled are sampled from different domains.

For this purpose, we used four public datasets in this study, including PICO-Corpus[26], EBM-NLP[19] samples (n = 1,200 abstracts), and two sets of RCT abstracts[20] focused on Alzheimer's disease (AD) and COVID-19. The number of PICO entities is summarized in **Table 1**.

PICO-Corpus[26] includes 1,011 RCTs related to breast cancer, where each abstract was manually annotated for the pre-defined PICO subcategories (e.g., total sample size, age, and outcome values). EBM-NLP corpus composes RCT abstracts in diverse domains, where the training set of the abstracts was annotated by Amazon Mechanical Turk, and inter-annotator conflicts were resolved via a voting strategy. Previous studies[19,21,22] reported a lack of consistency and agreement among the annotators, with Cohen's kappa coefficient of inter-rater reliability being 0.3[20]. Due to these limitations, we adopted the annotation scheme in PICO-Corpus and utilized EBM-NLP mainly for training data augmentation. We randomly picked 1,200 abstracts from EBM-NLP. The two datasets of AD and COVID-19 did not provide fine-grained PICO annotation; as such, these two were reserved for testing purposes only.

Table 1. Characteristics for four datasets used in this study ("-" indicates unavailable).

	PICO-Corpus	EBM-NLP	AD	COVID-19
Abstracts	**1,011**	**1,200**	**150**	**150**
Training	1010			
Validation	645			
Test	944			
Population (P)		3,951	215	262
Total sample size	1,094	-	-	-
Sample size in INT	887			
Sample size in CTL	784	-	-	-
Age	231	-	-	-
Eligibility	925	-	-	-
Ethnicity	101	-	-	-
Condition	327	-	-	-
Location	186	-	-	-
Intervention (I)	1,067	5,916	467	602
Control (C)	979	563	103	180
Outcome (O)		7,151	626	626
Study outcomes	5,053	-	-	-
Outcome measures	1,081	-	-	-
Binary outcomes				
- Absolute value, INT/CTL	556/465	-	-	-
- Percentage values, INT/CTL	1,376/1,148	-	-	-
Continuous outcomes				
- Mean, INT/CTL	336/327	-	-	-
- Median, INT/CTL	270/247	-	-	-
- Standard deviation, INT/CTL	129/124	-	-	-
- q1, INT/CTL	4/4	-	-	-
- q3, INT/CTL	4/4	-	-	-

*INT: intervention arm. CTL: control group.

Following the preprocessing workflow of earlier studies[32,33], we extracted PICO entities from each sentence in the abstract. The RCT abstracts (n = 2,511) were tokenized into sentences using a Python library NLTK[34]. We divided sentences from PICO-Corpus into training, validation, and testing sets. The train-test splitting ratio was set to 80:20, and within the training set, we reserved 10% of sentences for validation. Clinical trials in EBM-NLP with PICO annotations removed were included as the unlabeled data in the training set. The two datasets, AD and COVID-19, were reserved for testing purposes. We adopted the BIO2 tagging schema[35,36] in this task, which is widely used in NER tasks. Specifically, each token in a sequence is labeled with a combination of a prefix and the type of predefined entities. The prefix indicates the beginning (B), inside (I), or outside (O) of the entities.

2.3 Foundation Model Choice & Baseline Model

We first tested several open-source models (e.g., BiomedBERT[27], BioBERT[16], SciBERT[18], ClinicalBERT[37]) used by previous studies to extract fine-grained PICO entities. These models were built using all the labeled training data and were evaluated on the test set. We followed the same hyper-parameter settings described in the prior works[20,33], using a learning rate of 5e-5, a batch size of 8, and a total of 10 training epochs.

The performances of several BERT-based models (BioBERT, SciBERT, ClinicalBERT, BiomedBERT) are detailed in **Supplementary Table 2.** BiomedBERT achieved the highest macro-average precision of 0.662, recall of 0.716, and F1 score of 0.688 in extracting fine-grained PICO elements, outperforming the other models. Such results aligned with the findings of a previous study[32] focusing on extracting granular PICO information from texts, suggesting the superior performance of BiomedBERT in identifying PICO entities. Therefore, in the remaining experiments, we used BiomedBERT as the baseline model.

Considering the constraints of limited available annotations, we defined an ideal scenario where the unlabeled data would be annotated by human experts. We used the model performance from this ideal scenario as the upper bound of SSL model performance in our experiments.

2.4 Data Augmentation with Unlabeled Data

We augmented the training data with unlabeled text corpus from three distinct domains: in-domain (similar domain with the labeled data), cross-domain (different domains from the labeled data: EBM-NLP), and all-domain (both in-domain and cross-domain unlabeled data). To evaluate the in-domain and all-domain cases, we masked out annotations with different ratios in the training data. Specifically, we randomly selected 10%, 30%, 50%, 70%, 90%, and 100% of the sentences from the training set to act as labeled data and treat the rest as unlabeled data (**Supplementary Table 3**). The proposed algorithm was assessed across these different masking ratios and compared with the performances of the baseline model.

2.5 Generalizability Test on an Enhanced PICO Scheme

To demonstrate generalizability, we evaluated FinePICO on a newly annotated dataset under a revised guideline adopted from the one used for PICO-Corpus. The first change is a new demographic entity representing the genders of participants. Gender is an important demographic characteristic[38,39] that enables the exploration of varying treatment effects across different gender subgroups; however, it was not included in the original annotation scheme.

To streamline the gender entity labeling process, we constructed a gender entity tagger using the BiomedBERT fine-tuned on carefully selected samples from EBM-NLP. The samples were selected by first extracting sentences containing tokens tagged with the “sex” entity label, followed by manual validation, and supplemented by a keyword search approach to ensure accurate extraction of the “sex” entity from the text. The final data comprised 569 sentences, partitioned with 80% for training, 10% for validation, and 10% for testing.

We trained the model for 5 epochs with a learning rate of 5e-5, achieving a high F1 score of 0.989. The best-performing model was then utilized to label the "sex" tokens in the PICO-Corpus (training and validation set). Finally, two researchers (FC, YF) manually annotated "sex" tokens in the testing set to provide a benchmark.

The second change involves replacing and consolidating several categories to enhance clarity and efficiency. The revised PICO scheme is illustrated in **Figure 2**, and the details of the entity counts can be found in **Supplementary Table 4.** Specifically, we combined the "subject eligibility" and "conditions" into a single entity group now named "recruited participant eligibility conditions." This merger reflects their interrelated nature and simplifies the tagging process. Additionally, we combined "outcome names" and "outcome measures" into one group to avoid redundancy and streamline the dataset.

[Figure 2]

2.6 Evaluation Metrics

We tested our models on two independent test sets (PICO-Corpus, AD, and COVID-19 from Hu et al.[20]). In the first test set derived from the PICO-Corpus, we evaluated our NER models at a strict entity level that requires the recognition of the complete span of each entity. Since token-level evaluation can be misleadingly high for the intended task, as missing tokens could result in significant misinterpretation, it is essential to accurately capture entire PICO entities. We computed the macro-average precision, recall, and F1 score using seqeval[40], a well-tested tool often deployed in numerous NLP studies for system evaluation[41]. The 95% confidence interval of the performance was estimated based on the bootstrapped test samples.

