Reproducibility in Named Entity Recognition: A Case Study Analysis

Carlos Cuevas Villarmin, Sarah Cohen-Boulakia, Nona Naderi

Reproducibility in Named Entity Recognition: A Case Study Analysis

Carlos Cuevas Villarmin
Université Paris-Saclay, CNRS
Laboratoire Interdisciplinaire
des Sciences du Numérique
Orsay 91405, France
Email: carlos.cuevas-villarmin
@universite-paris-saclay.fr

Sarah Cohen-Boulakia
Université Paris-Saclay, CNRS
Laboratoire Interdisciplinaire
des Sciences du Numérique
Orsay 91405, France
Email: sarah.cohen-boulakia
@universite-paris-saclay.fr

Nona Naderi
Université Paris-Saclay, CNRS
Laboratoire Interdisciplinaire
des Sciences du Numérique
Orsay 91405, France
Email: nona.naderi
@universite-paris-saclay.fr

***Abstract*—Information extraction from text is essential in data science and artificial intelligence, especially with the increase in the number of scientific articles. Robust methods are needed to structure texts and highlight key information. Named Entity Recognition (NER) identifies and classifies major elements in text, aiding in dataset structuring and tagging.**

Under the FAIRClinical and ShareFAIR projects, we aimed to extract information from clinical trial publications using NER models. These models have been successfully used for clinical trial information extraction. In this paper, we report on the difficulties met in reusing existing NER solutions, focusing on mandatory replicability. This paper discusses the challenges faced in replicating a notable study, the difficulties encountered and lessons learned. It compares our experiment with the feedback provided by the literature and draws conclusions.

I. Introduction

Information extraction from text represents a key field in data science and artificial intelligence. With the significant increase in the number of available scientific articles, it is particularly important to develop robust information extraction methods allowing to structure published texts and highlight major information. More specifically, Named Entity Recognition (NER) is a process in natural language processing that identifies and classifies key elements within text into predefined categories (entities). NER has been successfully applied in the last years to biological texts, to detect the presence of papers describing information about disease, genes and proteins and possibly relationships between such entities [1], [2].

The present work has been performed in the context of FAIRClinical project (funded by CHIST-ERA) whose aim is to extract key information from text and more particularly from publications on clinical trials and clinical case reports. We naturally rely on NER models to extract such information. Boosted by the information extraction challenges posed by the Covid-19 crisis and the development of deep-learning based models, NER models have been used to extract information on clinical trials [3]–[5].

Following a cumulative science approach, we aimed to reuse existing NER solutions that had shown good results on similar datasets, with the intention of adapting them to new needs later. But before re-purposing an existing solution to new needs (new corpora), one may make sure that such a solution can be reproduced in a very similar setting (same corpus, same models with the same hyperparameters, . . .). This re-execution step allows to make sure that the original experiment is fully understood.

The contributions of this paper are: 1) we discuss the challenges we encountered while attempting to reproduce the NER results of a paper [6] of strong interest for our ongoing projects especially FAIRClinical and to some extent ShareFAIR (Section II and Section III). 2) We propose a set of lessons learned (Section IV), 3) compare them with the literature (Section V), and draw conclusions (Section VI).

II. Use-case study

In this section, we introduce the practical case that we faced when reproducing the performance of the models and findings reported in [6], which served as motivation for the development of this work. In II-A, we describe the reference corpus and models, along with the reasons for its selection and why we needed to reproduce the results. In II-B, we outline the reproducibility process describing the problems encountered and provide all the materials to redo our experiments.

A. Target Work: NER on Datasets with PICO Entities

There are several datasets frequently used in the literature on named entity recognition in medical domain texts, particularly using the well-known **PICO framework** [7]–[9]. This framework identifies the words in a given medical text that belong to the four components: *Participants/Problem (P), Intervention (I), Comparison/Control (C), and Outcome (O)*, which allows the formulation of a well-defined focused clinical question. While some datasets focus on these four top-level entities, a more precise labeling involves creating detailed labels for each starting span. For example, the EBM-NLP corpus [7] allows differentiation within *participants*, such as age and gender, or in the top-level *outcome* entity, pain, mortality, adverse effects, etc. can be identified. In this corpus, however, numeric texts that identify the number of participants who had certain outcomes are not annotated.

A novel publicly available dataset [6] addresses this lack of precision in randomized controlled trials, concretely in relation to numeric texts annotations. The corpus consists of 1,011 abstracts of breast cancer randomized controlled trials extracted from PubMed, the main searchable bibliographic database supporting scientific and medical research. The abstracts were annotated by two annotators and the inter-annotator agreement was calculated based on *Cohen Kappa* [10] which reached a score of 0.72. The PICO breast cancer dataset contains a total of **26 entities**, compared to the usual 4 found in PICO corpora. As indicated by its authors, a more precise annotation is provided by creating sub-entities rather than limiting to the four top-level entities. Especially within the outcome entity, the labeling is very detailed, with different classes created to differentiate outcomes of the intervention and control groups based on whether they are absolute values (e.g., the number of patients who had certain outcomes), percentages, average values, medians, quantiles, or standard deviations. Table I presents all the entities and sub-entities considered in this corpus. As far as we know, it is the only publicly available dataset that provides data with this level of precision in labeling PICO entities.

To demonstrate the quality of the dataset, the authors of [6] trained two models: BioBERT [1] and LongFormer [11], and **justified the quality of the dataset** by analyzing the performance of the models, **arguing** that they achieved **F1 scores greater than 0.80 for most entities**.

To be able to build on their models, we will first reproduce the performance of the models on the reference corpus. It is well-known that having a high number of classes in a classification problem makes the task more difficult because the model must learn more complex decision boundaries to distinguish each category from others. This leads to data sparsity, as each class has fewer examples, making it harder for the model to generalize well. Additionally, the computational load increases due to the larger output space, and the probability of misclassification rises, resulting in a lower overall accuracy.

