NERFAIR workflow: Towards systematic fairification on named entity recognition experiments

Yasmmin Martins[1*]

Corresponding author(s). E-mail(s): yasmmin.c.martins@gmail.com;

Abstract

Information extraction methods have been extensively applied in the biomedical scientific literature and a particular application is named entity recognition, which annotates these texts according to distinct knowledge areas. However, recent studies have demonstrated gaps in the limitation regarding standardization, automation, and reproducibility. A specific domain regarding NER application corresponds to the PICO (participant, intervention, control and outcome) framework that outlines important concepts lying in clinical trial publications, but few expert-curated annotations are available for training models, and often the corpora that are public concern specific diseases of interest.

In this paper, we present the NERFAIR workflow that uses a robust infrastructure for fine-tuning LLM models and handles the data processing from text snippets and annotations to the training, test and model predictions. It also provides a metadata enrichment module to enable the documentation of assets, model constraints, and configuration in line with interoperable standards for data exchange using our proposed extended ontology for NER experiments, and contributing to the experiment's full reproducibility, comprising the FAIR principles guidance. Complementarily, we also designed the CTPICO workflow that aims at generating trustworthy NER annotations for the PICO domain, taking into account clinical trials structured data. The workflow comprises the candidate generation part searching abstracts linked to up-to-date clinical trials; the annotation prediction using the NERFAIR workflow and then a validation procedure contrasting annotations with the specific parts of the structured data and ranking the most similar predicted annotations.

We exhaustively tested our NERFAIR workflow in benchmark NER datasets and also for the PICO dataset, demonstrating its module capabilities and surpassing previously reported evaluation metric values. We showed the augmented CTPICO dataset statistics, increasing the coverage for both clinical trials and PubMed abstracts. We also demonstrated the structural consistency, explainability and tested the correctness and completeness of our proposed ontology and showed example queries to traverse the NER experiments knowledge graph generated with the NERFAIR workflow module.

Keywords: NER workflow, fair principles, sparql query generation, retrieval augmented generation, Finetuning LLM, ontology design, reproducibility

1 Introduction

Information extraction stands for the process of analyzing textual information and detecting relevant information from this text related to the target contextual domain [1]. This type of analysis has been used for many purposes in the biomedical area, and several efforts were employed to extract, specifically, relations and named entity annotations from sentences [2]. Concerning the named entity recognition task, pretrained large language models like BioBERT [3], which are trained and fine-tuned using several scientific papers, can be retrained with the target domain dataset to adapt the model to extract the spans of the text related to the desired tags [4, 5]. These tags in the biomedical domains may be genes, proteins, diseases, drugs, clinical interventions, etc.

In [6], the authors highlighted a recurrent issue concerning NLP approaches, especially for the named entity recognition task, that fail to publish, either in the article or in the code repository, the necessary information to replicate their results, from the raw text preprocessing steps to the hyperparameters used to fine-tune a large language model. The same authors provided a detailed guideline, based on their own reproducibility experiment on replicating the results achieved by [7], describing all necessary pieces of information that a NER approach should provide in terms of the methods and parameter values applied in every stage of the NER task. These specific recommendations for the NER task are covered by the ones listed by [8] that comprise 40 recommendations of best practices for NLP tools aggregated by 5 topics of reproducibility (namely, traceability, versioning, standardization, usability, and shareability). This work showed that the six NLP tools being evaluated lack most of the items in their implementation and publishing process.

Considering these reported issues and recommendations to overcome them, we designed and implemented a domain-agnostic automated workflow for NER experiments based on LLM fine-tuning, which ensures reproducibility by documenting according to the FAIR standards the main parameters, the root general-purpose LLM model, the training hyperparameters, the achieved metrics, and the description of the data assets produced along the workflow execution. In line with the FAIR principles [9], we used the semantic web framework of resources to annotate and document the metadata regarding the experiment assets, and while mapping concepts across pertinent ontologies and vocabularies, some concepts related to the NER task were not covered, leading us to propose an ontology extension reusing mostly the XMLPO [10], STATO [11] and EDAM ontologies [12].

As a case study, we applied our workflow in the dataset concerning the PICO framework to compare our evaluation metrics with the reported ones in [13]. The few number of examples in the newest human-curated training dataset [7] for the application of NER task in this domain and the existence of continuously updated data concerning clinical trials suggested an opportunity to prepare a data augmentation methodology.

There are some methods that attempt to extract the PICO entities automatically, such as [7], in this case, the goal is to compare articles that describe the same event (intervention) across control and treated groups to feed a meta-analysis tool and perform statistical significance analysis. Recently, PICO annotation extraction methods focused on improving models to enhance the precision of detection, and some of them even generated new data but for specific diseases [14, 15]. These tools do not comprise the upstream candidate generation and downstream validation with the clinical trials' structured data. The closest published methodology for data augmentation [16], which presents a semisupervised learning strategy trained on a small dataset to generate new labels for a larger unlabeled one. This work also aims to achieve generalizability for any PICO framework. However, a systematic validation approach is not available to refine and enhance the human curation. Taking into account these gaps and limitations, complementarily, we designed and implemented a workflow specifically for the PICO Domain to generate new candidate annotations (dataset augmentation) in a disease-agnostic manner and validate the predictions achieved by the trained models on new published articles concerning clinical trial descriptions, taking advantage of the ground truth annotated information about them stored in the Clinical Trials (CT) API[1] maintained by the National Institute of Health (NIH). The main goal of this validation procedure is augmenting the original published PICO entities dataset according to newly published articles and refining the final annotation predictions by validating with clinical trials structured data.

2 Methods

2.1 NERFAIR Workflow

The proposed workflow (Figure 1) for the Named Entity Recognition NLP task comprises 5 main functionalities, which are 1) preprocessing of labeled spans of text together with the original plain text; 2) training procedure by fine-tuning a chosen pretrained large language model; 3) application of trained models in a new unseen test dataset; 4) application of trained models to predict entities in texts of a new corpus; and 5) statistical analysis concerning the metrics obtained during the model training to check robustness.

2.1.1 Data Preprocessing

This module is responsible for processing the raw text files with their respective entity annotations. These annotations contain the entity, the start and end positions, and the piece of text corresponding to this range. The BRAT tool [7] is used to parse the paired files (text-annotation) and transform them into the conll format, using the IOB (Inside, Outside, Begin) tagging format. In [13], the authors modified a part of the

[1] https://clinicaltrials.gov/data-api/api

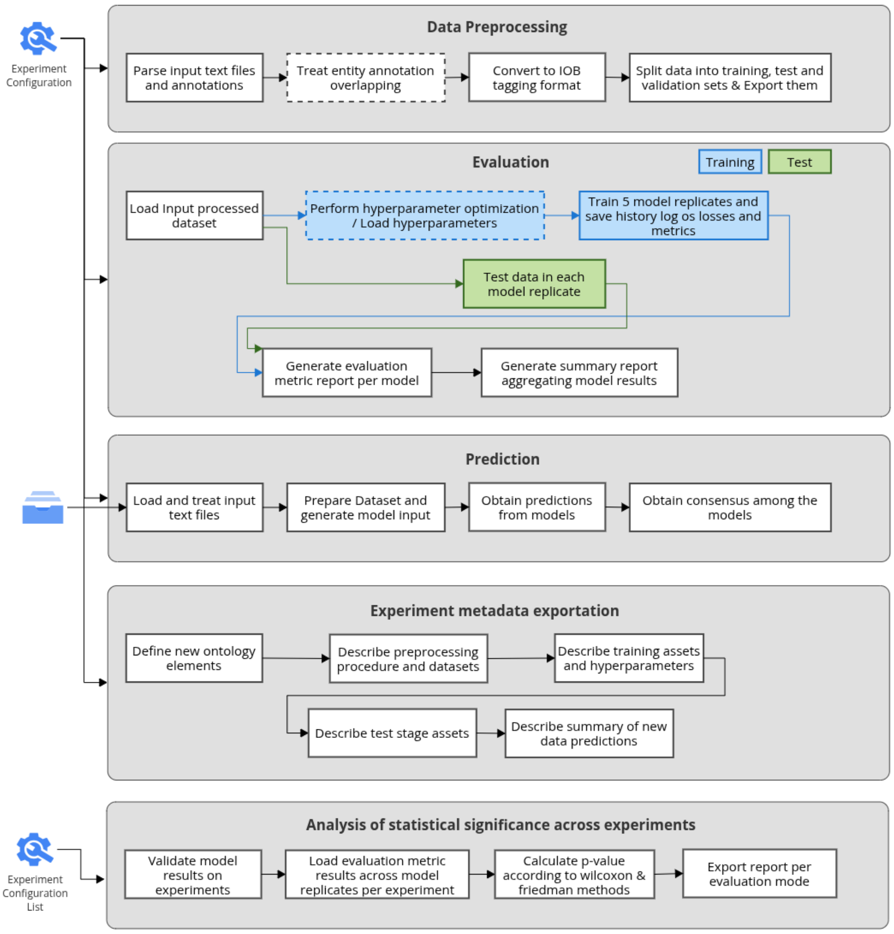

Experiment Configuration
Data Preprocessing
Parse input text files and annotations
Treat entity annotation overlapping
Convert to IOB tagging format
Split data into training, test and validation sets & Export them
Evaluation
Training
Test
Load Input processed dataset
Perform hyperparameter optimization / Load hyperparameters
Train 5 model replicates and save history log os losses and metrics
Test data in each model replicate
Generate evaluation metric report per model
Generate summary report aggregating model results
Prediction
Load and treat input text files
Prepare Dataset and generate model input
Obtain predictions from models
Obtain consensus among the models
Experiment metadata exportation
Define new ontology elements
Describe preprocessing procedure and datasets
Describe training assets and hyperparameters
Describe test stage assets
Describe summary of new data predictions
Analysis of statistical significance across experiments
Experiment Configuration List
Validate model results on experiments
Load evaluation metric results across model replicates per experiment
Calculate p-value according to wilcoxon & friedman methods
Export report per evaluation mode

Fig. 1 Diagram illustrating the five functionalities provided by the NERFAIR workflow.

conversion script to eliminate entity annotation overlapping, and in our workflow this procedure can be modified according to the configuration passed by the user.

The tagged produced files in conll format are processed to group the sentences and identify all the distinct NER tag sets through the annotations. The raw list is organized into three columns corresponding to the input file identifier, tokens, and their respective NER tags converted into numerical values. This list is then split into training (80%), validation (10%) and test (10%) and saved into a Dataset dictionary compatible with the transformers package. The NER tags of the words in each data row are stored as numerical values, but the original list of entities is stored as a dataset feature. This dataset is used by the training and test models of the evaluation module, but the raw list is also saved as a dataset without any split as an alternative. To transform and manipulate IO interactions, we reused some data processing utilities[2] presented in [13].