Acknowledging the variance in annotated spans across different datasets, we conducted a second evaluation using partial-matching[42] on AD and COVID-19 datasets. Here, we counted a predicted named entity as a true positive if it overlaps with the human-labeled entities with at least one token. It is worth noting that AD and COVID-19 did not include fine-grained PICO annotation. Therefore, we first converted the predicted fine-grained entities into coarse-level entities and evaluated them using a partial matching strategy[42].

3. Results

3.1 Performance on Limited Labeled Samples

The baseline models were established solely using labeled samples. The lower bound performance refers to the baseline model evaluated on the test set, whereas the upper bound corresponds to the model trained on the entire set of labeled training samples and evaluated on the test set.

In scenarios where limited labeled samples were available (e.g., case 1 with 10% labeled data, as shown in **Supplementary Table 3**), FinePICO notably surpassed the lower bound benchmarks in both data augmentation settings during the iterative training process (**Figure 3**). For instance, employing the confident-based approach, the model augmented with cross-domain data achieved the highest macro-average F1 score of 0.589 at the 7th iteration. This score marked an approximately 15% improvement over the lower bound (F1-score of 0.44). Similarly, statistical improvements over the baseline model were observed when different data augmentation strategies were applied, and when the model was adapted to the revised PICO scheme.

[Figure 3]

3.2 Performance Comparison of Different Self-cleaning Approaches

The performances of three self-cleaning strategies for optimizing pseudo-label selection are summarized in **Table 2**. All three self-cleaning methods outperformed the baseline models by over 10% in precisions, recall, and F1 scores, with their respective 95% confidence intervals (CIs) provided in **Supplementary Table 5**. In the original PICO scheme, GPT-based selection achieved the highest performance (average F1 of 0.6, 95% CI between 0.609 and 0.664) among the three methods. However, we did not perceive any statistical enhancement (p-value =0.171) using GPT-based selection over the confident-based masking algorithm. In the revised PICO scheme, the adaptive threshold-based method was the most effective in selecting high-quality pseudo-labels among the three self-cleaning approaches, obtaining the highest average F1 score of 0.653 (95% CI: 0.657 - 0.706) when augmented with in-domain unlabeled data. Additionally, both confident-based and adaptive threshold-based masking methods have performed statistically better than GPT-based selection (p-value < 0.05).

Table 2: Average performance of different self-cleaning approaches evaluated on the bootstrapped testing samples. R – Recall. P – Precision.

Self-cleaning Approaches	**Original Scheme**			**Revised Scheme**		
	R	P	F1	R	P	F1
Confident-based masking						
In-domain	0.607	0.566	0.586	0.675	0.628	0.651
Cross-domain	0.619	0.580	0.598	0.652	0.613	0.632
Class adaptive threshold masking						
In-domain	0.636	0.561	0.596	0.682	0.626	0.653
Cross-domain	0.617	0.571	0.594	0.677	0.627	0.651
GPT-based selection						

In-domain	0.607	0.591	0.599	0.639	0.607	0.622
Cross-domain	0.636	0.567	0.600	0.613	0.608	0.610
Baseline Model (BiomedBERT)	0.489	0.394	0.437	0.568	0.480	0.520

3.3 Generalizability Assessment

To assess the generalizability of FinePICO, with the consideration of available resources, we selected confident-based masking as the primary self-cleaning approach. The best-performing models were examined on additional data augmentation cases ranging from 30% to 100% of annotated samples.

3.3.1 Additional Data Augmentation Scenarios

Table 3 presents the average performances of models with different data augmentation cases, with the baseline levels detailed in **Supplementary Table 6**. Our analysis revealed a positive linear relationship between model performance and the number of annotated samples used for training. Specifically, performance increased from an F1 score of 0.667 (cross-domain) with 30% of the annotated data to 0.695 with the entire labeled data. This suggests that while additional labeled data continues to improve the model performance, the marginal gains diminish as the proportion of annotations approaches 100%.

As we increased the number of annotated samples while keeping the size of unlabeled training samples constant, we consistently observed statistically significant improvements (p-value <0.001) in the model's performance compared to the benchmark. These improvements were particularly notable in the extreme case when the maximum amount of labeled data was used (**Figure 4**). Furthermore, the performance of the proposed algorithm consistently surpassed the baseline levels across the revised PICO scheme, showcasing the model's robustness and adaptability.

[Figure 4]

Additionally, we examined the performance differences among semi-supervised learning under various data augmentation approaches (in-domain, cross-domain, all-domain). In the original PICO scheme, models trained on both cross-domain and all-domain data performed statistically better than models trained using in-domain data (p-value < 0.001), whereas, in the revised scheme, we observed the opposite trend.

Table 3. Average Performances on bootstrapped testing samples. R – Recall. P – Precision

Data Augmentation Cases	**Original Scheme**			**Revised Scheme**		
	R	P	F1	R	P	F1
Case 2: 30% Annotation						
In-domain	0.673	0.623	0.647	0.722	0.668	0.694
Cross-domain	0.674	0.616	0.644	0.712	0.650	0.680
All	0.689	0.645	0.667	0.708	0.675	0.691
Case 3: 50% Annotation						
In-domain	0.687	0.647	0.667	0.737	0.702	0.719
Cross-domain	0.699	0.647	0.672	0.717	0.691	0.703
All	0.699	0.650	0.673	0.730	0.700	0.714
Case 4: 70% Annotation						
In-domain	0.699	0.663	0.681	0.734	0.699	0.716
Cross-domain	0.702	0.649	0.674	0.737	0.697	0.716
All	0.699	0.645	0.646	0.735	0.700	0.717
Case 5: 90% Annotation						
In-domain	0.715	0.663	0.688	0.749	0.703	0.725
Cross-domain	0.728	0.672	0.699	0.750	0.706	0.727
All	0.717	0.678	0.697	0.742	0.693	0.717
Case 6: 100% Annotation						
In-domain	-	-	- -	-	-	-
Cross-domain	0.716	0.676	0.695	0.753	0.713	0.732
All	0.716	0.676	0.695	0.753	0.713	0.732

3.3.2 Evaluation on the Independent Testing Sets

We further applied the best-performing model to another independent testing corpus (AD, COVID-19)[20], and the averaged performances over 30 bootstrapped samples, along with the baseline levels, were recorded in **Table 4**. The proposed model demonstrated statistically significant improvement (p-value = 0.014 in the original scheme and p-value = 0.025 in the revised scheme) over the baselines evaluated under AD and COVID-19 corpus.

Table 4. Partial-matching performances of the optimal model evaluated on the external testing corpus (i.e., AD and COVID-19 datasets). The results were the average performances, and the 95% confidence interval obtained from bootstrapped samples with 30 iterations.