Taking into account the aforementioned disadvantage in terms of learning process and the recent publication of the dataset, especially after identifying the lack of information in [6] during the training process of the models, it is necessary to **confirm the quality of the dataset**. To do this verification, we have to reproduce the given baselines to demonstrate the same hypothesis. This way, we **obtain a greater understanding of the model learning process and gain reliability of the results** to make up for the lack of information provided.

If this hypothesis is confirmed, that is, if the published results can be reproduced by a third party, we will be able to use models trained with this level of accuracy in future entity recognition tasks in the context of FAIR-Clinical project. This would represent a significant advancement in the medical field, enabling a transparent and reliable process for higher precision in NER. As a result, more precise and understandable conclusions could be automatically extracted, allowing for quicker insights, such as drawing conclusions about the effects of a treatment in clinical trials.

B. Reproducibility task

The publicly available dataset[1] consists of two files for each abstract: a `.txt` file containing the plain text and a `.ann` file reporting the annotations made with the brat rapid annotation tool[2]. The manuscript, however, does not provide a link to any of the code used in the study. This has complicated the reproduction process, which was further accentuated by the lack of information in some parts of the procedure.

In the attempt to reproduce the baselines, we divide the process in 4 categories: (1) Data preprocessing, (2) experimental set-up, (3) training process, and (4) model evaluation.

1) Data preprocessing: Given the way the data is presented, as previously explained, the first step is to **transform the data into a format compatible with what is expected by the masked language models (MLMs)** on the HuggingFace platform[3]. This platform provides members of the machine learning community ways to exchange models, datasets, and applications. HuggingFace has developed under an emphasis on accessibility and community-driven development. As of July 2024, it gathers 800,407 models.

During this first step, we noticed lack of information in the process, preventing us from directly reproducing the baselines. We thus decided to use a script from the creators of BRAT[4] to transform our data into the CoNLL format, which is frequently used in NER tasks. CoNLL format is a text file with one word per line (together with its assigned label) with sentences separated by an empty line. The words and their assigned labels are separated by a tab character. It is worth noticing that the code we used here is not the only option available for performing the same task. We could have defined our own tool. However, we preferred to use a standard shared open-source code to ensure the quality of the results, allowing for better understanding and confidence in the outcomes.

Without knowing which tool (a third-party or in-house code) was used for transforming the data into the models' input format, nor guidelines to do it manually, nor direct access to the code used, an exact reproduction of the dataset with which we will train the models cannot be guaranteed. In particular, different tools can use different tagging schemes: differentiating only the words that are "Inside" of an entity from those that are "Outside" (IO format), distinguishing the word that "Begins" the entity chunk (IOB format), or even identifying the word that "Ends" the set of words of an entity and the chunks that are only a "Single" word (BIOES format). Additionally, overlapping entities can be handled differently, for example, by removing one of them.

Consequently, different configurations could result in the generation of differently preprocessed datasets, leading to the same model being trained on two not exactly identical datasets.

Nonetheless, despite not using exactly the same procedure, the outputs could be the same. Considering the statistical

[1] https://github.com/sociocom/PICO-Corpus
[2] http://brat.nlplab.org
[3] https://huggingface.co/
[4] https://github.com/nlplab/brat/blob/master/tools/anntoconll.py

information provided by the authors about the corpus, the only way to **compare the preprocessed data** is to determine **if at least the absolute frequency of each entity class is accurately reproduced** after transforming the data from CoNLL format to IOB format. The frequency obtained in our study in comparison with the distribution reported in the reference paper [6] can be seen in Table I (the difference between our values and the reference values are in parentheses). It can be seen that the results are not the same in 3 entities: total-participants, outcome and outcome-Measure. Although the difference is not very high, we must keep in mind that from now on we will not be working with exactly the same dataset. Therefore, the results we obtain are already subject to this difference.

Entity	Count (Diff with [6])	nº files
Participants		
total-participants	1093 (-1)	847
intervention-participants	887	674
control-participants	784	647
age	231	210
eligibility	925	864
ethinicity	101	83
condition	327	321
location	186	168
Intervention & Control		
intervention	1067	1011
control	979	949
Outcomes		
outcome	5038 (-15)	978
outcome-Measure	1077 (-4)	413
iv-bin-abs	556	288
cv-bin-abs	465	258
iv-bin-percent	1376	561
cv-bin-percent	1148	520
iv-cont-mean	366	154
cv-cont-mean	327	154
iv-cont-median	270	140
cv-cont-median	247	133
iv-cont-sd	129	69
cv-cont-sd	124	67
iv-cont-q1	4	3
cv-cont-q1	4	3
iv-cont-q3	4	3
cv-cont-q3	4	3

TABLE I
CORPUS STATISTICS: THE FREQUENCY OF EACH ENTITY/SUB-CATEGORY, THE DIFFERENCE WITH RESPECT TO [6] IF APPLICABLE (COUNT COLUMN), AND THE NUMBER OF ABSTRACTS IN WHICH EACH ENTITY IS FOUND AFTER TRANSFORMATION (Nº FILES COLUMN). THE ABBREVIATIONS "IV" AND "CV" REFER TO INTERVENTION AND CONTROL GROUP RESPECTIVELY, "BIN" IS USED FOR BINARY OUTCOMES AND "CONT" REFERS TO CONTINUOUS ONES. MORE DETAILS ABOUT EACH OF THE ENTITIES IN [6].