[2] https://github.com/cuevascarlos/ClinicalTrials/blob/main/utils_preprocessing.py

2.1.2 Evaluation

The workflow is designed to work with the large language models that enable multiple downstream natural language processing tasks; in our case, we are interested in the Named Entity Recognition. The user may configure the execution for any model published on Hugging Face [17], a repository of models and datasets, regarding this scope, such as the BioBERT [3].

According to the chosen model, we parse the input dataset and tokenize the data rows to transform the original words into a standard 512-length feature set with their respective numerical labels.

In the training mode, the user has an option of hyperparameter optimization activation or loading some precalculated hyperparameters. The possible items that may be fine-tuned are learning_rate, weight_decay, per_device_train_batch_size, per_device_eval_batch_size. If any of this is provided, the default values for each of these four items are 2e-05, 0.01, 16, 16, respectively. The Optuna framework [18] is used as the optimization method, and the range of values used for the hyperparameter search is the same as those recommended in [13]. We added the possibility of changing the objective metric for maximization; the default is the F1 score, but precision, recall and accuracy can also be used. The training process loads the available hyperparameters and uses the validation set of the split to evaluate the batches along the 40 epochs. To evaluate the robustness of the produced models, the training procedure is repeated five times.

In the test mode, it takes by default the test set from the split dataset and loads each of the five trained models to obtain the predictions. The user can also choose another external dataset for testing as long as it is compatible with the transformers dataset format.

For each model replicate, we generate a report concerning the evaluation metrics. The report takes into account four modes of evaluation according to the analysis granularity. The list of all possible NER tags comprises the prefix of the IOB tagging format plus the 24 desired domain classes. It is possible to use the traditional scikit-learn metrics considering a multi-class experiment or a specific library for NLP prediction evaluation (SeqEval)[3]. So, we perform the evaluation using scikit-learn with and without removing the class prefixes (B-, I-, O-) and use the metrics available by the SeqEval in the default and in the strict mode. The reported metrics using seqeval are f1, precision, recall and accuracy, whereas for scikit-learn we include these metrics but also the Mathews correlation coefficient [19], the Cohen kappa coefficient [20] and the area under the receiver operation characteristic (ROC) curve. The final report generated in this module aggregates the results of these evaluation metrics for all five models according to the minimum, maximum, average, median, and standard deviation statistical functions and also retains the summary of these metrics acquired for each entity to facilitate a posterior downstream analysis.

In the training mode, we compute these reports for the token level; however, in the test mode, we group the tokens into sentences and also evaluate according to the word level.

[3] https://github.com/chakki-works/seqeval

2.1.3 Prediction

This module takes as input a path to a directory containing plain text files, and the prediction input data is organized by the file names. For each model replicate, it loads its classifier and applies it to the pieces of text, generating a dataset consisting of the score, start and end positions, the predicted entity, and the word assigned for this entity. The prediction process can be executed in an HPC environment in case the user provides an HPC configuration file; in this case, the texts are split in an array job. Considering the results table generated individually per model, we provide a consensus report, grouping all the predictions identified by Pubmed ID, label, and word, and saving its score and start and end positions across the models. Once having the final list, we calculate the maximum, minimum, and average score values across the five models per prediction entry. From this information, a general report is stored with all the predictions, and a secondary top-ranked report is produced containing only the predictions whose average was above a predefined cutoff; its default value is 0.8 but the user may also modify it.

2.1.4 Experiment metadata exportation

In order to organize the concepts according to each layer of metadata information, we reused several available ontologies. One of these is a controlled vocabulary specifically designed for named entity recognition in the biomedical field that is named NERO [21]. Most of the classes and properties were taken from the XMLPO ontology [10] due to its coverage on describing the relationships among datasets, machine learning procedures such as training, test and prediction, and the aggregation of the hyperparameters and the algorithms used in the experiment. The evaluation metric concepts (mcc, f1-score, kappa, accuracy, precision, roc-auc and recall) used in our workflow reports were described (according to Table 1) using the STATO ontology [11], which is a general-purpose statistics ontology.

Table 1 Mapping between the evaluation metrics and the respective concept IDs in the STATO ontology.

Entity	STATO Concept ID
accuracy	0000415
precision	0000416
recall	0000233
aucroc	0000608
f1-score	0000628
mcc	0000524
kappa	0000630

However, we still needed to complement and added 15 object properties and 6 datatype ones (table with name, domain and range). We also created some classes to adapt the general ML purpose of XMLPO ontology to the natural language processing scope and the NER task, such as the NLPExperiment class being a subclass of Experiment and *NEREvaluationMeasure* as a kind of *ClassPredictionEvaluationMeasure*. We also extended the format branch from EDAM ontology [12] to describe tagging format types (IO, IOB, BIOES) used to preprocess raw annotated files for NER task. We also linked the xmlpo to the edam ontology by creating a class equivalence axiom between the *xmlpo:Operation* and *edam:operation_0004* (which also represents Operation), we represent the procedures executed along the pipeline.

The module annotates and organizes the knowledge graph according to the assets generated through the steps' execution. During the execution it looks for all the assets that are expected to be generated by the other steps of the workflow like the preprocessed datasets, the training and test evaluation reports, and the annotations dataset generated with the prediction module. As a result, it exports a semantic graph only with the ontology definitions and another one encompassing all the individual annotations along with the ontology definitions.

2.1.5 Analysis of statistical significance across experiments

This module tracks all the summary reports generated by the training and test steps according to the four evaluation modes and the two levels (8 combinations) of analysis, and for each experiment it extracts the model replicates values for all available evaluation metrics. Once this data is organized according to the evaluation method, we apply statistical measures to evaluate the significance and the null hypothesis's compliance or rejection on a pairwise analysis of these experiments.

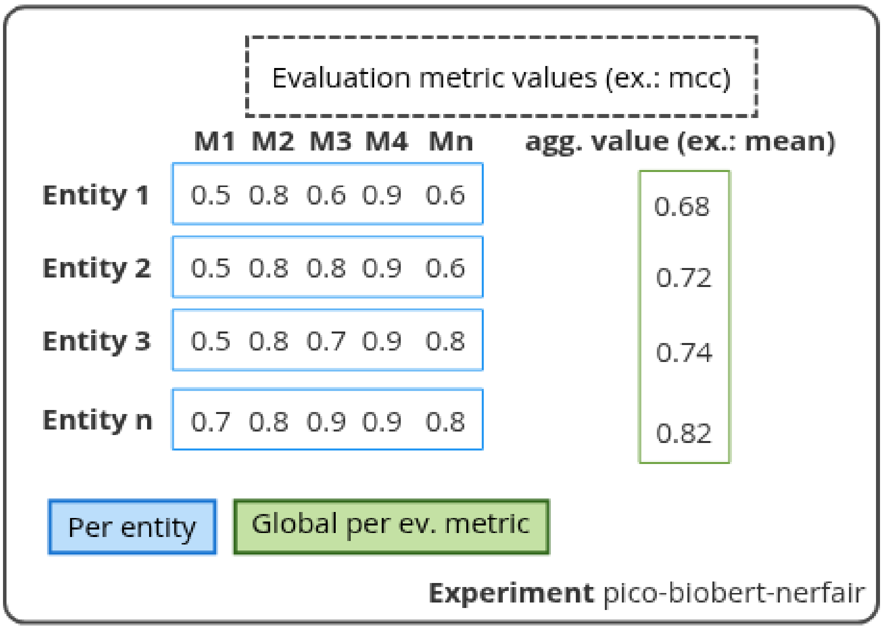
According to [22] the most recommended statistical tests to compare the reliability of the models on repeated cross-validations across training and valid datasets for multi-class classification (available entities for a given token) are the Wilcoxon and Friedman chi-square tests. We implemented both for the main evaluation metrics: F1-score, Mathews correlation coefficient, accuracy, precision, and recall.

This module computes the significance according to two levels of granularity (illustrated in Figure 2). The first level is per entity, where the values of each evaluation metric are tested using the model replicate results. The second level takes into account the aggregate value (minimum, maximum, median or mean) and catches for each evaluation metric (f1-score, accuracy, mcc, etc.) the score from model replicates for all entities and computes a global p-value for the evaluation metric.

The p-value is computed by pairwise comparison for the Wilcoxon with the goal of assessing if the changes in performance have a significant impact on the evaluation metrics. The Friedman chi-square-based analysis groups the lists of the series of values to be tested and obtains the p-value in the sense of testing consistency among the models of the experiments being tested. Each experiment identifier is considered an evaluator to be compared in this module, and it is possible to configure external evaluator metrics for each entity in case of a comparison with other results generated by published methodologies for the same context and entities.

2.2 PICO Domain augmentation workflow

The original PICO (Participation, Intervention, Control, Outcome) dataset comprises 1011 abstracts annotated by two experts; they considered four main categories and a total of 26 subcategories, considering the entities under each of the four groups,


Evaluation metric values (ex.: mcc)
Per entity
Global per ev. metric
Experiment pico-biobert-nerfair

	M1	M2	M3	M4	Mn	agg. value (ex.: mean)
Entity 1	0.5	0.8	0.6	0.9	0.6	0.68
Entity 2	0.5	0.8	0.8	0.9	0.6	0.72
Entity 3	0.5	0.8	0.7	0.9	0.8	0.74
Entity n	0.7	0.8	0.9	0.9	0.8	0.82

Fig. 2 Diagram showing the data generation to apply in the statistical tests. Using the values in the blue regions, the p-values will be reported separately for each entity across the evaluators. In the global mode, it will take the representative aggregated values of all entities using some grouping function (ex.: mean), and the p-value is provided per evaluation metric (ex.: mcc). These values show one experiment (evaluator), but the same rationale is applied for the other evaluators.

as described in [7]. Considering the difficulty of annotating new CT paper abstracts as they are released, we designed an augmentation workflow (Figure 3) that helps to automate the acquirement of annotations concerning the PICO entities, generating labeled text spans through a strict series of filters matching them against their respective CT data stored in the Clinical Trials API database.

2.2.1 CT data processing

Firstly, the pipeline extracts all completed clinical trials available in the clinical trials official API maintained by the NIH[4]. The retrieval acquires batches of 1000 records, adding a delay of 1 second to avoid the request blockage between the request submissions across the paginated results. The structured retrieved data contains sections with details of each clinical trial, and we focused on the following ones: the participants' characterization, the interventions applied for each stratified participant group, the outcome measures to evaluate the interventions, and the references to PubMed articles (when available). From this raw information, the parsing procedure generates a mapping between the CT identifiers and the PubMed identifiers of the papers included in the references.