Models	**Original Scheme**			**Revised Scheme**		
	Recall	Precision	F1	Recall	Precision	F1
Baseline	0.922 (0.902, 0.943)	0.780 (0.756, 0.804)	0.845 (0.825, 0.870)	0.931 (0.913, 0.95)	0.778 (0.745, 0.811)	0.848 (0.825, 0.870)
FinePICO	0.919 (0.896, 0.943)	**0.795** **(0.762, 0.830)**	**0.853** **(0.826, 0,879)**	0.928 (0.91, 0.946)	**0.789** **(0.767, 0.811)**	**0.853** **(0.826, 0.879)**

4. Discussion

In this study, we developed a semi-supervised learning approach to overcome several key challenges in fine-grained PICO entity recognition, including the limited amount of high-quality annotated data and the lack of standardized fine-grained PICO annotation guidelines. These limitations have historically hindered the adaptability and generalizability of existing PICO extraction models.

FinePICO demonstrated substantial improvements (p-value < 0.001) compared to the baseline models across various experimental settings, including in-domain, cross-domain, and all-domain datasets. This was especially evident in scenarios where a large percentage of trained samples were unannotated. For instance, in the case where only 10% of the training sample was labeled, FinePICO demonstrated an overall improvement of over 16% in F1 score compared to the conventional supervised learning-based approach (in the original PICO scheme, our best model using a GPT-based label selector achieved an average F1 of 0.60 versus 0.437 for the baseline model, p-value < 0.001). FinePICO also consistently outperformed the benchmarks when applied to the revised PICO scheme, demonstrating its robustness and adaptability to varied annotation guidelines. This flexibility allows users to use their preferred fine-grained PICO scheme. As shown in the experiments (**Figure 4**), the proposed algorithm effectively enhanced the model performance by augmenting training samples without needing an additional manual labeling process, significantly surpassing the models trained exclusively on fully annotated datasets.

Prior research[43–45] suggested that leveraging abundant unlabeled data with a small portion of labeled data can greatly improve learning performance. Conversely, in certain situations, semi-supervised learning offers no benefits and may even lead to performance degradation. Such situations include distribution mismatches between labeled and unlabeled data or when the labeled or unlabeled datasets are too small to extract any meaningful patterns and information[46,47].

The outcomes from the study revealed the feasibility of using a semi-supervised learning-based approach to optimize fine-grained PICO entity recognition. In our experiments, we also compared the performances of the models using unlabeled datasets from three different sources: in-domain (similar domain as the labeled data), cross-domain (different domain from the labeled data), and a combination of both. In the original PICO scheme, the models trained with cross-domain data consistently exhibited better (p-value <0.01) performances than those trained with in-domain data. This improvement may be due to the increased data diversity and the introduction of new useful context information. These findings suggested the potential of using published cross-domain RCTs to enhance PICO extraction, especially when in-domain RCT studies were scarce.

Despite the promising results of our model, several major types of errors were recognized. The first was the boundary detection error, where the model failed to capture the complete entity span, especially in the names of the intervention arm and outcome measured. For instance, "feasibility of achieving 12 meth/week (metabolic equivalent of task hours per week)" was annotated as the outcome measured in the sentence "A key secondary endpoint was the feasibility of achieving 12 meth/week (metabolic equivalent of task hours per week)". However, our model failed to identify the content within the parathesis as the outcome name. Boundary detection error is common in other PICO NER models as well[20,23], and part of these errors is potentially attributed to a lack of consistency in human annotation. We believe that a clear annotation guideline that explicitly defines what to include and exclude in the labeling process can minimize these errors.

Second, for certain cases, we perceived that our model has difficulty differentiating between values in the intervention arm and control group. For example, the sentence "Patients were randomized to receive zoledronic acid administered intravenously every 4 weeks (n = 911) vs every 12 weeks (n = 911) for 2 years" from the RCT aims to compare the effect of a longer dosing interval (12 weeks) versus the standard dosing interval (every 4 weeks). The model misannotated the first "911" as the intervention sample size, and the second "911" as the control sample size; however, such confusion was understandable. It is also challenging for humans to make this decision without considering broader contextual information. To improve future outcomes, performing PICO recognition on a wider contextual level, rather than limiting it to the sentence level, may mitigate this confusion.

Lastly, we noticed that our model often confused with background information as one of the PICO population entities (e.g., sex, race). Such as in the sentence "breast cancer, with an incidence of 32%, is the most frequent cancer among Egyptian women" which depicts the general information of breast cancer in a subpopulation, our model identified the "Egyptian women" as the recruited population demographic characteristics. Even though the main recruited participants were under this category, it is inaccurate to assume the study recruited participants to match the population mentioned in the background section. Thus, it is beneficial to leverage section information in determining final participants and reported results. Recently, Hu et al.[48] developed a few-shot prompt learning-based approach to classifying sentences in RCTs into different subsections (Introduction, Background, Methods, Results). This demonstrates state-of-the-art performance with minimal training samples required. In the future, we can potentially incorporate the sentence classifiers before applying fine-grained PICO extractors.

5. Conclusion

In this paper, we proposed a semi-supervised learning approach to address two notable challenges in fine-grained PICO extraction: the scarcity of high-quality annotation samples and the absence of standardized annotation guidelines. To our knowledge, this is the first attempt to comprehensively examine the performance of semi-supervised learning in fine-grained PICO extraction across various experimental settings. The findings suggested that leveraging the SSL model can effectively enhance the performance of traditional supervised learning-based models by augmenting training datasets without relying on extensive human annotation. The approach exhibited superior results compared to the benchmark, with high robustness and generalizability to other user-defined annotation schemes. This encourages the adoption of SSL techniques in extracting fine-grained PICO entities from RCTs and inspires more innovative SSL algorithms in this field.

6. Author Contributions

FC: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing-original draft, writing-review & editing. GZ: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing-original draft, writing-review & editing. YF: Data curation, investigation, validation, writing-review & editing. YP: Conceptualization, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, writing-review & edition. CW: Conceptualization, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, writing-review & edition. All authors have read and approved the manuscript.

7. Competing Interests

All authors declare no competing interests.

8. Fundings

This project was sponsored by the National Library of Medicine grant R01LM014344, R01LM014573, R01LM009886, T15LM007079, the National Human Genome Research Institute grant R01HG012655, and the National Center for Advancing Translational Sciences

UL1TR001873 and UL1TR002384. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

9. Data availability

The data and codes underlying the study will be available upon request.

References

1. Collins J. Evidence-based medicine. Journal of the American College of Radiology. 2007;4(8):551–4.

2. You S. Perspective and future of evidence-based medicine. Stroke and vascular neurology. 2016;1(4).

3. Akobeng AK. Principles of evidence based medicine. Archives of disease in childhood. 2005;90(8):837–40.

4. Peng Y, Rousseau JF, Shortliffe EH, Weng C. AI-generated text may have a role in evidence-based medicine. Nature medicine. 2023;29(7):1593–4.

5. Zhang G, Jin Q, McInerney DJ, Chen Y, Wang F, Cole CL, et al. Leveraging generative AI for clinical evidence synthesis needs to ensure trustworthiness. Journal of Biomedical Informatics. 2024;153:104640.