2) Experimental set-up: Once the data was prepared for training the models, we **encountered a significant information gap**. In order to reproduce the baselines, we would need a substantial amount of information to conduct the training under exactly the same conditions, such as the **same data split, method for calculating evaluation metrics, versions of libraries used, or hardware employed**, among others. Here, the only information provided is that the abstracts were randomly split into 80% for training data and 20% for test data and that training was conducted with five different seeds, averaging the F1-score results. No information was provided on how the metrics were calculated exactly or whether an external tool or a custom script was used. We cannot replicate the exact split performed, so we will most likely train the models with different samples in each set. Additionally, the versions of the pre-trained transformer-based models used are not specified, nor whether/which hyperparameters were optimized.

Therefore, at this stage, having to make many decisions to define our set-up, we concluded that reproducing the evaluation metrics scores to determine the performance of the models without a statistically significant difference was challenging because many factors came into play. However, at this stage it remains possible to reproduce the findings and conclusions.

To address the information gaps and to reproduce the baseline, we conducted two experiments: one maintaining the same proportions of data as that of the corpus publication for the training set (80%) and the test set (20%), referred to as *Experiment 1*, and another where we introduced a validation set to perform hyperparameter optimization by considering half of the test set as a validation set, referred to as *Experiment 2*.

We made our dataset splits publicly available on HuggingFace[5]. In both experiments, we have used the same models as the ones used in the presentation of the dataset: BioBERT and LongFormer. Concretely, as there exist different versions of the pre-trained models publicly available, we used two models from HuggingFace[6]. Following [6], we trained 5 models in each experiment. To ensure F1-scores reproducibility, we controlled the different seeds used. The number of epochs (an epoch is an hyperparameter that refers to one complete cycle through the entire training dataset) was arbitrarily set to 40 and the configuration saved the weights of the epoch where the micro F1-score achieved its maximum (on the test set in Experiment 1 and on the validation set in Experiment 2). In terms of hardware specifications, we used a single NVIDIA RTX A6000 with 48GiB of GDDR6 GPU to fine-tune both NER models in both experiments.

To compute the evaluation metrics, concretely F1-score, since there was no information about it in the publication, we used the `seqeval` library (version 1.2.2) [12], which appears in the HuggingFace documentation and has been mainly used to calculate evaluation metrics and to analyze the performance of models trained for the NER task on the PICO corpora [13], [14]. We chose the default mode which makes the **reported metrics compatible with the evaluation done in CoNLL shared tasks** of the early 2000 [15]–[17]. These events established the standard of considering a true positive prediction typically if there is an exact match in both the span (the tokens or characters in a chunk) and the type assigned to the chunk (e.g., person) in named entity recognition (NER) and other chunk extraction tasks [18].

[5]https://huggingface.co/datasets/cuevascarlos/PICO-breast-cancer

[6]https://huggingface.co/dmis-lab/biobert-base-cased-v1.2
https://huggingface.co/allenai/longformer-base-4096

However, this is not the only way to calculate the metrics. Instead of calculating true positives at the entity-level, as explained in the standard previously described, true positives can be calculated at the token-level. Each token is considered independently and compared only to its target value, regardless of the other tokens belonging to the same label. Therefore, in this case, a true positive prediction is considered if there is an exact match in the type assigned to the token. This alternative, being more permissive in the computation of true positives, is known as the *relaxed* or *lenient* approach [19].

3) Training process.: For **hyperparameter optimization** in Experiment 2, we used the Optuna Framework [20]. This tool allowed us to fine-tuned four hyperparameters: `learning_rate`, `weight_decay`, `per_device_train_batch_size`, `per_device_eval_batch_size`. It is a stochastic approach where 15 trials are done by combining randomly different values of the hyperparameters. In each attempt, we restored the weights from the epoch where the loss function on the validation set was the lowest and then selected the combination of hyperparameters that reported the highest micro F1-score. The `learning_rate` and `weight_decay` were the same for both models, while the `batch_size` possible values were $8, 16,$ or 32 for BioBERT model and 8 or 16 for LongFormer due to computational capacity. Precisions about the hyperparameter optimization process and the training step in general can be found in our code[7]. The obtained hyperparameters for each model in Experiment 2 are summarized in Table II. In contrast, in Experiment 1, the hyperparameters were set to: `learning_rate=2e-5`, `weight_decay=0.01`, `per_device_train_batch_size=16` and `per_device_eval_batch_size=16`. We used the default values of TrainingArguments[8] from transformers library for the hyperparameters that have not been explicitly mentioned before in this paper.

4) Model evaluation: The F1-scores obtained on the test set in both Experiments with BioBERT model are summarized in Table III and the results of LongFormer model are in our repository. The performance of BioBERT and LongFormer obtained in our experiments seem to be considerably worse in comparison with those reported in the reference paper [6]. The per-class F1-scores of all the entities were worse than the baseline, even in Experiment 2 where the model performances surpass the performance of models in Experiment 1 after doing a more exhaustive hyperparameter optimization. In general terms, looking at the macro F1-score in Experiment 2, we get similar values to the original paper but it is caused by the fact that the entities iv-cont-q1, cv-cont-q1, iv-cont-q3, cv-cont-q3 do not have representation in our test set and therefore the F1-score per class was not computed and the macro F1 is computed without these four entities. This fact does not entail any loss of information given the low frequency of the entities in question. However, that is why we will focus exclusively on the weighted F1-score to compare the results. Here, we see a difference up to 10 percentage points.

[7]https://github.com/cuevascarlos/ClinicalTrials/blob/main/TrainingNER.py

[8]https://huggingface.co/docs/transformers/v4.41.0/en/main_classes/trainer

Hyperparameter	Model	
	BioBERT	LongFormer
Learning rate	$4.976e-05$	$4.471e-05$
Weight decay	0.003	0.007
per device train batch size	8	8
per device eval batch size	16	16

TABLE II
HYPERPARAMETERS OBTAINED DURING HYPERPARAMETER STOCHASTIC SEARCH USING THE OPTUNA FRAMEWORK IN EXPERIMENT 2. THE VALUES HAVE BEEN ROUNDED IN THE TABLE, THE EXACT HYPERPARAMETERS ARE IN THE GITHUB REPOSITORY.