[4] https://clinicaltrials.gov/data-api/api

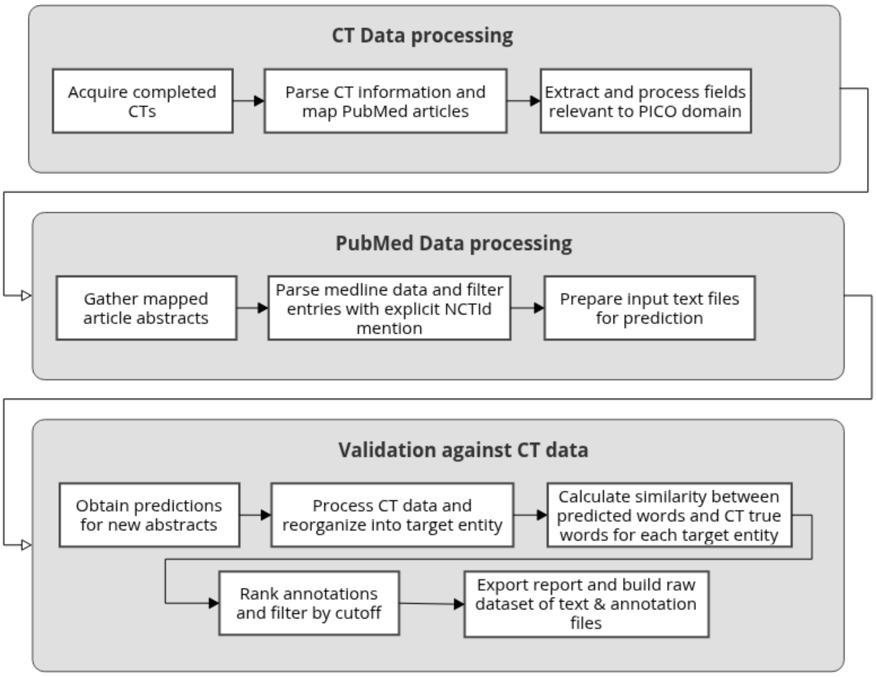

CT Data processing
Acquire completed CTs
Parse CT information and map PubMed articles
Extract and process fields relevant to PICO domain
PubMed Data processing
Gather mapped article abstracts
Parse medline data and filter entries with explicit NCTId mention
Prepare input text files for prediction
Validation against CT data
Obtain predictions for new abstracts
Process CT data and reorganize into target entity
Calculate similarity between predicted words and CT true words for each target entity
Rank annotations and filter by cutoff
Export report and build raw dataset of text & annotation files

Fig. 3 Diagram depicting the core tasks of each module of the workflow for validating CT entities annotations.

The PubMed article identifiers were applied to retrieve the abstract text and the publication type. The type of article and the corresponding clinical trial identifier inside the abstract text are used as criteria to filter the exact article that describes the CT from those that only cited it and do not fit in the clinical trial type.

The raw information extracted from the CT API was parsed to filter the specific information concerning the target labels of the NER task, and the eligibility criteria field was treated to mine each inner item from the entire text and to differentiate inclusion from exclusion criteria. But both types fit in the general eligibility entity type, so we concatenated to export the processed items into structured files.

2.2.2 PubMed data processing

The PubMed papers associated to the CTs are extracted from the NCBI API in batches of 500 at a time using the same delay among the requests.

The abstract paragraphs received in Medline XML format through the API are often labeled according to the human-readable sections in the browser. Therefore, we save the texts in files for prediction, naming these files by the PubMed ID combined with the section name (ex.: 16264159_INTERVENTIONS). These texts are treated to clean special character occurrences and quotes. The article identifier, publication type, label of the abstract piece, and the abstract piece are organized to prepare the prediction input files. This table-structured information is then filtered to extract only the texts from articles that have mentioned some CT identifier and whose publication type is related to a clinical trial report. The remaining records generate text files named by the publication ID and the section label.

2.2.3 Validation against CT data

This part of the workflow is activated after executing the NER FAIR workflow for prediction using the previously generated text files. The first task parses the entities and the text spans from the consensus table, considering the predictions agreed upon by the majority of models.

The labels and texts acquired form a dataset mapping the original article ID and the mentioned CT identifiers extracted by regular expression from the raw integral abstract text.

This dataset is used by the third task to calculate the pairwise similarity between each predicted text span and the values contained for the entity in the respective real CT data, filtered by the predicted entity.

The comparison between the predicted texts for the entities and the correct texts from the CTs for each available entity was performed by the cosine similarity metric [23]. To decide which metric has the best performance, we used the dataset of human-curated annotations against the CTs information and optimized the search of a metric using Optuna, according to two parameters: usage of string normalization and a range of eleven similarity metrics (Levenshtein, Damerau, Jaccard, Cosine, Jaro Winkler, Longest Common Subsequence, NGram, QGram, Optimal string alignment, Overlap coefficient, and Sorensen Dice) provided by the Python package strsimpy[5]. This evaluation yielded the best result using cosine and with string normalization, which is removing special characters and leaving only spaces, hyphens, letters, and numbers, and then applying the lowercase. The objective function computed the pairwise similarity of the strings and grouped the results by CT ID, PubMed ID, human annotation, and the human-assigned entity, aggregating and keeping one hit that had the maximum similarity score. The mean similarity value from this aggregated dataset was returned as being the value to be maximized. The optimization strategy is available as a fully reproducible pipeline[6] that handles the data acquirement and processing until the generation of the best trial parameters and graphical summary plots.

Once all the pairs are evaluated, the score table is aggregated by the CT ID, PubMed ID, the predicted entity, and the text snippet to gather only the hit found in CT data that produced the maximum similarity score. Then, these dataset items are ranked and ordered by similarity score. Finally, this ranked dataset is filtered with a cutoff value of 0.8 to generate the text files and their respective annotation files. The new augmented dataset follows the same pattern used in the input prediction, which is the concatenation of the PubMed ID and abstract section.

2.3 Ontology extension evaluation

We followed and adapted the evaluation procedure executed by [24] which comprises the assessment of explanation consistency, structural consistency, and correctness and

[5] https://github.com/luozhouyang/python-string-similarity/

[6] https://github.com/YasCoMa/pico_augmentation$_{w}orkflow/blob/master/supplementary\_scripts/optimization\_choice\_string\_met$

completeness based on some competence questions to check whether the concepts and properties modeled in the ontology are enough to correctly answer these questions.

Concerning the explanation consistency, we also used the Hermmit reasoner, but it was executed under the Owlready2 python package[7] automatically after loading the rdf files exported as results of the metadata enrichment for all the experiments conducted to evaluate the NERFAIR workflow. The structural consistency was also assessed using the OOPS! [25] online tool[8]. We exported the ontology declarations of classes and properties that we extended together with the axioms and scanned with this tool that analyzes 41 criteria classified according to the levels of importance (critical, important, and minor).

Regarding the correctness and completeness, based on the features available along the execution steps of the workflow and the data generated and exported in tabulated tables. We elaborated ten competence questions (CQs) to test if we retrieved the reproducible minimum required items recommended for experiment reproducibility. So, the competency questions focus on retrieving the available models, the data splits and the quantity of examples, the evaluation strategies, and the score values of the evaluation metrics. Using these CQs, we evaluated the ontology capability to cover the paths in the graph to answer them correctly. As a second stage of the analysis, we compared the prepared sparql queries prepared by human with those queries generated using LLM model through the langchain python package[9]. More specifically, we implemented the integration with the SPARQL retriever to formulate the context from a knowledge graph, enabling the agent to formulate the answer[10]. The models used for question answering in the retrieval augmented generation exercise were the gemini 2.0 flash[11] from google and the llama3.2[12] and mistral[13] from the Ollama library because they have been used for NLP tasks assessment [26] and employed for retrieval augmented generation in clinical context [27]. They are also light and can be run on a regular personal computer (16gb RAM memory, 1Tb hard disk space). The other important factor for choosing only one model outside ollama library is the token usage limitation, since the sparql generation needs to send the tokens corresponding to the entire graph to be able to produce the sparql query.

According to google AI tariffs[14], in the free version, this model allows up to 1 million tokens per request, whereas the others are limited by 250,000. This exercise presented the knowledge graph fed with ontology and individual annotations as a context, and we prompted the same CQs in natural language. Then, we compared the query itself, and if this was valid, we checked if the results are in line with those produced from human queries. The comparison checked the following items: 1) Ontology prefixes assigned to the respective classes and properties; 2) Sufficiency of statements

[7] https://owlready2.readthedocs.io/en/v0.48/reasoning.html
[8] https://oops.linkeddata.es/catalogue.jsp , acessed 17 october 2025
[9] https://docs.langchain.com/oss/python/langchain/overview
[10] https://python.langchain.com/docs/integrations/graphs/rdflib_sparql/
[11] https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash
[12] https://ollama.com/library/llama3.2
[13] https://ollama.com/library/mistral
[14] https://ai.google.dev/gemini-api/docs/rate-limits

in the “where” to return the required information; 3) Conformity of resource placement in the statements regarding the domain and range constraints of the property linking them.

3 Results

3.1 Evaluation setup

All the experiments were executed in a CPU-based node with the processor AMD EPYC 7453 28-Core, containing 56 CPUs and 1Tb of RAM memory. We evaluated the NERFAIR and the PICO Domain augmentation workflows by executing the following four experiments:

- **Evaluation on PICO Dataset.** We used the original PICO dataset that was annotated by human experts and we evaluated the impact of hyperparameter optimization using only CPU-based training. We also compared the optimized values and the performance based on F1-score with the ones reported in previous studies [7, 13].
- **Experiment on NER benchmark datasets.** The NERFAIR was evaluated in four benchmark datasets, namely bc5cdr [28], chiads [29], ncbi [30] and biored [31], concerning the NER task in biomedical texts [32]. We evaluated two performance evaluation metrics, which are F1-score and MCC. We established a standard procedure to process these datasets, allowing for the experiment’s reproducibility, from the raw downloaded data until achieving the final result reports. We also aimed at documenting the metadata of these datasets with the underrepresented entities. Finally, we also carried out an analysis merging the raw data for training provided for the four datasets to test whether the performance for the entities would improve, and the test set was also merged in a separate dataset to avoid data leakage.
- **Experiment on PICO dataset augmentation.** We firstly verified the optimized search for the best string similarity metric on the human-curated PICO dataset for matching the pairwise combinations of CT items and annotations for each entity. We then report the results for 300,500 completed clinical trials published up to August 14, 2025. The evaluation comprised the verification of the ranked matching pairs according to the similarity value distribution.
- **Experiment of semantic metadata enrichment.** We used the Experiment metadata exportation module of the nerfairwf workflow to generate and export the details about each step of the workflow, from the data preprocessing constraints up to the detailed performance evaluation metrics in the summary reports. We documented the specification of the concepts added to the original XMLPO ontology [10] to adapt to the NER NLP task, the metrics regarding the number of classes and instances considering the experiments conducted in the PICO dataset and in the NER benchmark datasets. We evaluated the ontology extension regarding explanation and structural consistency. Some examples of queries were prepared to navigate through the knowledge graph and evaluate the individual annotations with respect to correctness and completeness.