6. Berlin JA, Golub RM. Meta-analysis as evidence: building a better pyramid. Jama. 2014;312(6):603–6.

7. Dawes M, Pluye P, Shea L, Grad R, Greenberg A, Nie JY. The identification of clinically important elements within medical journal abstracts: Patient--Population--Problem, Exposure--Intervention, Comparison, Outcome, Duration and Results (PECODR). Informatics in Primary care. 2007;15(1).

8. Demner-Fushman D, Lin J. Answering clinical questions with knowledge-based and statistical techniques. Computational Linguistics. 2007;33(1):63–103.

9. Chabou S, Iglewski M. Combination of conditional random field with a rule based method in the extraction of PICO elements. BMC medical informatics and decision making. 2018;18:1–14.

10. Jin D, Szolovits P. Pico element detection in medical text via long short-term memory neural networks. In 2018. p. 67–75.

11. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks. 2005;18(5–6):602–10.

12. Ma X, Hovy E. End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:160301354. 2016;

13. Jin D, Szolovits P. Advancing PICO element detection in biomedical text via deep neural networks. Bioinformatics. 2020;36(12):3856–62.

14. Brockmeier AJ, Ju M, Przybyła P, Ananiadou S. Improving reference prioritisation with PICO recognition. BMC medical informatics and decision making. 2019;19:1–14.

15. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018;

16. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020;36(4):1234–40.

17. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019;

18. Beltagy I, Lo K, Cohan A. SciBERT: A pretrained language model for scientific text. arXiv preprint arXiv:190310676. 2019;

19. Nye B, Li JJ, Patel R, Yang Y, Marshall IJ, Nenkova A, et al. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In NIH Public Access; 2018. p. 197.

20. Hu Y, Keloth VK, Raja K, Chen Y, Xu H. Towards precise PICO extraction from abstracts of randomized controlled trials using a section-specific learning approach. Bioinformatics. 2023;39(9):btad542.

21. Lee GE, Sun A. A study on agreement in PICO span annotations. In 2019. p. 1149–52.

22. Abaho M, Bollegala D, Williamson P, Dodd S. Correcting crowdsourced annotations to improve detection of outcome types in evidence based medicine. In 2019. p. 1–5.

23. Dhrangadhariya A, Müller H. Not so weak PICO: leveraging weak supervision for participants, interventions, and outcomes recognition for systematic review automation. JAMIA open. 2023;6(1):ooac107.

24. Sanchez-Graillet O, Witte C, Grimm F, Cimiano P. An annotated corpus of clinical trial publications supporting schema-based relational information extraction. Journal of Biomedical Semantics. 2022;13(1):1–18.

25. Ahn E, Kang H. Introduction to systematic review and meta-analysis. Korean journal of anesthesiology. 2018;71(2):103.

26. Mutinda F, Liew K, Yada S, Wakamiya S, Aramaki E. PICO corpus: A publicly available corpus to support automatic data extraction from biomedical literature. In 2022. p. 26–31.

27. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, et al. Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH). 2021;3(1):1–23.

28. Ferreira RE, Lee YJ, Dórea JR. Using pseudo-labeling to improve performance of deep neural networks for animal identification. Scientific Reports. 2023;13(1):13875.

29. Xie Q, Dai Z, Hovy E, Luong T, Le Q. Unsupervised data augmentation for consistency training. Advances in neural information processing systems. 2020;33:6256–68.

30. Zhang W, Lin H, Han X, Sun L. De-biasing distantly supervised named entity recognition via causal intervention. arXiv preprint arXiv:210609233. 2021;

31. Hu Y, Chen Q, Du J, Peng X, Keloth VK, Zuo X, et al. Improving large language models for clinical named entity recognition via prompt engineering. Journal of the American Medical Informatics Association. 2024;ocad259.

32. Zhang G, Zhou Y, Hu Y, Xu H, Weng C, Peng Y. A span-based model for extracting overlapping PICO entities from randomized controlled trial publications. Journal of the American Medical Informatics Association. 2024;31(5):1163–71.

33. Mutinda FW, Liew K, Yada S, Wakamiya S, Aramaki E. Automatic data extraction to support meta-analysis statistical analysis: a case study on breast cancer. BMC Medical Informatics and Decision Making. 2022;22(1):158.

34. Bird S, Klein E, Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.; 2009.

35. Sang EF, Buchholz S. Introduction to the CoNLL-2000 shared task: Chunking. arXiv preprint cs/0009008. 2000;

36. He S, Wang T, Lu Y, Lin H, Han X, Sun Y, et al. Document Information Extraction via Global Tagging. In Springer; 2023. p. 145–58.

37. Wang G, Liu X, Ying Z, Yang G, Chen Z, Liu Z, et al. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. Nature Medicine. 2023;29(10):2633–42.

38. Assmann SF, Pocock SJ, Enos LE, Kasten LE. Subgroup analysis and other (mis) uses of baseline data in clinical trials. The Lancet. 2000;355(9209):1064–9.

39. Bhandari M, Devereaux P, Li P, Mah D, Lim K, Schünemann HJ, et al. Misuse of baseline comparison tests and subgroup analyses in surgical trials. Clinical Orthopaedics and Related Research®. 2006;447:247–51.

40. Nakayama H. A Python Framework for Sequence Labeling Evaluation (Named-Entity Recognition, Pos Tagging, etc.) [Internet]. 2018. Available from: https://github.com/chakki-works/seqeval

41. Heddes J, Meerdink P, Pieters M, Marx M. The Automatic Detection of Dataset Names in Scientific Articles. Data. 2021 Aug;6(8):84.

42. Seki K, Mostafa J. A probabilistic model for identifying protein names and their name boundaries. In IEEE; 2003. p. 251–8.

43. Hong S, Noh H, Han B. Decoupled deep neural network for semi-supervised semantic segmentation. Advances in neural information processing systems. 2015;28.

44. Banitalebi-Dehkordi A. Knowledge distillation for low-power object detection: A simple technique and its extensions for training compact models using unlabeled data. In 2021. p. 769–78.

45. Chen Y, Tan X, Zhao B, Chen Z, Song R, Liang J, et al. Boosting semi-supervised learning by exploiting all unlabeled data. In 2023. p. 7548–57.

46. Oliver A, Odena A, Raffel CA, Cubuk ED, Goodfellow I. Realistic evaluation of deep semi-supervised learning algorithms. Advances in neural information processing systems. 2018;31.

47. Singh A, Nowak R, Zhu J. Unlabeled data: Now it helps, now it doesn't. Advances in neural information processing systems. 2008;21.

48. Hu Y, Chen Y, Xu H. Towards More Generalizable and Accurate Sentence Classification in Medical Abstracts with Less Data. J Healthc Inform Res. 2023 Dec;7(4):542–56.

Figure legends

Figure 1. The overview of the study workflow.

Figure 2. The enhanced PICO scheme.