Entity	[6]	Experiment 1	Experiment 2
Participants			
total-participants	0.94	0.9065 (+-0.0096)	0.9313 (+-0.0048)
intervention-participants	0.85	0.7431 (+-0.0123)	0.8177 (+-0.0135)
control-participants	0.88	0.7846 (+-0.0108)	0.8480 (+-0.0124)
age	0.80	0.5638 (+-0.0300)	0.5724 (+-0.0731)
eligibility	0.74	0.6049 (+-0.0131)	0.6382 (+-0.0219)
ethinicity	0.88	0.7135 (+-0.0433)	0.7163 (+-0.0353)
condition	0.80	0.6412 (+-0.0469)	0.7122 (+-0.0421)
location	0.76	0.6156 (+-0.0226)	0.6258 (+-0.0363)
Intervention & Control			
intervention	0.84	0.7805 (+-0.0047)	0.7899 (+-0.0095)
control	0.76	0.6780 (+-0.0205)	0.6529 (+-0.0190)
Outcomes			
outcome	0.81	0.6321 (+-0.0056)	0.6667 (+-0.0151)
outcome-Measure	0.84	0.7441 (+-0.0274)	0.8003 (+-0.0240)
iv-bin-abs	0.80	0.6184 (+-0.0278)	0.7640 (+-0.0352)
cv-bin-abs	0.82	0.6557 (+-0.0214)	0.8195 (+-0.0219)
iv-bin-percent	0.87	0.6460 (+-0.0174)	0.6731 (+-0.0317)
cv-bin-percent	0.88	0.6919 (+-0.0224)	0.7549 (+-0.0233)
iv-cont-mean	0.81	0.5081 (+-0.0352)	0.4271 (+-0.0334)
cv-cont-mean	0.86	0.4711 (+-0.0160)	0.4117 (+-0.0297)
iv-cont-median	0.75	0.6630 (+-0.0336)	0.7415 (+-0.0216)
cv-cont-median	0.79	0.6937 (+-0.0195)	0.7769 (+-0.0373)
iv-cont-sd	0.83	0.4606 (+-0.0424)	0.6274 (+-0.0683)
cv-cont-sd	0.82	0.4711 (+-0.0514)	0.7264 (+-0.0826)
iv-cont-q1	0	0.0000 (+-0.0000)	*
cv-cont-q1	0	0.0000 (+-0.0000)	*
iv-cont-q3	0	0.0000 (+-0.0000)	*
cv-cont-q3	0	0.0000 (+-0.0000)	*
micro avg	-	0.6845 (+-0.0032)	0.7261 (+-0.0119)
macro avg	0.6973	0.5495 (+-0.0022)	0.7043 (+-0.0138)
weighted avg	0.8282	0.6872 (+-0.0031)	0.7273 (+-0.0118)

TABLE III
F1-SCORE COMPARISON BETWEEN THE VALUES REPORTED IN [6] AND THE RESULTS OBTAINED IN EXPERIMENTS 1 AND 2 USING BIOBERT MODEL. THE MACRO AND WEIGHTED AVERAGE F1-SCORES OF [6] HAVE BEEN REIMPLEMENTED AND COMPUTED. ALL THE VALUES ARE THE AVERAGE OF F1-SCORES OBTAINED FROM THE 5 TRAINING MODELS. IN THE CASE OF EXPERIMENT 1 AND EXPERIMENT 2 ,THE STANDARD DEVIATION IS SHOWN IN PARENTHESES. *PER-CLASS F1-SCORE IS NOT COMPUTED IN THE LAST FOUR ENTITIES IN EXPERIMENT 2 BECAUSE NO SAMPLES OF THESE ENTITIES EXISTED IN TEST SET.

Given the multitude of decisions we had to make independently, it is difficult to pinpoint the exact cause of the difference in the F1-scores. However, we can observe that the quality of the dataset is not as clear-cut as it seemed. Identifying the main differences that have caused this discrepancy in the results is crucial to conclusively establish the validity of both outcomes and gain reliability in result quality, which is particularly important in the biomedical field.

III. Analysis of the results

After the attempt to reproduce the results of [6], in this section, we provide an analysis of the outcomes to enhance the explainability of the difference between our results and those presented in [6]. In III-A, we present some evidence we found that allowed us to deduce some of the missing information, thereby increasing the credibility and reliability of our results despite not matching those previously published. We conclude this section in III-B, discussing briefly the reproducibility of our results.

A. Discovering missing information to increase explainability

Following the statement "As a community, we need to know where our approaches fail, as much –if not more– as where they succeed" by [21], we aimed to identify what we might have done wrong or what decisions might have caused each of the differences. Without questioning the previously published results, we needed to confirm that ours were equally valid. As we have observed, considering the outcomes provided in [6], we have only two options for **tracking** the **reproducibility process** of the outcomes: **corpus statistics** and **evaluation metrics**.

Firstly, we focus on identifying the differences seen in Table I, and their cause. As a reminder, in this point we aimed to reproduce the number of occurrences of each entity. After a deep analysis, we can conclude that in the case of total-participants, there was a file (*15023242.ann*) where the same word was tagged twice with the same label but with different boundaries and in [6] both are considered. The tool we used, removes overlapping entities, so we modified a function of the BRAT tool to remove one of these annotations (the modified script is available in our repository). In the case of outcome and outcome-Measure, we concluded that NLP-lab BRAT tool concatenates consecutive chunks of words with the same entity name, while the reference paper considers the sets as two different entities.