3.2 Evaluation on PICO Dataset

Using the same dataset that was evaluated in a previous study [13], we tested our pipeline performing the hyperparameter optimization to check the differences in these parameter values and compare the values of the F1-score metric.

Although we trained and optimized without GPU usage, Table 2 shows that the values found for "per device train batch size" and "per device eval batch size" were exactly the same as the ones in Experiment-2. The learning rate value difference regarding experiment-2 was only 0.923. Interestingly, the value for the weight decay parameter obtained the highest reduction, decreasing from 1e-2 in experiment-1 to 1.013e-4 in our experiment.

Table 2 Comparison of the four hyperparameters used in the optimization study in the biobert model. Experiment-1 and Experiment-2 correspond to the values acquired by the study, fixing values for the hyperparameters (Experiment-1), and performing the optimization according to the available hardware architecture (Experiment-2). The column "nerfairwf" refers to the values obtained by our workflow.

Parameter/Experiment	Experiment-1	Experiment-2	NERFAIR
Learning rate	2e-5	4.976e-5	4.053e-5
Weigth decay	1e-2	3e-3	1.013e-4
per device train batch size	16	8	8
per device eval batch size	16	16	16

We used the same hyperparameters found here for all the experiments reported in this paper. According to table 3, We observed that apart from the ethnicity and cv-cont-mean, our workflow was able to reach the highest F1-score value for all the entities. In most cases (91%, 20 out of 22 entities), "Experiment-1" and "Experiment-2" obtained values lower than the original study. We obtained the lowest value (0.64) only for the *ethinicity* entity* considering the other approaches, whereas for the *cv-cont-mean* we reached the absolute minimum value (0.57) but we improved the scores observed in Experiments 1 and 2 for this class reached 0.47 and 0.41, respectively. According to Supplementary table 1, there are only 83 articles from the 1011 and 101 annotations for this entity; this is one of the most underrepresented classes. In total, there are 26 entities; however, there are only 4 annotations for the entities *iv-cont-q1*, *cv-cont-q1*, *iv-cont-q3* and *cv-cont-q3*.

Finally, We compared the f1-score mean values of the three published methodologies (PICO original reference, Experiment-1 and Experiment-2) using the significance module of the nerfair workflow. Since the f1-scores for the model replicates are not available, we compared putting together the mean values for the entities in the same order for the token level in the test scope. According to supplementary table 2, the Wilcoxon paired test showed that apart from the seqeval evaluation technique in the strict mode (with entity prefixes), the models produced F1-scores significantly distinct when comparing our trained models against the external evaluators, with p-values under 0.01. The p-values also reflected the closeness between our results and the original PICO reference values, and for all the evaluation modes the global p-value

Table 3 Comparison of the mean and standard deviation values obtained for the entities in the two experiments of the reproducibility study, the PICO dataset original study and our workflow. The original study had not reported the standard deviation.

Entity/Experiment	**PICO-Reference**	**Experiment-1**	**Experiment-2**	**NERFAIR**
total-participants	0.94	0.9065 (+- 0.0096)	0.9313 (+- 0.0048)	0.9601 (+- 0.0048)
intervention-participants	0.85	0.7431 (+- 0.0123)	0.8177 (+- 0.0135)	0.8511 (+- 0.0143)
control-participants	0.88	0.7846 (+- 0.0108)	0.8480 (+- 0.0124)	0.8561 (+- 0.0154)
age	0.80	0.5638 (+- 0.0300)	0.5724 (+- 0.0731)	0.9283 (+- 0.0113)
eligibility	0.74	0.6049 (+- 0.0131)	0.6382 (+- 0.0219)	0.8840 (+- 0.0161)
ethinicity	0.88	0.7135 (+- 0.0433)	0.7163 (+- 0.0353)	0.6399 (+- 0.0137)
condition	0.80	0.6412 (+- 0.0469)	0.7122 (+- 0.0421)	0.8931 (+- 0.0174)
location	0.76	0.6156 (+- 0.0226)	0.6258 (+- 0.0363)	0.8791 (+- 0.0157)
intervention	0.84	0.7805 (+- 0.0047)	0.7899 (+- 0.0095)	0.8517 (+- 0.0049)
control	0.76	0.6780 (+- 0.0205)	0.6529 (+- 0.0190)	0.8007 (+- 0.0096)
outcome	0.81	0.6321 (+- 0.0056)	0.6667 (+- 0.0151)	0.8792 (+- 0.0037)
outcome-Measure	0.84	0.7441 (+- 0.0274)	0.8003 (+- 0.0240)	0.9665 (+- 0.0036)
iv-bin-abs	0.80	0.6184 (+- 0.0278)	0.7640 (+- 0.0352)	0.9099 (+- 0.0109)
cv-bin-abs	0.82	0.6557 (+- 0.0214)	0.8195 (+- 0.0219)	0.8925 (+- 0.0338)
iv-bin-percent	0.87	0.6460 (+- 0.0174)	0.6731 (+- 0.0317)	0.8432 (+- 0.0142)
cv-bin-percent	0.88	0.6919 (+- 0.0224)	0.7549 (+- 0.0233)	0.8531 (+- 0.0153)
iv-cont-mean	0.81	0.5081 (+- 0.0352)	0.4271 (+- 0.0334)	0.6872 (+- 0.0678)
cv-cont-mean	0.86	0.4711 (+- 0.0160)	0.4117 (+- 0.0297)	0.5737 (+- 0.0627)
iv-cont-median	0.75	0.6630 (+- 0.0336)	0.7415 (+- 0.0216)	0.8288 (+- 0.0198)
cv-cont-median	0.79	0.6937 (+- 0.0195)	0.7769 (+- 0.0373)	0.8312 (+- 0.0095)
iv-cont-sd	0.83	0.4606 (+- 0.0424)	0.6274 (+- 0.0683)	0.8988 (+- 0.0385)
cv-cont-sd	0.82	0.4711 (+- 0.0514)	0.7264 (+- 0.0826)	0.8731 (+- 0.0406)

regarding this evaluator ranged from 0.02 to 0.55. The minimum and maximum p-values correspond to the evaluation modes Scikit-learn without prefix and seqeval in strict mode, respectively. Since the first evaluation mode rendered the highest evaluation metric values, it was expected that its p-value would be lower than in the other modes. The Friedman chi-square statistical test also rejected the null hypothesis, showing that the distribution of F1 scores for the entities coming from the distinct evaluators are not homogeneous and differ among them. These findings corroborate the above-mentioned results, showing that our model contributed positively and significantly to improving the F1-score results.

3.3 Experiment on NER benchmark datasets

In this experiment, the goal was evaluating our workflow in related biomedical NER datasets [32] as a benchmark analysis. We also created a pipeline[15] to process the raw dataset files and generate the input files needed for our workflow to enable the reproducibility of this experiment. Due to the improvement observed in the evaluation metrics reported in the experiment for PICO dataset, we kept the same hyperparameter values.

We provide four evaluation modes in the workflow, and we ranked the evaluation modes taking into consideration the values of two target evaluation metrics, which were F1-score and MCC (matthews correlation coefficient). We set a cutoff of 0.5 in Figure 4 to show only the classes that obtained at least 0.5, and in the chiads plot there were no representatives for the token level for the entities *Informed_consent* and *Multiplier*. The complete results for MCC and F1-score obtained for all levels and datasets can be found in Supplementary table 3.

As expected and mentioned in [13], the lenient evaluation modes that only take into account the labels and not the prefixes (B, I or O) are the ones that produce the highest values of F1-score (with mean values per dataset and levels ranging from 0.81 to 0.99) considering the four evaluation modes, since the constraints are more relaxed. Seqeval evaluation mode does not provide the MCC metric values, so for this metric the ranking was calculated only between the scikit-learn modes. The minimum and maximum values obtained across the datasets and levels for the MCC metric were 0.55 and 0.92, respectively. The minimum values for both evaluation metrics occurred in the chiads dataset, which has 30 different classes. The lowest performance results were indeed obtained by the models with more number of possible entities, such as the chiads and the merged_train.

The comparative analysis between the merged_train model and each individual model for the four datasets (Supplementary Table 4), grouped by level, metric and dataset, showed that the merged_model only improved the metric values for chiads entities, with a p-value of 0.009. While the f1-scores of 11 entities (word or token levels) were enhanced by the merged model, the MCC values followed the tendency for 29 entities. The differences in the F1-score values were more expressive (difference above 0.05) for the entities *Parsing_Error* (0.25 to 0.45), Visit (0.60 to 0.74) and *Pregnancy_considerations* (0.40 to 0.52). The results for the most significant entities regarding biomedical domain (*Chemical*, *Drug*, *Condition*, *Measurement* and *Disease*) remained with a performance above 0.7 in the merged model results. In the ncbi dataset disease concept is split into disease class and specific disease, and the model lost the performance in both favoring the major disease class from bc5cdr dataset. Besides, the merged model could not improve the metrics for all the entities; it could still generalize the learning for the core biomedical entities and needed more examples for the other classes to improve the evaluation metric values for them.