Figure 3. Performance of the proposed models using 10% annotated data augmented with in-domain, cross-domain data. Lower bound performance is detonated as the baseline model evaluated on the test set. The upper bound refers to the baseline model trained using the whole labeled training samples and evaluated on the test set.

Figure 4. (a) Statistical performance comparison to baseline models in 6 simulated cases and (b) experimental setting (in-domain, cross-domain, and all) comparison. *p<0.05, **p<0.01, ***p<0.001

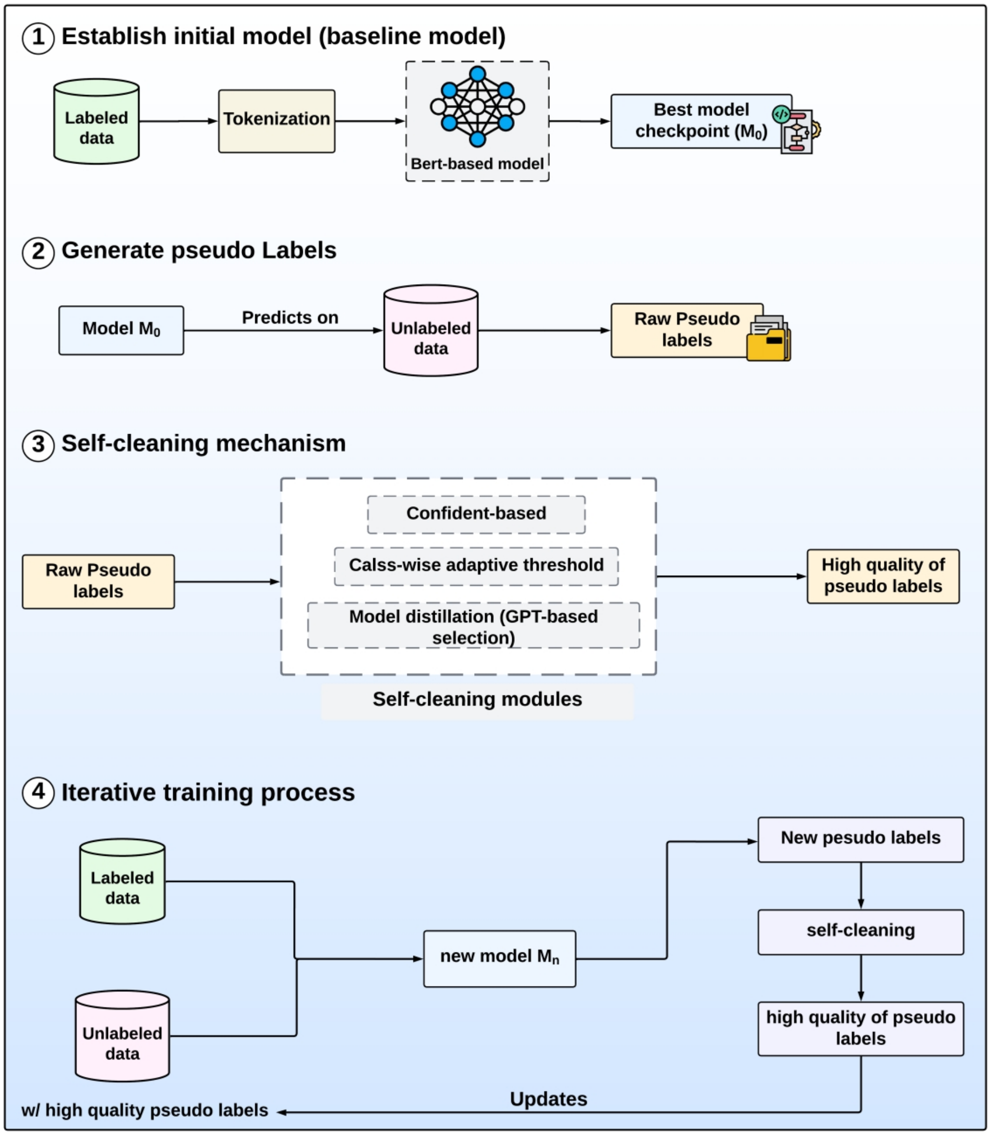

1 Establish initial model (baseline model)
Labeled data
Tokenization
Bert-based model
Best model checkpoint (M_0)
2 Generate pseudo Labels
Model M_0
Predicts on
Unlabeled data
Raw Pseudo labels
3 Self-cleaning mechanism
Raw Pseudo labels
Confident-based
Calss-wise adaptive threshold
Model distillation (GPT-based selection)
Self-cleaning modules
High quality of pseudo labels
4 Iterative training process
Labeled data
Unlabeled data
new model M_n
New pesudo labels
self-cleaning
high quality of pseudo labels
Updates
w/ high quality pseudo labels

Figure 1. The overview of the study workflow.

905x1035mm (118 x 118 DPI)

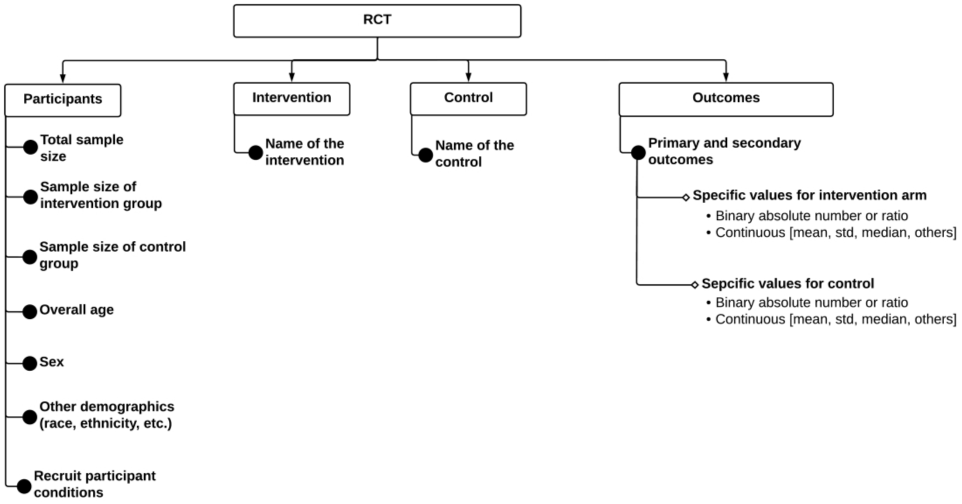

RCT
Participants
Intervention
Control
Outcomes
Total sample size
Sample size of intervention group
Sample size of control group
Overall age
Sex
Other demographics (race, ethnicity, etc.)
Recruit participant conditions
Name of the intervention
Name of the control
Primary and secondary outcomes
Specific values for intervention arm
• Binary absolute number or ratio
• Continuous [mean, std, median, others]
Sepcific values for control
• Binary absolute number or ratio
• Continuous [mean, std, median, others]

Figure 2. The enhanced PICO scheme.