Secondly, we examine the evaluation metric scores. Particularly for NER tasks, there are different possible approaches for their calculation, as we have explained before, for example, the strict CoNLL standard or the relaxed approach. Given the high values reported by the reference paper of the dataset, it appears that the methodology for calculating the metrics has been more lenient than ours and that token-by-token classification has been considered instead of chunk-by-chunk that is mainly used for NER tasks. To compute the metrics in a less restrictive manner on the test set, we used the classification report of `sklearn` library [22]. To do that, we have removed the target labels that correspond to the *Outside* entity and we have transformed the data into the IO format (by removing the prefix in the prediction and the target labels). As the experiment 2 results outperform those of Experiment 1, we will only provide the outputs of experiment 2. Table IV summarizes the lenient F1-scores in BioBERT models and the lenient F1-scores with LongFormer models are available in our repository. We can observe that the scores are more similar, although there are still large differences in some entities. Many other factors could have contributed to the less promising outcomes obtained in these classes but we cannot determine with certainty which of them have been influential given the information we have. Nevertheless, we can conclude that it indeed appears that a more relaxed approach was taken in computing the metrics, which considerably increases the achieved values and this decision should have been reported.

Entity	Experiment 2	Mean - [6]
Participants		
total-participants	0.9601 (+-0.0030)	0.0201
intervention-participants	0.8552 (+-0.0066)	0.0052
control-participants	0.8576 (+-0.0093)	-0.0224
age	0.9310 (+-0.0141)	0.1310
eligibility	0.8850 (+-0.0227)	0.1450
ethinicity	0.6321 (+-0.0308)	-0.2479
condition	0.8744 (+-0.0350)	0.0744
location	0.8774 (+-0.0094)	0.1174
Intervention & Control		
intervention	0.8433 (+-0.0104)	0.0033
control	0.7974 (+-0.0165)	0.0374
Outcome		
outcome	0.8775 (+-0.0121)	0.0675
outcome-Measure	0.9643 (+-0.0146)	0.1243
iv-bin-abs	0.8610 (+-0.0504)	0.0610
cv-bin-abs	0.8741 (+-0.0336)	0.0541
iv-bin-percent	0.8326 (+-0.0264)	-0.0374
cv-bin-percent	0.8444 (+-0.0195)	-0.0356
iv-cont-mean	0.6552 (+-0.0353)	-0.1548
cv-cont-mean	0.5666 (+-0.016)	-0.2934
iv-cont-median	0.8241 (+-0.0171)	0.0741
cv-cont-median	0.8325 (+-0.0182)	0.0425
iv-cont-sd	0.8364 (+-0.0717)	0.0064
cv-cont-sd	0.8517 (+-0.0439)	0.0317
iv-cont-q1	*	*
cv-cont-q1	*	*
iv-cont-q3	*	*
cv-cont-q3	*	*
accuracy	0.7997 (+-0.0125)	†
macro avg	0.7971 (+-0.0106)	0.0998
weighted avg	0.8712 (+-0.0078)	0.0430

TABLE IV
MEAN AND STANDARD DEVIATION F1-SCORE (LENIENT MODE) IN BIOBERT MODELS IN EXPERIMENT 2 AND THE DIFFERENCE BETWEEN THE MEAN ACHIEVED IN OUR SCORES AND THE BASELINE [6]. *PER-CLASS F1-SCORE IS NOT COMPUTED FOR THE LAST FOUR ENTITIES BECAUSE THERE ARE NO MENTIONS OF THEM APPEARED IN THE TEST SET. †ACCURACY IS NOT REPORTED IN [6]

To sum up, we gain reliability of our findings, despite differing from those reported in [6]. The reference values themselves are not wrong; rather, they were obtained using a different methodology that has not been explained in sufficient detail to be reproduced. This lack of detailed explanation in the original study means that even though our results differ, they are still valid within the context of our clearly defined methods and parameters. By meticulously documenting our processes and decisions, we provide a transparent and reproducible framework for others to follow. This transparency is crucial because it allows for a better understanding of the variability in outcomes and highlights the importance of methodological clarity in research. Our findings, although not identical to the original study, offer **valuable insights and underscore the**

need for comprehensive reporting in scientific research to ensure full reproducibility and reliability.

B. Reproducibility of the results

Reporting experiments to verify the reproducibility of one's own results is not a common practice in many communities. We have observed through various runs that the only way to reproduce exactly the same values is by using the same GPU and the same hyperparameters. By fixing these variables and by controlling the seed during training, we were able to reproduce the exact values. Altering these variables cause different outcomes, making it necessary to conduct a significance test to determine if the difference in the results is statistically significant. We performed these checks using only the BioBERT model because its training process is faster, and testing all possible combinations with both models would be computationally expensive and unnecessary for determining that our setup is reproducible. We chose the Wilcoxon rank-sum test because we cannot assume that the results follow a specific distribution as, for example, the t-test does.

Using the same GPU, we compared the results obtained when training with rounded hyperparameters (Table II) and the exact hyperparameters used to report the values of the F1-scores in this manuscript. In terms of micro, macro, and weighted F1 scores, the p-values obtained were 0.8345, 0.9168, and 0.7540, respectively. Additionally, for the per-class F1-scores, the high p-values (all above 0.05) indicate that there are no statistically significant differences between the F1-scores of the models trained with rounded hyperparameters and those trained with exact hyperparameters. We can conclude that **small modifications or rounding of hyperparameters do not significantly affect the model's performance**, which simplifies hyperparameter tuning and enhances the reproducibility of experiments.

Conversely, we have found that under different hardware and/or with different hyperparameters, some p-values do occasionally fall below 0.05. Therefore, to reproduce our results, it would be necessary to use exactly the same hyperparameters or approximate ones to ensure that there is no statistically significant difference between the results across all entities.