[15] https://github.com/YasCoMa/ner-fair-workflow/tree/master/downstream_experiments/prepare_benchmark_datasets

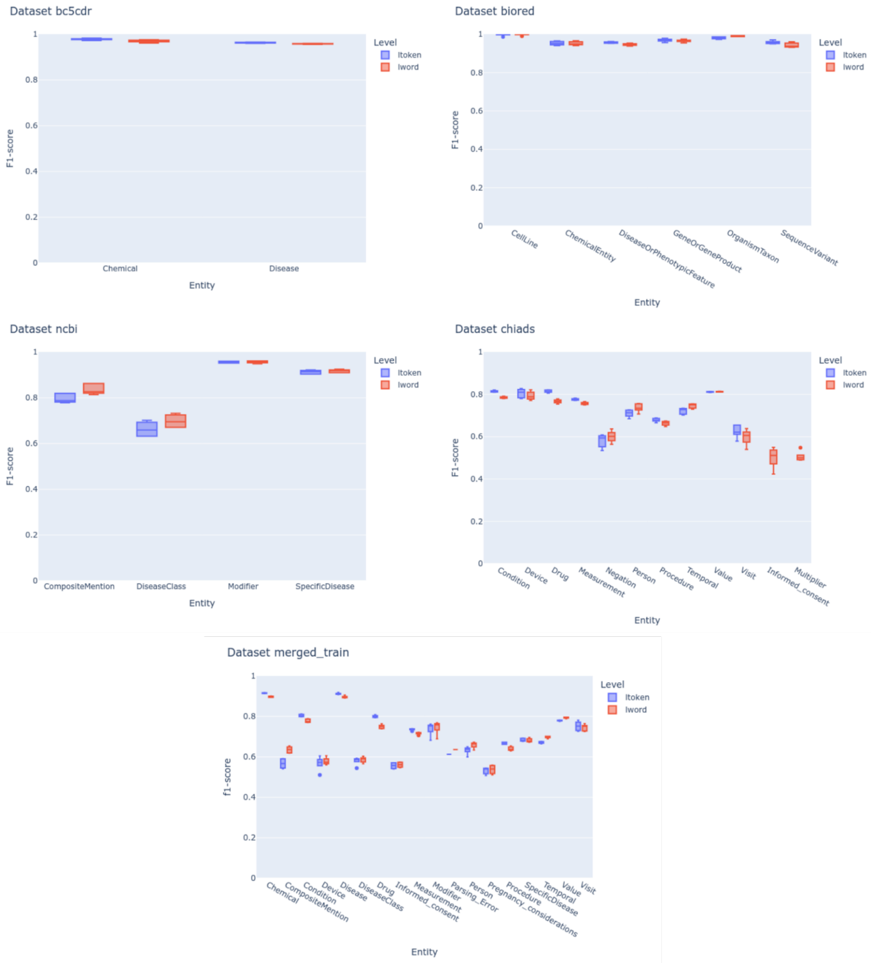


Fig. 4 Comparison of F1-score values obtained for the entities in each of the four datasets (bc5cdr, ncbi, chiads and biored) and the merged model with all training examples provided by these datasets at the word and token levels.

3.4 Experiment on PICO dataset augmentation

3.4.1 Optimization of string similarity metrics

The validation workflow for the PICO entities concerning clinical trials information was evaluated firstly on the human-curated dataset. From the 1011 abstracts of PubMed papers, only 129 abstract texts contained a direct mention of the respective clinical trial identifier. We followed the analysis with these texts, and it covered 117 registered clinical trials, and we considered this dataset as our gold standard.

We performed an evaluation concerning the validation strategy in which the goal was to optimize the choice of text processing and string similarity metric in order to minimize the pairwise text item similarity for each entity across CT information and the human-curated labels in the gold dataset. According to the optimization trials, the combination that yielded the best score was the cosine similarity applying a transformation function to remove soft variations across strings that are being tested. The optimization study computed the global mean of similarities grouped by the top 1 value for each annotation, and Figure 5 shows the distribution of computed cosine similarity values per entity. In most of the cases, the average similarity was shifted and improved by the text normalization process. According to this optimization analysis, the cosine similarity metric was fixed in the PICO Domain validation workflow.

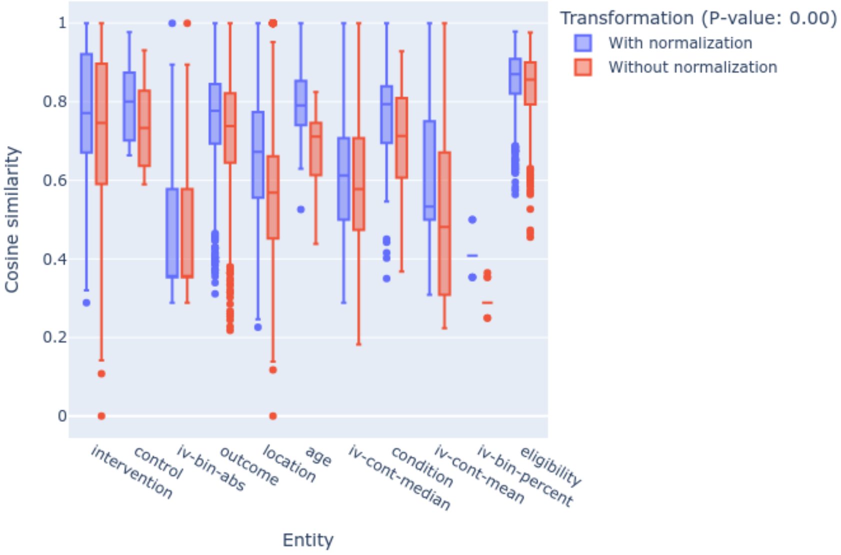


Fig. 5 Distribution of cosine similarity across the entities, grouped by the transformation process, showing the significant improvement on the matchings induced by the string pair normalization.

3.4.2 Application of the PICO Domain validation workflow

The NER fair workflow was applied in a case study concerning 24 entities found on article abstract texts in the clinical trials scope. The experiment goal was the augmentation of the original PICO dataset, which was manually curated, by applying a validation pipeline on the predictions gathered from new CT abstracts. The validation pipeline uses the original correct information of the CT annotated in the NIH clinical trial API according to the respective property related to the target entity to check the similarity score against the predicted sentence span.

We gathered information comprising 300,500 completed clinical trials up to August 14, 2025. However, only 116,755 entries contained references to 402,374 articles published on PubMed, and from these articles, we were able to extract 381,688 complete abstract texts with their respective publication types. A total of 87486 abstract texts passed the clinical trial publication type and a direct reference to the clinical trial identifiers (64091) inside the sentences. The number of clinical trials decreased because not all the references in the CT-structured file refer to the publication that is directly associated and describes the clinical trial.

From these clinical trial structured files, only 17866 of them have the results section, which reduces the number of annotations for the entities based on the outcome measures. A total of 62037 contained the minimum number of sections to extract the true values for each entity.

These abstract texts were evaluated by the prediction module of the nerfairwf workflow and Table 4 shows overall counts of distinct papers and annotations retrieved in each step of the validation procedure. The intervention, outcome and eligibility entities were identified in about 85% of the available abstract texts and, as expected, produced the largest number of top ranked annotations, which were 116429, 288816 and 176567.

The high count values, surpassing 1 million for the outcome, eligibility, and location entities, are correlated to the overall pairwise pairs compared in the similarity analysis and the number of items extracted for testing in each entity from the clinical trial data. The complete table of entity counts stratified by clinical trial data can be found in supplementary table 5, and Figure 6 shows a summary of the entity item counts considering all the available clinical trials. which shows the same trend observed for the three entities that produced the largest diversity of top-ranked annotations.

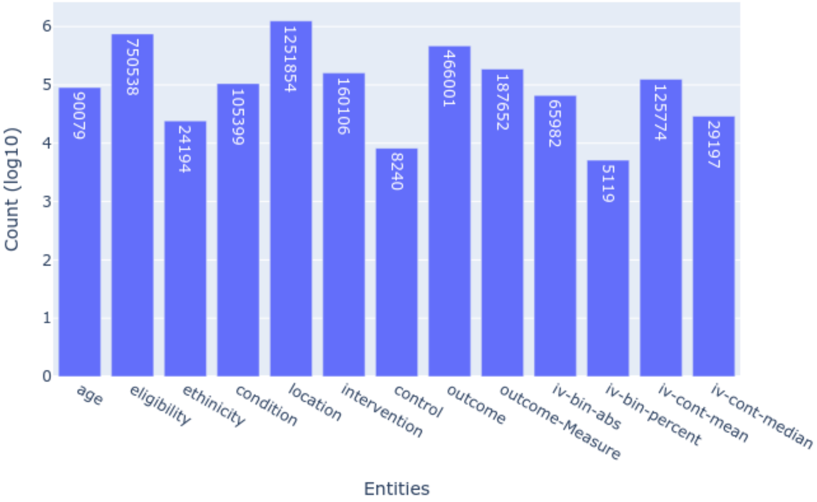


Fig. 6 Number of entity items extracted in the available clinical trials data. While the normalized values are shown on the y-axis, the absolute values are found inside each bar.

We compared the cosine string similarity value distributions between the grouped validated and the grouped gold standard datasets to check the behavior for each entity. Figure 7 shows that the similarity value distributions overlap for both datasets in practically all entities except for *iv-bin-percent*, which is the most underrepresented entity in the clinical trials data (5119 values extracted in only 1191 clinical trials). Although *age*, *outcome*, *outcome-Measure* and *eligibility* have high mean similarity values (above 0.7), they comprise many outlier samples with similarity values ranging from 0.3 to 0.45.

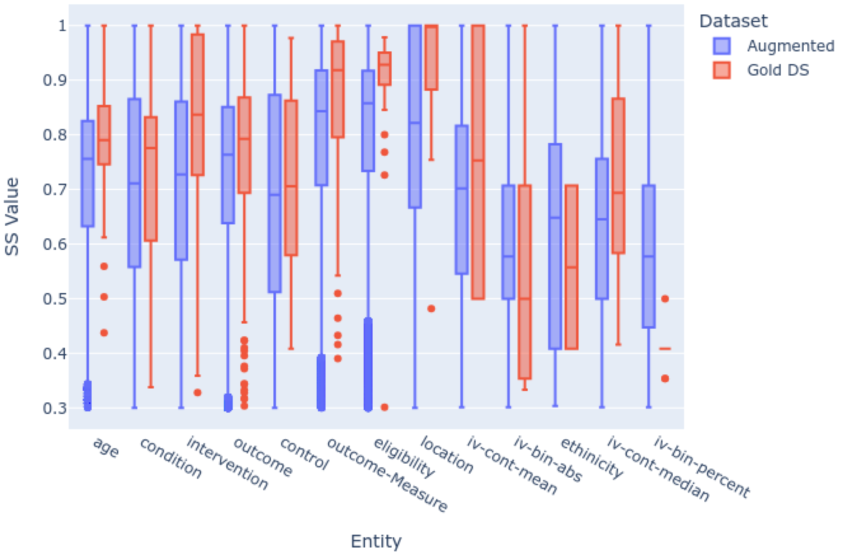


Fig. 7 String similarity values distribution for the 13 entities present in the validation results for the Augmented PICO dataset and the original gold dataset.