906x478mm (118 x 118 DPI)

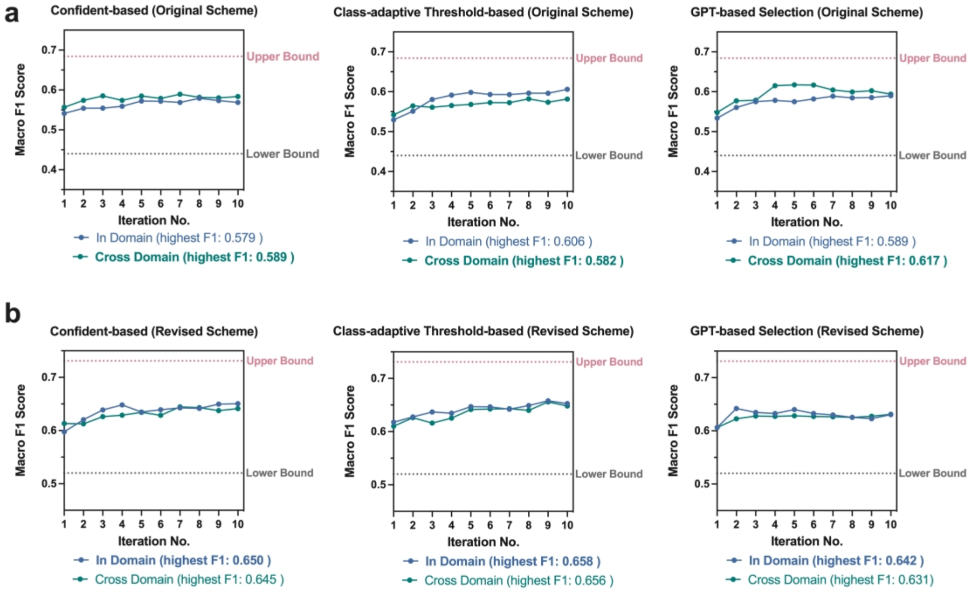


Figure 3. Performance of the proposed models using 10% annotated data augmented with in-domain, cross-domain data. Lower bound performance is detonated as the baseline model evaluated on the test set. The upper bound refers to the baseline model trained using the whole labeled training samples and evaluated on the test set.

278x169mm (300 x 300 DPI)

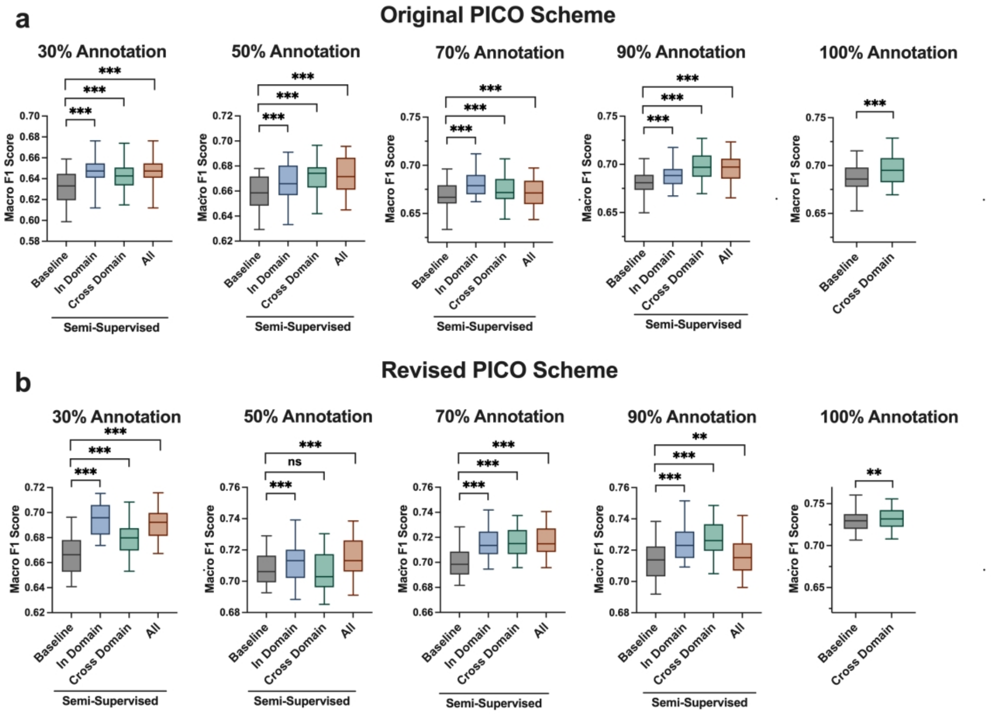


Figure 4. (a) Statistical performance comparison to baseline models in 6 simulated cases and (b) experimental setting (in-domain, cross-domain, and all) comparison. *p<0.05, **p<0.01, ***p<0.001

280x202mm (300 x 300 DPI)

Supplementary Materials for “Fine-grained PICO entity recognition using semi-supervised learning approach

Fangyi Chen, MS[1,*], Gongbo Zhang, PhD[1,*], Yilu Fang, MA[1], Yifan Peng, PhD[2, #], Chunhua Weng, PhD[1,#]

Affiliation

[1]Department of Biomedical Informatics, Columbia University, New York, NY, USA

[2]Department of Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

[*]Equal contribution first authors

[#]Equal contribution senior corresponding authors

Table of Contents

Supplementary Table 1. Customized Prompts for each entity.

Entity	**Prompt**
Condition	Based on the entity definition below, check if the detected tokens '{*tokens*}' describe part of the condition in this sentence: "{*sentence*}". Return yes or no only. Definition: condition refers to medical conditions that patients often experience, which can be the symptoms that an RCT attempts to prevent or alleviate Sample output: Yes
Eligibility	Based on the entity definition below, check if the detected tokens '{*tokens*}' describe part of the eligibility in this sentence: "{*sentence*}". Return yes or no only. Definition: eligibility specifies the particular health conditions or stages of a disease, or medical history that participants must have, or medication treatment participants receive. Sex or age is not included in this category. Sample output: Yes
Total sample size	Check if the detected tokens '{*tokens*}' describe the total sample size of the recruited participants in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Age	Check if the detected tokens '{*tokens*}' describe the age in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Location	Check if the detected tokens '{*tokens*}' describe part of the location in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Ethnicity	Check if the detected tokens '{*tokens*}' describe the ethnicity in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Intervention name	Check if the detected tokens '{*tokens*}' describe part of the intervention under the PICO framework in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes

Intervention arm sample size	Check if the detected tokens '{*tokens*}' describe the sample size of the intervention arm in this sentence: "{*sentence*}". Return yes or no only. Sample size in control arm (e.g., placebo group) should not be included. Sample output: Yes
Control name	Check if the detected tokens '{*tokens*}' describe part of the control under the PICO framework in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Control arm sample size	Check if the detected tokens '{*tokens*}' describe the sample size of the control arm in this sentence: "{*sentence*}". Return yes or no only. Sample size in intervention arm should not be included. Sample output: Yes
Outcome	Check if the detected tokens '{*tokens*}' describe part of the outcome in this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Outcome measure	Based on the entity definition below, Check if the detected tokens '{*tokens*}' describe part of the outcome measure in this sentence: "{*sentence*}". Return yes or no only. Definition: outcome measure refers to the metrics used to quantify the outcomes of an RCT study Sample output: Yes
Discrete outcome values in intervention arm	Based on the example below, Check if the detected tokens '{*tokens*}' describe the results in the intervention arm this sentence: "{*sentence*}". Return yes or no only. Example 1: Input: - Check tokens: 79 - Sentence: 79 deaths were observed in the HDCT arm and 77 deaths were observed in the placebo arm. Output: Yes Example 2: Input: - Check tokens: 77