After detailing all the decisions made to compensate the missing information in [6] either arbitrarily or with the aim of facilitating the reproducibility of our experiments, and also having conducted an analysis of our own reproducibility, **we provide the necessary information for our experiment to be replicated under the configuration indicated** in Section II-B and thus to be able to obtain the same results. Information about the software used and how to use our code can be found in more detail in our GitHub repository[9].

IV. Generalization for NER tasks

We will gather information found in the literature and add the lessons learned during the reproducibility process to classify best practices based on workflow stages. This helps to trace discrepancies in outputs as they occur, enhancing reliability and explainability. Detecting reasons for differences allows rectification without compromising reliability. Our novel contribution is a systematic, step-by-step reproducibility approach, providing essential information to enhance reproducibility, and it could be helpful to define standard practices in NER.

A. Defining relevant information for reproducibility in NER

The proposed workflow for defining reproducible NER is divided into five main steps:

1. ***Data collection.*** It defines the collection of the data: where it has been obtained from, the queries made to perform data screening, if they have an identifier (ID), a list with the IDs of the selected documents. That is, all the **necessary information to obtain exactly the same dataset** as the one proposed. Conversely, if the samples are a subset of a known dataset, the criteria chosen for data selection should be specified, or a script that was used to retrieve/select exactly the same instances, the importance of retrieval method details was also pointed out in [23]. Findable and accessible according to FAIR principles [24] should be guaranteed at this step of the workflow.
   Without providing sufficient information to work with the same dataset, comparison of the outcomes of both experiments would be impossible, because, fundamentally, the models have been trained and evaluated with different datasets and therefore, possibly with different distributions of variables such as the number of words per text, the types of text used, as well as the vocabulary that appears, among others.
2. ***Data preprocessing.*** Once the data is selected, it has to be preprocessed in order to have an understandable format for the models. This step includes the **cleaning process** of texts, such as removing special characters, white space, etc., or for example, if all the text of an article has been concatenated consecutively into a plain text without differentiating sections, or how the appearance of tables, figures, captions, etc., has been managed. Additionally, for supervised learning the **labeling data process** is required and sometimes the aid of an expert is mandatory. Specifically, in NER task, if a novel dataset is being presented, the preparation of texts annotated by experts is a necessary and meticulous process that must be explained in detail: how many experts have contributed to labeling the texts, what instructions they received to carry out entity recognition, as well as an analysis of the reliability of the generated annotated data. This provides transparency in the process of generating labeled data, increasing its reliability for subsequent use. Usually, unless presenting a new dataset, the process explained previously at this point has already been conducted by other researchers before, and direct access to the cleaned and labeled dataset is available.
   In any case, in terms of preprocessing, the only remaining decision would be to choose the **transformation of the**

[9]https://github.com/cuevascarlos/ClinicalTrials

dataset into the models input format. In order to ensure that models are trained with the same data in the same format, the necessary information to obtain the same inputs must be reported, e.g., whether it is the publicly available framework used for transformation or a script developed specifically for the task. Additionally, in the particular case of NER tasks, other factors must be addressed, such as the labeling format used (IO, IOB, BIOES, etc.) and how situations like overlapping or nesting entities have been managed. If not, when transforming the data into the required format using different tools, factors such as those previously mentioned can be managed in different ways. This would again cause differences in the datasets used to train and evaluate the models, making it unreliable to compare the results of the two studies. In case of discrepancies, it would be impossible to determine whether they are caused by the training process or are a consequence of working with different datasets.

3. ***Experimental set-up.*** This part of the workflow is used for defining the **partition/split of the data** considered, the **evaluation metrics considered** (per-class, micro, macro, weighted, etc.), the modes of metric computation, and the choice of the library used for their computation. For example, in NER, metrics can be computed using two different approaches, as it has been previously discussed and reported in [13], [25]. Details of all these decisions involves communicating the set of samples each dataset contains (train, development, test), the metrics chosen to analyze model performance, and how they have been calculated. This is especially important when it comes to reproducing outcomes. Different sizes of the sets will lead to different results and different libraries and frameworks may approach the calculation of the same metric in different ways, resulting in different outcomes [25].
   Furthermore, it is very important to provide **software information**, such as the versions of the used libraries, due to possible future incompatibilities or changes in the library's internal methodology and the **hardware used** to run the training process which was also identified by [26]. Finally, it should be specified **where the considered models come from and the versions used**, for example, there are several versions of BioBERT publicly available in HuggingFace and each has been pre-trained with a different approach and produces different outputs.
4. ***Training process.*** The fourth step of the workflow focuses on the training process of the models. During this step the model(s) is/are trained using a selection of hyperparameters or by defining a parameter search and taking into account stochasticity of the models.
   Information about this step of the workflow is not commonly reported in detail in articles, which hinders the possibility of reproducing the results. It is crucial to provide relevant information during this process. As shown in our study, the outputs of the models would vary and be strongly affected by the hyperparameters selection. Moreover, prior work has shown the impact of hyperparameters on the robustness of transformer models [27], therefore details about the **parameters search** (if done) or the **hyperparameters used** should be reported. Finally, the seed plays a non-negligible role and **several runs** must be carried out to take this factor into consideration.
5. ***Model evaluation.*** Finally, the last part of the workflow focuses on the performance assessment of the trained models. Once experimental set-up has been defined, the performance is analyzed in terms of the evaluation metrics.
   Reporting the **mean and standard deviation** values and doing a **statistical analysis** of the results enhance the robustness of the model versus stochasticity in parameter initialization. This increases the reliability of the results and facilitates the comparison between studies. Otherwise, if this practice is not carried out, only the best performance of the models could be reported, making reproducibility even more complicated. This approach is more realistic and allows for a more in-depth analysis of the quality of the obtained models. Prior work also promoted reporting such information by requiring conference submissions to mark a checklist [28].