The original human-curated dataset had 26 distinct entities; however, the augmentation validation results contain only the 13 entities that were retrieved in the clinical trial data. Even cutting in half and filtering the annotations with a similarity value above 80% regarding the true items matched in the clinical trial, we were able to augment the original dataset surpassing the original dataset numbers in 10 out of 13 entities (Figure 8). The panel shows that we increased the diversity and coverage of papers and annotations, and the workflow allows for the automatic update acquiring and processing both recent clinical trials and PubMed abstracts data, especially for the *outcome* (+68514), *eligibility* (+74096), *intervention* (+62697) and *condition* (+25932) entities. The augmented dataset annotations and respective text snippets from the PubMed abstracts are publicly available at https://zenodo.org/records/17437153.

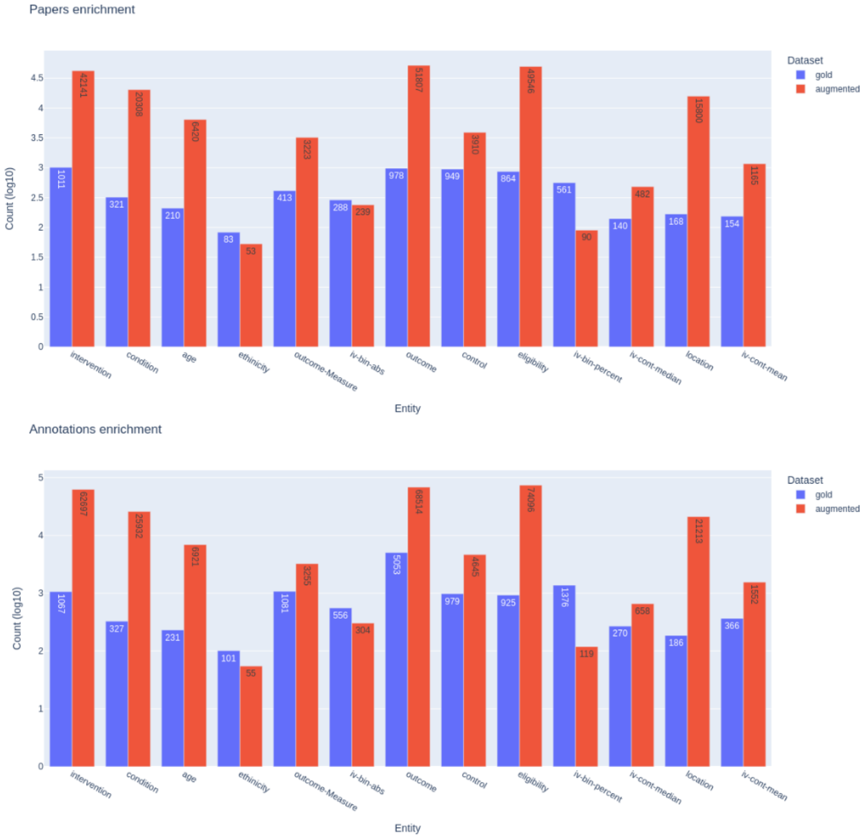


Fig. 8 Panel comparing the total papers and annotations count in the top-ranked and in the gold datasets.

3.5 Experiment of semantic metadata enrichment

Firstly, we have created classes and properties to extend the original XMLPO ontology [10] to describe specific features of natural language procedures and the named entity recognition tasks. This ontology was originally designed to describe machine learning workflow experiments and derive explainable parameters. We included references to other related ontologies in order to be in compliance with the linked open data principles [33] and reuse as many concepts as possible from external ontologies such as edam [12], nero [21], STATO [11], MeSH[16], and National Cancer Institute Thesaurus (ncit)[17]. We created six new classes, 15 object properties, and 6 datatype properties. The extension also comprises some relationships connecting concepts that have equivalencies in other ontologies, like the *Operation* class in XMLPO and the *operation_0004* one in EDAM ontology. We completed the hierarchy (Figure 9) from

[16] https://bioportal.bioontology.org/ontologies/MESH

[17] https://bioportal.bioontology.org/ontologies/NCIT

the general class model to the specific LLM model that comprehends the type of model that the NERFAIR workflow trains and evaluates.

We annotated the main procedures concerning the workflow functionalities (preprocessing, train, test and prediction) using the operation concepts that were described in edam and xmlpo ontologies. The newly created object property *containsProcedure* creates a link between the workflow and these operations. For the preprocessing, *edam:operation_1812* and *edam:operation_0335* identifiers represented the data parsing and the data formatting to the IOB structure, and xmlpo:DatasetSplit represented the exportation of the split datasets compatible with transformer models. The train, test and prediction workflow modules were covered by the *xmlpo:Train*, *xmlpo:Test* and *edam:operation_2423* (Prediction and recognition), respectively. The datasets and the model evaluation results are linked to the workflow through these operation nodes.

As an overview (Table 5), the proposed extension of published ontologies to address NLP experiments with LLM models contains 64 constraint declarations in the ontology element definitions, 62 classes, 33 object properties creating the relationship among these classes, and 12 datatype properties to annotate their attributes.

Considering the merged graph[18] from the experiments with PICO dataset and the datasets of the benchmark analysis, the number of individual annotations is 51215. The pipeline to replicate the metadata enrichment module execution and merge the knowledge graph from the mentioned datasets can be found at https://github.com/YasCoMa/ner-fair-workflow/tree/master/downstream_experiments/semantic_enrichment. The ontology extension definition was serialized in rdf/xml[19], and the tables with the property[20] and class[21] information are also publicly available.

3.5.1 Structural consistency

The structural consistency results provided by the OOPS tool detected 9 pitfalls: one of them was critical, five are classified as important, and three are considered minor issues.

- P08 (Missing annotations) - There were five classes (*NEREvaluationMeasure*, *NLPExperiment*, *ValidationSet*, *SummaryPrediction*, *TaggingFormat* and *MLExperiment*) in the ontology that lacked a definition or a comment. We annotated them and clarified their concept usages.
- P10 (Missing disjointness) - There were no declarations of disjoint classes, so we analyzed in detail in which new classes this axiom was applicable. We declared statements for disjointness among the classes *nf:ValidationSet*, *xmlpo:TrainSet* and *xmlpo:TestSet*.
- P11 (Missing domain or range in properties) - There was one property (*nf:executedBy*) in our ontology extension that lacked the domain and range declaration. The domain and range were declared accordingly, linking *nf:NLPExperiment* (domain) to *xmlpo:workflow* (range).

[18] https://github.com/YasCoMa/ner-fair-workflow/blob/master/results/all$_n erfair_g raph.ttl$
[19] https://github.com/YasCoMa/ner-fair-workflow/blob/master/results/nerml_ontology.xml
[20] https://github.com/YasCoMa/ner-fair-workflow/blob/master/results/properties_information.tsv
[21] https://github.com/YasCoMa/ner-fair-workflow/blob/master/results/classes_information.tsv

- P13 (Inverse relationships not explicitly declared) - There were no statements adding the inverse property axiom. The tool recommended originally fifteen candidate properties to originate their inverse properties; however, only three of these inversions were plausible. We appended the properties *nf:executesExperiment*, *nf:describesFeatureOf* and *nf:isParameterOf* as inverse properties of *nf:executedBy*, *nf:describedBy* and *nf:hasParameter*, respectively. One of the cases was actually a transitive property (*nf:finetunedFrom*), which links instances of *mesh:D000098342* (LLM model).
- P22 (Using different naming conventions in the ontology) - While we were completing the connection of concepts traversing distinct classes and properties from other ontologies, it identified some concepts using camel case and others using underscore. We ignored these alerts since all the new elements followed the camel case pattern.
- P34 (Untyped class) - This issue is related to classes being referred to without being explicitly declared as being instances of *owl:Class*. However, the 15 classes pointed out by the tool are classes from other ontologies that we integrated by associating their public prefix and URI.
- P38 (No OWL ontology declaration) - This issue was fixed by adding a statement linking the URI of the ontology resource as an instance of *owl:Ontology* class.
- P39 (Ambiguous namespace) - The exportation of the xml serialization of the ontology missed the ontology URI in the xml:base attribute of the root tag. We fixed adding this information.
- P41 (No license declared) - We addressed this pitfall by adding a statement linking the ontology instance (from P38 fix) to the Creative Commons license version 4.0[22], using the property license from the Dublin Core Terms vocabulary[23].

3.5.2 Explanation consistency

We assessed the explanation consistency by incrementing the difficulty level in the following manner: (1) only the ontology definition of classes, properties, and axioms; (2) the ontology element definitions and the individuals for one of the experiments that we performed (ncbi) for the other evaluation parts of the workflow; lastly, (3) we integrated the ontology definitions and created a global knowledge graph with all the triples concerning the graphs generated for all experiments. The goal of this analysis in levels is dealing with the possible inconsistencies in a controlled and paced way.

In the level (1), the reasoner complained about some classes that were reused as part of domain or range definitions of the new object properties. We redeclared these classes as instances of owl:Class and added label and comment to remain in compliance with the previous structural consistency analysis. Another issue that was fixed in this level was the addition of the base IRI that is not automatically exported in the rdf/xml serialization of the graph by the RDFLib package. The reasoner executed correctly after these two changes.

In the level (2), we transferred these corrections to the single-experiment graph, and there were 1378 inconsistencies, and they all had the same cause, which is the range declaration of the new data type properties. Originally, they were declared with

[22] http://creativecommons.org/licenses/by/4.0

[23] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

the specific XML schema variable type (boolean, integer and string), but the individual annotations were assigned using the literal format from the RDF schema vocabulary. We changed the range definition of all data properties to be in line with the automatic annotation. Then, the Hermit could execute and derive the inferences for the small knowledge graph. In the level (3), we applied all the previous changes and reran the annotation process for all the experiments, merged the datasets, and exported as rdf/xml. There were no new consistency issues in the large and fused graph.

3.5.3 Correctness and completeness

We prepared ten competence questions (CQs) (Table 6) taking into account the merged knowledge graph with the experiment information and the results obtained by the models.

Firstly, for each CQ, we manually developed a SPARQL query[24] and evaluated the data retrieved from the graph to check if the ontology is complete and correct. We compared the query results with the expected outputs provided by the original tabular data exploration. For all CQs, the extended properties and classes were enough to build the queries and retrieve the expected result.

In the next stage, the SPARQL queries designed by humans were considered the gold standard to compare against those generated by LLM models using retrieval augmented generation. The chain of thought for the test is based on the following procedure:

1. Identification of intent (in this test only select), but could be ask, describe, or update, following the sparql graph manipulation options[25].
2. Sparql query generation to obtain data to formulate the context to answer the question
3. Parsing of gathered triples from the query execution.