	- Sentence: 79 deaths were observed in the HDCT arm and 77 deaths were observed in the control arm. Output: No
Discrete outcome values in control arm	Based on the example below, Check if the detected tokens '{*tokens*}' describe the results in the control arm this sentence: "{*sentence*}". Return yes or no only. Example 1: Input: - Check tokens: 77 - Sentence: 79 deaths were observed in the HDCT arm and 77 deaths were observed in the ST arm Output: Yes Example 2: Input: - Check tokens: 79 - Sentence: 79 deaths were observed in the HDCT arm and 77 deaths were observed in the ST arm Output: No
Continuous numeric outcome values in control arm	Check if the detected tokens '{*tokens*}' describe the continuous numeric values in the control arm this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Continuous numeric outcome values in intervention arm	Check if the detected tokens '{*tokens*}' describe the continuous numeric values in the intervention arm this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Standard deviation in intervention arm	Check if the detected tokens '{*tokens*}' describe the standard deviation values in the intervention arm this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes
Standard deviation in control arm	Check if the detected tokens '{*tokens*}' describe the standard deviation values in the control arm this sentence: "{*sentence*}". Return yes or no only. Sample output: Yes

Supplementary Table 2. Macro-average performance of models (BioBERT, SciBERT, ClinicalBERT, BiomedBERT) on the testing set.

Models	**Raw Scheme**			**Modified Scheme**		
	Recall	Precision	F1	Recall	Precision	F1
Baseline	0.922	0.780	0.845	0.931	0.778	0.848
Proposed Model	0.919	**0.795**	**0.853**	0.928	**0.789**	**0.853**

Supplementary Table 3. Number of sentences in train, validation, and test sets in different simulated cases.

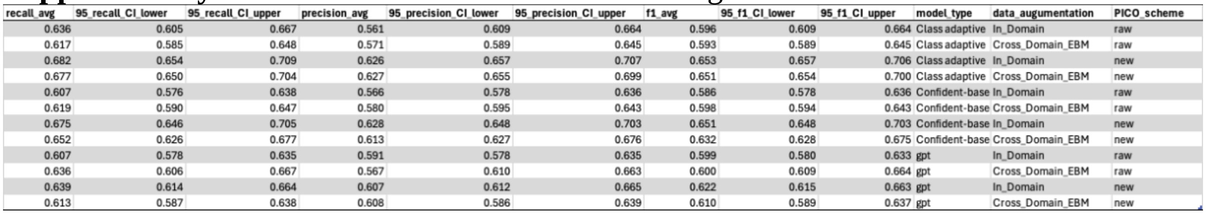
Category	**Simulated Cases**					
	Case 1: 10%	Case 2: 30%	Case 3: 50%	Case 4: 70%	Case 5: 90%	Case 6: 100%
Train						
(In Domain setting: PICO-Corpus)						
w/ labels	981	2,945	4,909	6,873	8,837	-
w/o labels	8,838	6,874	4,910	2,946	982	-
(Cross Domain setting: EBM-NLP)						
w/ labels	981	2,945	4,909	6,873	8,837	9,819
w/o labels	12,700	12,700	12,700	12,700	12,700	12,700
(All Domains setting: In Domain + Cross Domain)						
w/ labels	981	2,945	4,909	6,873	982	9,819
w/o labels	21,538	19,547	17,610	15,646	21,537	12,700
Validation	1,091	1,091	1,091	1,091	1,091	1,091
Testing corpus 1: PICO-Corpus						
	2,717	2,717	2,717	2,717	2,717	2,717
Testing corpus 2: AD + COVID-19						
	1,627	1,627	1,627	1,627	1,627	1,627

Supplementary Table 4: Entity counts of modified PICO scheme.

	PICO-Corpus	**EBM-NLP**	**AD**	**COVID-19**
Abstracts	**1,011**	**1,200**	**150**	**150**
Training	1010			
Validation	645			
Test	944			
Population (P)		3,951	215	262
Total sample size	1,094	-	-	-
Sample size in INT	887			
Sample size in CTL	784	-	-	-
Sex	1,991			
Age	231	-	-	-
Eligibility condition & criteria	1,252	-	-	-
Other demographics (location, ethnicity, race, etc.)	287	-	-	-
Intervention (I)	1,067	5,916	467	602
Control (C)	979	563	103	180
Outcome (O)		7,151	626	626
Names of study outcomes	6,134	-	-	-
Binary outcomes				
- Absolute value, INT/CTL	556/465	-	-	-
- Percentage values, INT/CTL	1,376/1,148	-	-	-
Continuous outcomes				
- Mean, INT/CTL	336/327	-	-	-
- Median, INT/CTL	270/247	-	-	-
- Standard deviation, INT/CTL	129/124	-	-	-
- Others, INT/CTL	8/8	-	-	-

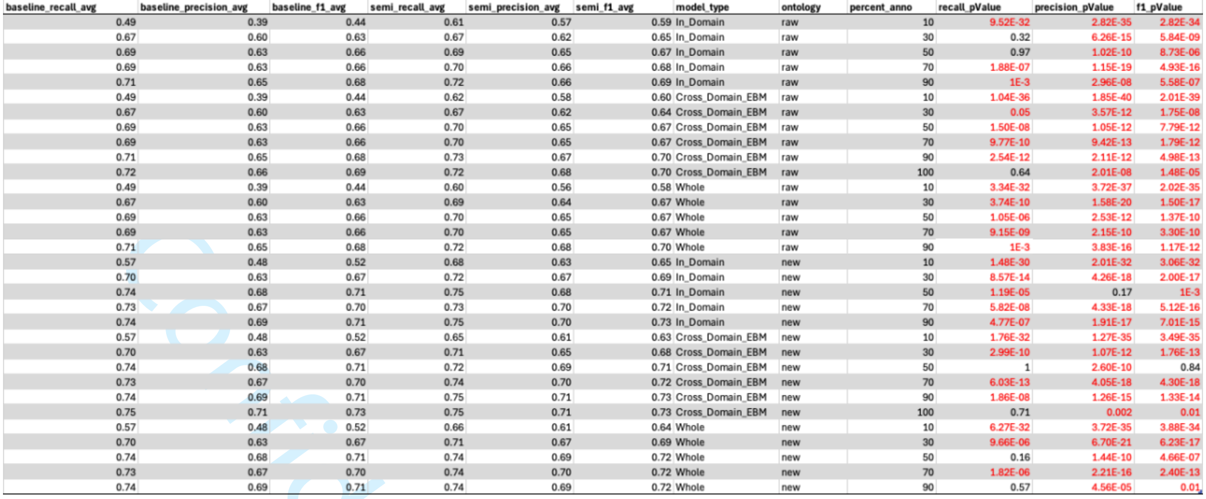
*INT: intervention arm. CTL: control group.