Table V provides a summary of the information we propose in each of the steps of the workflow. It is intended to enhance process tracking and identify the cause of differences at each step if they exist. This fact allows to analyze **reproducibility in two steps**: the *preparation of data* and the *models training* which allows to identify internal discrepancies in each of the workflow steps and analyze the validity of each of the approaches that cause the difference. Additionally, we summarize the information provided in the reference paper and the information provided here based on the data we have identified as necessary to be able to reproduce the results.

V. Related work

Reproducibility is one of the cornerstones of scientific research, as the inability to reproduce results hampers the progress of cumulative science. Many research communities have reported significant challenges in ensuring observational, statistical, and computational reproducibility. The full extent of the **reproducibility crisis** was highlighted in an early study published in 2016 in Nature [29], which surveyed over 1,500 researchers and revealed that 70% of scientists were unable to reproduce someone else's results, and more than half were unable to reproduce their own experiments. Other surveys have followed a few years later, focusing on computational reproducibility [30], [31]. They report that most of the survey's respondents are aware of the issues of reproducibility and aware that publishing code in a repository was an insufficient practice to guarantee reproducibility.

In response to this crisis, **various initiatives have been developed** to promote computational reproducibility, particularly in the database community (e.g., reproducibility badges in VLDB and SIGMOD conferences) and in the Machine Learning community (e.g., the Reproducibility in ML Workshop series at ICML and the reproducibility checklist required by

Step	Information	Reported in [6]	Reported in Section II-B
Data collection	1.1. Origin of the data*	✓✓	
	1.2. Sample size*	✓✓	
Data preprocessing	2.1. Cleaning process*		
	2.2. Annotation preparation (number of experts, instructions for entity recognition, quality analysis)*	✓	
	2.3. Transformation tool for compatibility with models input format (entity format, dealing with overlapping entities,etc.)		✓✓
Experimental set-up	3.1. How samples were allocated for training/validation/testing	✓	In HuggingFace
	3.2. Definition of the specific metrics or statistics used to report results	✓✓	✓✓
	3.3. Framework/scorer used for computing the metrics or statistics		✓✓
	3.4. Software environment		In GitHub
	3.5. Hardware environment		✓✓
	3.6. Clear description of the model (availability and version)		✓✓
Training process	4.1. The range of hyperparameters considered, method to select the best hyperparameter configuration, and specification of all hyperparameters used to generate results		✓✓
	4.2. Exact number of evaluation runs	✓✓	✓✓
Model evaluation	5.1. A description of results with central tendency & variation		✓✓
	5.2. Statistical analysis (significance test)		✓✓

TABLE V

SUMMARY OF THE EXPECTED INFORMATION TO BE REPORTED IN EACH OF THE STEPS OF THE NER REPRODUCIBILITY WORKFLOW. STEP COLUMN INDICATES THE PART OF THE WORKFLOW EACH HELPFUL INFORMATION BELONGS TO.* CASES IN INFORMATION COLUMN ARE SUPPOSED TO HAVE BEEN PREVIOUSLY DONE BY THE DATASET PUBLISHERS. REPORTED IN [6] COLUMN SUMMARIZES THE INFORMATION PROVIDED IN THE REFERENCE PAPER WHILE REPORTED IN SECTION II-B SUMMARIZES THE INFORMATION WE PROVIDE IN THIS MANUSCRIPT. ONE CHECK MARK MEANS PARTIALLY EXPLAINED AND DOUBLE CHECK MEANS EXPLAINED IN DETAILS. SOME TERMS HAVE BEEN EXTRACTED FROM THE CHECKLIST PROPOSED IN [28].

major ML conferences[10]). In the field of Natural Language Processing (NLP), similar initiatives have been introduced a few years later, and the issue remains considered as an increasingly important topic.

The **definitions of what constitutes a reproducible, replicable, or re-runnable result have also been extensively discussed** across communities. A major report published in 2019 by the National Academies Press [32] has provided a comprehensive review of the state of the art in this domain. In the NLP community, two early papers attempted to offer some clarity on the current definitions and even proposed consensual definitions. Notably, a meta-review by Belz *et al.* (2021) [33] delineates differences and similarities between the existing approaches and provides pointers to common denominators. On the other hand, Cohen *et al.* (2018) [34] emphasize that reproducibility should be considered across three distinct dimensions: values, findings (the relationship between the values for some reported figure of merit with respect to two or more dependent variables), and conclusions (a broad inference drawn from the results of the reported research, interpreting the findings). This multidimensional approach to reproducibility allows for a more precise identification of how results have been reproduced—or not. In contrast, many other research communities typically consider only two dimensions, focusing on the ability to obtain again the exact same values or the same scientific result [35].

In the particular case of NER tasks, [36] studied the lack of information for experiment reproducibility purpose by comparing different NER software results and showed the influence of the NER system selection on the performance. In addition, previous studies [21], [37] have conducted an exhaustive analysis of reproducibility in NER, highlighting and analyzing the **difficulties in reproducing results in NER tasks**. However, both studies use the Stanford Named Entity Recogniser [38] and do not employ pre-trained transformers as models, which introduces new factors that complicate the reproducibility task.

Concretely, in [27] the literature is analyzed to understand how pre-trained transformers are used in practice, focusing on how models are tuned and results on downstream tasks are reported. A **widespread lack of information is identified in the manuscripts** that utilize these models. Casola *et al.* also analyzed the relation between lack of robustness and hyperparameters. Considering that this process is not commonly reported in detail, and that many studies present only a single run without accounting for stochasticity, it is concluded that the comparison between models is often problematic and shared and a trusted framework for model comparison is needed in the NLP community.