Supplementary table 6 contains the queries generated for each question retrieved from the LLM models and from the human curation. Interestingly, the unique model that was able to retrieve valid results for all queries was the gemini-2.0-flash. Table 7 shows the common error types found in the SPARQL queries produced by the LLM models, summarized in four categories, and the performance of the models according to them. The llama3.2 model produced only two valid and syntactically correct queries for the questions “Get the distinct evaluation metrics used to evaluate the models” and “Get the distinct evaluation techniques used in the experiments”, and the mistral model committed syntactic errors in all 10 queries. Most of these errors are related to the usage of prefix in the condition statements that were not declared in the beginning, and the misplacement of the variables retrieved in the SELECT block of the query. The answers for the first four competence queries were correctly retrieved by the SPARQL queries produced by gemini-2.0-flash model. Three of these queries were very straightforward, and they were considered easy to reproduce the human-based queries, since their goal was just listing the entities (CQ1), statistical functions (CQ3) and evaluation metrics (CQ4). The question “Retrieve the number of models and

[24] https://github.com/YasCoMa/ner-fair-workflow/tree/master/downstream$_{e}xperiments/human_{q}ueries.txt$

[25] https://www.w3.org/TR/2013/REC-sparql11-query-20130321/

datasets by experiment" (CQ2) required aggregation and count to retrieve the number of models and datasets per experiment.

Google's model used almost the same query statements as the human-based query. As for the other 6 questions, one of them ("Retrieve the name and value of the hyperparameters used by each model" CQ6), this model produced a semantically correct query; however, it did not retrieve the model as result variable in the select block, this error led to the retrieval of fewer records (49) than the human-based query that stratified the hyperparameters together with their respective model (106 records). The queries for the remaining 5 questions retrieved zero records, besides the query is correct in terms of syntax, they returned no answers due to a confusion regarding the filters for level (token or word) and context. In the questions related to the evaluation metrics, instead of using the the filter regex in the object of underContext to filter results in test as specified in the question, it tried to directly match the "test" as the object of reportLevel, which is a semantic type of error.

4 Discussion

In this work, we presented the NERFAIR workflow that implements all the features recommended by recent studies [13] in order to document the NER experiments during the execution. The workflow generates reports per model replicate and also aggregates the evaluation metric results by statistical functions to facilitate their downstream analysis. We exhaustively tested it for the NER task concerning the PICO domain and 4 other benchmark datasets and demonstrated that we were able to increase the F1-score values reported by the previous studies using hyperparameter optimization. All processed datasets generated during these experiments can be found in a collection[26] on the Hugging Face platform. We demonstrated the usage of the significance analysis module of our workflow by comparing our mean F1-score values with the other external evaluators in the PICO dataset. This analysis showed that our results improved significantly in relation to the other ones reported for this dataset.

To support the generation of new annotations for the PICO domain, We designed and evaluated the CTPICO augmentation workflow that automatically extracts up-to-date completed clinical trials structured data and performs cross-references with PubMed abstracts to generate annotation candidates that undergo prediction in the NERFAIR workflow. In a second stage it evaluates the generated annotations strictly to the ground truth items for each entity in the clinical trials, helping to validate the predictions. The results showed that the coverage for many entities was increased, we provided the input for retraining models from the top ranked annotations. However, the remaining annotations could be refined by a human curation. So, this workflow could help to narrow down the possibilities for the human-based analysis, optimizing the annotation process.

For both workflows, we chose Nextflow [34] as the engine to orchestrate the tasks and manage the execution of the modules, due to its broad usage in bioinformatics automated pipelines and its capacity to enable reproducibility. We document the experiment assets, parameters, and configurations at every step to be as transparent

[26] https://huggingface.co/collections/yasmmin/nerfair-processed-datasets

as possible. The NERFAIR workflow was built in line with all the recommendations stated in [8, 13]. To report the results following the FAIR principles cookbook, we added an experiment metadata enrichment module and systematically evaluated our ontology terms extension through the procedure adopted in [24]. Our ontology helped to fill the gaps in the existing vocabularies to describe the specificities of a ML experiments focused on NER.

We created a knowledge graph from all the experiments executed in the analysis carried out by our NERFAIR workflow and we were able to retrieve correct and complete answers for all of them, according to human-generated SPARQL queries.

To take into account the publicly available LLM model capabilities for question answering, we evaluated three lightweight models in terms of retrieval augmented generation, using exactly the same questions, given the experiments' knowledge graph as context. The gemini-2.0-flash surpassed by far the other tested models with zero syntactic errors and retrieved the correct and complete results in 3 out of the ten questions. This may indicate that similar models trained with more tokens could help to extract insights from knowledge graphs using SPARQL as search engine to navigate along the results of the annotated NER experiments. The study [35] showed that the fine-tuning of the LLM models and increment of epochs while training improve the accuracy of the generated SPARQL query in relation to the reference. They also observed that not only was the wrong property chosen in the conditions of the where clause, but also the generation process was not taking into account the ontology axioms as the domain and range of the properties, which led to syntactically correct queries that connect the concepts wrongly. We carried out these experiments in a zero-shot configuration, but, according to the results of the fine-tuned models, these models may have more accuracy to answer SPARQL queries that efficiently retrieve the same results as expert-based ones.

5 Conclusion

Studies have demonstrated that reproducibility has been a recurrent issue in the natural language processing (NLP) area. Our study aimed to create an agnostic workflow for NER experiment execution, allowing the full documentation of research assets, model training conditions, hyperparameters, and experiment configuration. We also took into consideration a higher level of scope and packaged our method to meet the traceability, versioning, standardization, usability, and shareability categories highlighted in [8, 13].

The application study case led to the design of an augmentation workflow for the PICO domain extracting candidates and validating the predicted label annotations using clinical trials structured and curated data. This contribution allows researchers to accelerate and turn more efficient the work of human annotators by ranking the closest annotations regarding the ground truth data.

Furthermore, we also demonstrate the validity of our metadata enrichment module and the proposed ontology extension designed to describe NER experiments. We filled the gaps existing in published ontologies for machine learning and workflow domains and provided fully functional usage examples of experiment data integration and SPARQL queries traversing the knowledge graph generated from a set of experiments. Finally, we provided a detailed discussion about the performance of LLM models for SPARQL query generation from natural language questions by retrieval augmented generation. This exercise aimed to give hints about the main issues and recurrent mistakes that should be addressed when building training datasets for model fine-tuning.

References

[1] Hobbs, J.R.: Information extraction from biomedical text. J. Biomed. Inform. **35**(4), 260–264 (2002)

[2] Goyal, N., Singh, N.: Named entity recognition and relationship extraction for biomedical text: A comprehensive survey, recent advancements, and future research directions. Neurocomputing **618**(129171), 129171 (2025)

[3] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)

[4] Jung, S.J., Kim, H., Jang, K.S.: LLM based biological named entity recognition from scientific literature. In: 2024 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 433–435. IEEE, ??? (2024)

[5] Monajatipoor, M., Yang, J., Stremmel, J., Emami, M., Mohaghegh, F., Rouhsedaghat, M., Chang, K.-W.: LLMs in biomedicine: A study on clinical named entity recognition. arXiv [cs.CL] (2024) [cs.CL]

[6] Cuevas Villarmin, C., Cohen-Boulakia, S., Naderi, N.: Reproducibility in named entity recognition: A case study analysis. In: 2024 IEEE 20th International Conference on e-Science (e-Science), vol. 17, pp. 1–10. IEEE, ??? (2024)

[7] 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: Proceedings of the First Workshop on Information Extraction from Scientific Publications, pp. 26–31. Association for Computational Linguistics, Stroudsburg, PA, USA (2022)

[8] Digan, W., Névéol, A., Neuraz, A., Wack, M., Baudoin, D., Burgun, A., Rance, B.: Can reproducibility be improved in clinical natural language processing? a study of 7 clinical NLP suites. J. Am. Med. Inform. Assoc. **28**(3), 504–515 (2021)

[9] Lamprecht, A.-L., Garcia, L., Kuzak, M., Martinez, C., Arcila, R., Martin Del Pico, E., Dominguez Del Angel, V., Sandt, S., Ison, J., Martinez, P.A., McQuilton, P., Valencia, A., Harrow, J., Psomopoulos, F., Gelpi, J.L., Chue Hong, N., Goble, C., Capella-Gutierrez, S.: Towards FAIR principles for research software. Data Sci. **3**(1), 37–59 (2020)

[10] Xhani, D., Rebelo Moreira, J.L., Sinderen, M., Ferreira Pires, L.: XMLPO: An ontology for explainable machine learning pipeline. In: Frontiers in Artificial Intelligence and Applications. Frontiers in artificial intelligence and applications, pp. 193–206. IOS Press, ??? (2024)

[11] Zheng, J., Harris, M.R., Masci, A.M., Lin, Y., Hero, A., Smith, B., He, Y.: The ontology of biological and clinical statistics (OBCS) for standardized and reproducible statistical analysis. J. Biomed. Semantics **7**(1), 53 (2016)

[12] Ison, J., Kalas, M., Jonassen, I., Bolser, D., Uludag, M., McWilliam, H., Malone, J., Lopez, R., Pettifer, S., Rice, P.: EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats. Bioinformatics **29**(10), 1325–1332 (2013)

[13] Cuevas Villarmin, C., Cohen-Boulakia, S., Naderi, N.: Reproducibility in named entity recognition: A case study analysis. In: 2024 IEEE 20th International Conference on e-Science (e-Science), vol. 17, pp. 1–10. IEEE, ??? (2024)

[14] Hu, Y., Keloth, V.K., Raja, K., Chen, Y., Xu, H.: Towards precise PICO extraction from abstracts of randomized controlled trials using a section-specific learning approach. Bioinformatics (2023)

[15] 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. J. Am. Med. Inform. Assoc. **31**(5), 1163–1171 (2024)

[16] Chen, F., Zhang, G., Fang, Y., Peng, Y., Weng, C.: Semi-supervised learning from small annotated data and large unlabeled data for fine-grained participants, intervention, comparison, and outcomes entity recognition. J. Am. Med. Inform. Assoc. **32**(3), 555–565 (2025)

[17] Jain, S.M.: Hugging face. In: Introduction to Transformers for NLP, pp. 51–67. Apress, Berkeley, CA (2022)

[18] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19, pp. 2623–2631. Association for Computing Machinery, New York, NY, USA (2019)

[19] Jurman, G., Riccadonna, S., Furlanello, C.: A comparison of MCC and CEN error measures in multi-class prediction. PLoS One **7**(8), 41882 (2012)

[20] Vieira, S.M., Kaymak, U., Sousa, J.M.C.: Cohen's kappa coefficient as a performance measure for feature selection. In: International Conference on Fuzzy Systems, pp. 1–8. IEEE, ??? (2010)