Supplementary Table 5. Performances of self-cleaning methods.



recall_avg	95_recall_CI_lower	95_recall_CI_upper	precision_avg	95_precision_CI_lower	95_precision_CI_upper	f1_avg	95_f1_CI_lower	95_f1_CI_upper	model_type	data_augumentation	PICO_scheme
0.636	0.605	0.667	0.561	0.609	0.664	0.596	0.609	0.664	Class adaptive	In_Domain	raw
0.617	0.585	0.648	0.571	0.589	0.645	0.593	0.589	0.645	Class adaptive	Cross_Domain_EBM	raw
0.682	0.654	0.709	0.626	0.657	0.707	0.653	0.657	0.706	Class adaptive	In_Domain	new
0.677	0.650	0.704	0.627	0.655	0.699	0.651	0.654	0.700	Class adaptive	Cross_Domain_EBM	new
0.607	0.576	0.638	0.566	0.578	0.636	0.586	0.578	0.636	Confident-base	In_Domain	raw
0.619	0.590	0.647	0.580	0.595	0.643	0.598	0.594	0.643	Confident-base	Cross_Domain_EBM	raw
0.675	0.646	0.705	0.628	0.648	0.703	0.651	0.648	0.703	Confident-base	In_Domain	new
0.652	0.626	0.677	0.613	0.627	0.676	0.632	0.628	0.675	Confident-base	Cross_Domain_EBM	new
0.607	0.578	0.635	0.591	0.578	0.635	0.599	0.580	0.633	gpt	In_Domain	raw
0.636	0.606	0.667	0.567	0.610	0.663	0.600	0.609	0.664	gpt	Cross_Domain_EBM	raw
0.639	0.614	0.664	0.607	0.612	0.665	0.622	0.615	0.663	gpt	In_Domain	new
0.613	0.587	0.638	0.608	0.586	0.639	0.610	0.589	0.637	gpt	Cross_Domain_EBM	new

Supplementary Table 6. Baseline model and FinePICO performance comparison.



baseline_recall_avg	baseline_precision_avg	baseline_f1_avg	semi_recall_avg	semi_precision_avg	semi_f1_avg	model_type	ontology	percent_anno	recall_pValue	precision_pValue	f1_pValue
0.49	0.39	0.44	0.61	0.57	0.59	In_Domain	raw	10	9.52E-32	2.82E-35	2.82E-34
0.67	0.60	0.63	0.67	0.62	0.65	In_Domain	raw	30	0.32	6.26E-15	5.84E-09
0.69	0.63	0.66	0.69	0.65	0.67	In_Domain	raw	50	0.97	1.02E-10	8.73E-06
0.69	0.63	0.66	0.70	0.66	0.68	In_Domain	raw	70	1.88E-07	1.15E-19	4.93E-16
0.71	0.65	0.68	0.72	0.66	0.69	In_Domain	raw	90	1E-3	2.96E-08	5.58E-07
0.49	0.39	0.44	0.62	0.58	0.60	Cross_Domain_EBM	raw	10	1.04E-36	1.85E-40	2.01E-39
0.67	0.60	0.63	0.67	0.62	0.64	Cross_Domain_EBM	raw	30	0.05	3.57E-12	1.75E-08
0.69	0.63	0.66	0.70	0.65	0.67	Cross_Domain_EBM	raw	50	1.50E-08	1.05E-12	7.79E-12
0.69	0.63	0.66	0.70	0.65	0.67	Cross_Domain_EBM	raw	70	9.77E-10	9.42E-13	1.79E-12
0.71	0.65	0.68	0.73	0.67	0.70	Cross_Domain_EBM	raw	90	2.54E-12	2.11E-12	4.98E-13
0.72	0.66	0.69	0.72	0.68	0.70	Cross_Domain_EBM	raw	100	0.64	2.01E-08	1.48E-05
0.49	0.39	0.44	0.60	0.56	0.58	Whole	raw	10	3.34E-32	3.72E-37	2.02E-35
0.67	0.60	0.63	0.69	0.64	0.67	Whole	raw	30	3.74E-10	1.58E-20	1.50E-17
0.69	0.63	0.66	0.70	0.65	0.67	Whole	raw	50	1.05E-06	2.53E-12	1.37E-10
0.69	0.63	0.66	0.70	0.65	0.67	Whole	raw	70	9.15E-09	2.15E-10	3.30E-10
0.71	0.65	0.68	0.72	0.68	0.70	Whole	raw	90	1E-3	3.83E-16	1.17E-12
0.57	0.48	0.52	0.68	0.63	0.65	In_Domain	new	10	1.48E-30	2.01E-32	3.06E-32
0.70	0.63	0.67	0.72	0.67	0.69	In_Domain	new	30	8.57E-14	4.26E-18	2.00E-17
0.74	0.68	0.71	0.75	0.68	0.71	In_Domain	new	50	1.19E-05	0.17	1E-3
0.73	0.67	0.70	0.73	0.70	0.72	In_Domain	new	70	5.82E-08	4.33E-18	5.12E-16
0.74	0.69	0.71	0.75	0.70	0.73	In_Domain	new	90	4.77E-07	1.91E-17	7.01E-15
0.57	0.48	0.52	0.65	0.61	0.63	Cross_Domain_EBM	new	10	1.76E-32	1.27E-35	3.49E-35
0.70	0.63	0.67	0.71	0.65	0.68	Cross_Domain_EBM	new	30	2.99E-10	1.07E-12	1.76E-13
0.74	0.68	0.71	0.72	0.69	0.71	Cross_Domain_EBM	new	50	1	2.60E-10	0.84
0.73	0.67	0.70	0.74	0.70	0.72	Cross_Domain_EBM	new	70	6.03E-13	4.05E-18	4.30E-18
0.74	0.69	0.71	0.75	0.71	0.73	Cross_Domain_EBM	new	90	1.86E-08	1.26E-15	1.33E-14
0.75	0.71	0.73	0.75	0.71	0.73	Cross_Domain_EBM	new	100	0.71	0.002	0.01
0.57	0.48	0.52	0.66	0.61	0.64	Whole	new	10	6.27E-32	3.72E-35	3.88E-34
0.70	0.63	0.67	0.71	0.67	0.69	Whole	new	30	9.66E-06	6.70E-21	6.23E-17
0.74	0.68	0.71	0.74	0.69	0.72	Whole	new	50	0.16	1.44E-10	4.66E-07
0.73	0.67	0.70	0.74	0.70	0.72	Whole	new	70	1.82E-06	2.21E-16	2.40E-13
0.74	0.69	0.71	0.74	0.69	0.72	Whole	new	90	0.57	4.56E-05	0.01