[21] Wang, K., Stevens, R., Alachram, H., Li, Y., Soldatova, L., King, R., Ananiadou, S., Schoene, A.M., Li, M., Christopoulou, F., Ambite, J.L., Matthew, J., Garg, S., Hermjakob, U., Marcu, D., Sheng, E., Beißbarth, T., Wingender, E., Galstyan, A., Gao, X., Chambers, B., Pan, W., Khomtchouk, B.B., Evans, J.A., Rzhetsky, A.: NERO: a biomedical named-entity (recognition) ontology with a large, annotated corpus reveals meaningful associations through text embedding. NPJ Syst. Biol. Appl. **7**(1), 38 (2021)

[22] Rainio, O., Teuho, J., Klén, R.: Evaluation metrics and statistical tests for machine learning. Sci. Rep. **14**(1), 6086 (2024)

[23] Lahitani, A.R., Permanasari, A.E., Setiawan, N.A.: Cosine similarity to determine similarity measure: Study case in online essay assessment. In: 2016 4th International Conference on Cyber and IT Service Management. IEEE, ??? (2016)

[24] Faria, D., Eugénio, P., Contreiras Silva, M., Balbi, L., Bedran, G., Kallor, A.A., Nunes, S., Palkowski, A., Waleron, M., Alfaro, J.A., Pesquita, C.: The ImmunoPeptidomics ontology (ImPO). Database (Oxford) **2024**, 014 (2024)

[25] Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: OOPS! (OntOlogy pitfall scanner!). Int. J. Semant. Web Inf. Syst. **10**(2), 7–34 (2014)

[26] Sanjib, N., Bihung, B., Haradip, M., Mahananda, B., Bidisha, S., Sukumar, N.: Comparative study of zero-shot cross-lingual transfer for bodo POS and NER tagging using gemini 2.0 flash thinking experimental model. arXiv [cs.CL] (2025) [cs.CL]

[27] Mondillo, G., Colosimo, S., Perrotta, A., Frattolillo, V., Masino, M., Martino, M., Miraglia Del Giudice, E., Marzuillo, P.: Artificial intelligence for solving pediatric clinical cases: A retrieval-augmented approach utilizing Llama3.2 and structured references. Int. J. Med. Inform. **203**(106027), 106027 (2025)

[28] Li, J., Sun, Y., Johnson, R.J., Sciaky, D., Wei, C.-H., Leaman, R., Davis, A.P., Mattingly, C.J., Wiegers, T.C., Lu, Z.: BioCreative V CDR task corpus: a resource for chemical disease relation extraction. Database (Oxford) **2016**, 068 (2016)

[29] Kury, F., Butler, A., Yuan, C., Fu, L.-H., Sun, Y., Liu, H., Sim, I., Carini, S., Weng, C.: Chia, a large annotated corpus of clinical trial eligibility criteria. Sci. Data **7**(1), 281 (2020)

[30] Doğan, R.I., Leaman, R., Lu, Z.: NCBI disease corpus: a resource for disease name recognition and concept normalization. J. Biomed. Inform. **47**, 1–10 (2014)

[31] Luo, L., Lai, P.-T., Wei, C.-H., Arighi, C.N., Lu, Z.: BioRED: a rich biomedical relation extraction dataset. Brief. Bioinform. **23**(5), 282 (2022)

[32] Abdul, W.M., Pimentel, M.A.F., Salman, M.U., Raha, T., Christophe, C., Kanithi, P.K., Hayat, N., Rajan, R., Khan, S.: Named clinical entity recognition benchmark. arXiv [cs.CL] (2024) [cs.CL]

[33] Hasnain, A., Rebholz-Schuhmann, D.: Assessing FAIR data principles against the 5-star open data principles. In: Lecture Notes in Computer Science. Lecture notes in computer science, pp. 469–477. Springer, Cham (2018)

[34] Di Tommaso, P., Chatzou, M., Floden, E.W., Barja, P.P., Palumbo, E., Notredame, C.: Nextflow enables reproducible computational workflows. Nat. Biotechnol. **35**(4), 316–319 (2017)

[35] Pan, X., Boer, V., Ossenbruggen, J.: FIRESPARQL: A LLM-based framework for SPARQL query generation over scholarly knowledge graphs. arXiv [cs.AI] (2025) [cs.AI]

Table 4 Comparison of the number of papers and annotations identified for each entity in each step of the validation procedure. The counts are organized according to (1) all initial predictions and all the items for the same entity in the respective CT data; (2) the values after reducing the pairwise similarity grouping by the hit in CT data that had the maximum value of String Similarity (SS); and (3) the top-ranked matchings between the predicted annotations and the CT item hits.

Entity / Dataset		**Papers**			**Annotations**	
	All predicted	**Grouped by max SS**	**Top ranked**	**All predicted**	**Grouped by max SS**	**Top ranked**
age	27170	27170	15146	73350	48625	19584
condition	47590	47590	25293	211064	123339	43499
intervention	76640	76640	51840	850359	317842	116429
outcome	75520	75520	64907	6446762	717696	288816
control	9405	9405	5439	24403	23469	8368
outcome-Measure	15908	15908	13577	972451	80415	48347
eligibility	71257	71257	63504	3377276	274406	176567
location	26040	26040	18469	1539615	59736	32380
iv-cont-mean	4483	4483	2330	243852	20979	6127
iv-bin-abs	3086	3086	930	56016	9334	1678
ethinicity	1510	1510	378	9195	3167	625
iv-cont-median	2097	2097	792	32731	6185	1363
iv-bin-percent	887	887	251	10412	3375	433

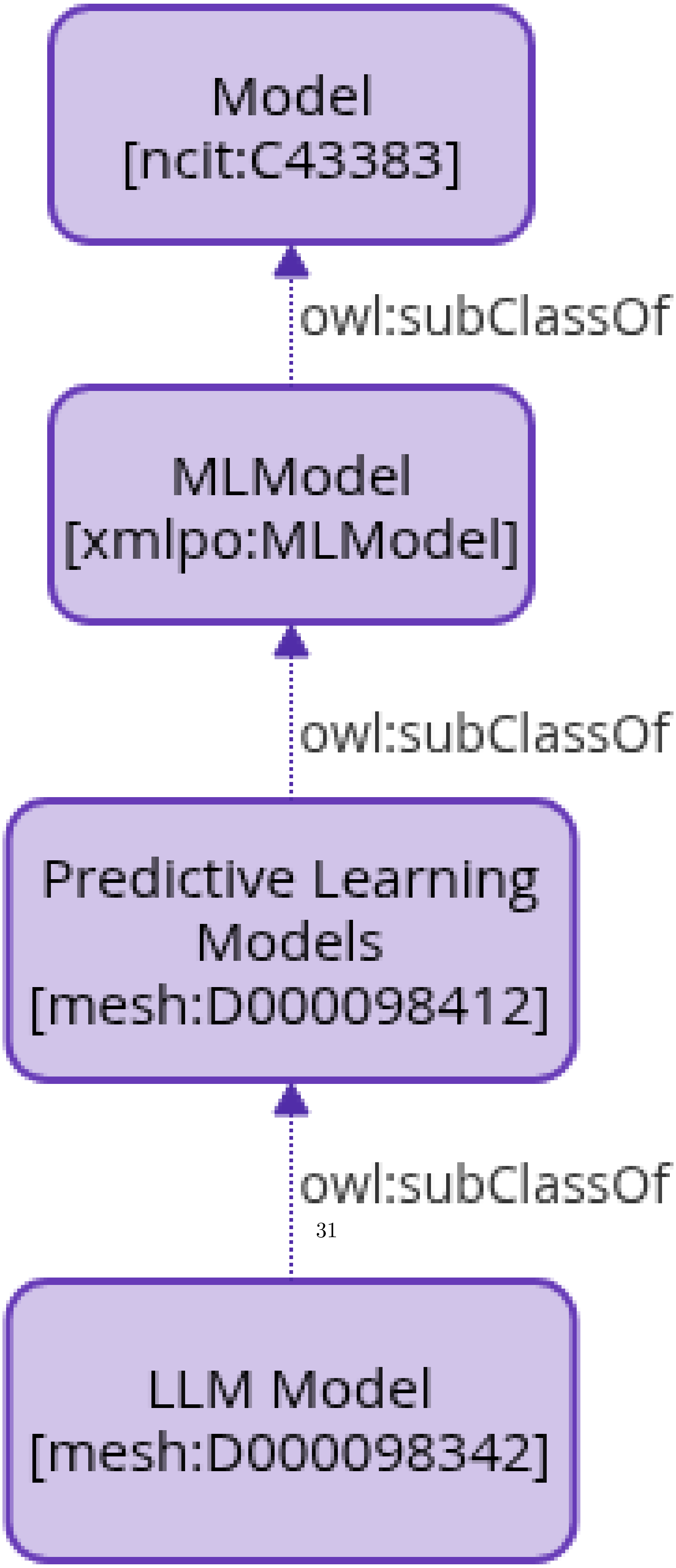


Fig. 9 Hierarchy of the model machine learning models connecting the concepts spread across the mesh, xmlpo and ncit ontologies.

Table 5 Descriptive statistics of the ontology extension for NLP experiments.

Item	Count
Axioms	64
Classes	62
Object properties	33
Data properties	12
Individuals	51,215
Annotation Properties	24
Parent declarations (subClassOf)	15
Disjoint classes	2
Inverse object properties	3
Functional object properties	2
Transitive object properties	1

Table 6 Competence questions used for the correctness and completeness evaluation of the ontology extension and also for the comparison between human-based and LLM-generated SPARQL queries.

Identifier	Question
CQ1	Get the distinct names of named entities used in each experiment
CQ2	Retrieve the number of models and datasets by experiment
CQ3	Get the distinct evaluation metrics used to evaluate the models
CQ4	Get the distinct statistical functions used to aggregate the evaluation metrics
CQ5	Get the distinct evaluation techniques used in the experiments
CQ6	Retrieve the name and value of the hyperparameters used by each model
CQ7	What is the number of features and instances of the largest dataset?
CQ8	Which evaluation technique is associated with the highest mcc values?
CQ9	For each experiment, retrieve the evaluation technique and the level that obtained the highes
CQ10	For each level, technique, and entity, retrieve the f1-score values aggregated by max per mode

Table 7 Summary of the CQ counts in which the listed error types were committed by each LLM model in the retrieval augmented generation experiment.

Error types / Models	llama3.2	mistral	gemini-2.0-flash
Prefix declaration	2	3	0
Variable selection	5	1	1
Missing select variables	10	10	7
Missing condition statements	10	10	6