Knowing the existence of the problem, there are many studies on reproducibility in NLP, particularly proposing good practices to facilitate result replication. However, **these studies mostly categorize premises based on the expected objectives** in terms of reproducibility (e.g., Digan *et al.* [39] categorize the guidelines in terms of traceability, standarization, versioning, usability and shareability). Moreover, [26] highlights nine critical aspects for reproduction in text mining. Other studies such as [28], [40], [41] address the reproducibility of machine learning in general where guidelines are proposed to improve the possibilities of reproducing results. In the particular case of NER, [18] provides guidelines to address the problem we have found in terms of the computation of evaluation metrics due to the fact that the difference between models can be statistically significant depending on which approach has been chosen for the metrics.

[10]https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf

VI. Conclusion

Reproducibility remains a critical issue in the field of natural language processing (NLP), particularly with pre-trained transformer models for tasks such as named entity recognition (NER). Our study aimed to replicate and validate the results reported by previous researchers using a precisely annotated dataset, so that we can then build on it for the purpose of our projects, especially FAIRClinical and to some extent ShareFAIR. However, we **encountered significant challenges due to insufficient methodological details provided** in the original work [6]. We identified several key gaps, including unspecified preprocessing steps, lack of clarity on hyperparameter optimization, and the absence of model configuration details. These gaps needed independent decisions that likely contributed to variations in our results compared to the original study.

Despite these obstacles, we made substantial contributions to the field by **taking comprehensive measures to ensure the robustness of our findings**. We utilized open-source tools whenever possible and provided detailed information for all decisions made throughout each experiment. This level of precision and transparency in the training process, coupled with a thorough analysis of the nature of the differences obtained, enhances the reliability and explainability of the results reported by pre-trained transformers on the PICO-breast-cancer dataset compared to previously published results. Consequently, this setup can be used in future projects for NER applied in various scenarios, allowing the transfer of knowledge and comprehension acquired in this case study. In addition, ongoing work includes working on a framework to promote the use of tools enhancing computational reproducibility. More specifically we are designing a set of reprohackathons [42] to help getting a more precise understanding of the results of papers, specifically in NER tasks.

Furthermore, this work also contributes to **generalizing the issues faced**, and comparing these problems with existing literature. We hope that the information gathered will help establish standards for decisions made throughout the workflow, potentially increasing reproducibility. This should also motivate researchers to provide more detailed information when reporting results in scientific articles, as we have done, thereby serving as a guide to enhance reproducibility within natural language processing.

Acknowledgment

This work was supported in part by the National Research Agency under the France 2030 program, with reference to ANR-22-PESN-0007. This work was also supported by the CHIST-ERA grant CHIST-ERA-22-ORD-02, by the Agence Nationale de la Recherche (ANR-23-CHRO-0008-01) and the ANR grant CHAIRE CPJ (23HR3101).

References

[1] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[2] J. M. Giorgi and G. D. Bader, "Transfer learning for biomedical named entity recognition with neural networks," *Bioinformatics*, vol. 34, no. 23, pp. 4087–4094, 2018.

[3] B. C. Wallace, J. Kuiper, A. Sharma, M. B. Zhu, and I. J. Marshall, "Extracting pico sentences from clinical trial reports using supervised distant supervision," *Journal of Machine Learning Research*, vol. 17, no. 132, pp. 1–25, 2016.

[4] T. Kang, S. Zhang, Y. Tang, G. W. Hruby, A. Rusanov, N. Elhadad, and C. Weng, "Eliie: An open-source information extraction system for clinical trial eligibility criteria," *Journal of the American Medical Informatics Association*, vol. 24, no. 6, pp. 1062–1071, 2017.

[5] S. Tian, A. Erdengasileng, X. Yang, Y. Guo, Y. Wu, J. Zhang, J. Bian, and Z. He, "Transformer-based named entity recognition for parsing clinical trial eligibility criteria," in *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2021, pp. 1–6.

[6] F. Mutinda, K. Liew, S. Yada, S. Wakamiya, and E. Aramaki, "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*, 2022, pp. 26–31.

[7] B. Nye, J. J. Li, R. Patel, Y. Yang, I. J. Marshall, A. Nenkova, and B. C. Wallace, "A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2018. NIH Public Access, 2018, p. 197.

[8] D. Jin and P. Szolovits, "PICO element detection in medical text via long short-term memory neural networks," in *Proceedings of the BioNLP 2018 workshop*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 67–75. [Online]. Available: https://aclanthology.org/W18-2308

[9] S. N. Kim, D. Martinez, L. Cavedon, and L. Yencken, "Automatic classification of sentences to support evidence based medicine," in *BMC bioinformatics*, vol. 12. Springer, 2011, pp. 1–10.

[10] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[11] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[12] H. Nakayama, "seqeval: A python framework for sequence labeling evaluation," 2018. [Online]. Available: https://github.com/chakki-works/seqeval

[13] Y. Hu, V. K. Keloth, K. Raja, Y. Chen, and H. Xu, "Towards precise PICO extraction from abstracts of randomized controlled trials using a section-specific learning approach," *Bioinformatics*, vol. 39, no. 9, p. btad542, 2023.

[14] Q. Wang, J. Liao, M. Lapata, and M. Macleod, "PICO entity extraction for preclinical animal literature," *Systematic Reviews*, vol. 11, no. 1, p. 209, 2022.

[15] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the CoNLL-2000 shared task chunking," in *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, 2000. [Online]. Available: https://aclanthology.org/W00-0726

[16] E. F. Tjong Kim Sang, "Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition," in *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002. [Online]. Available: https://aclanthology.org/W02-2024

[17] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147. [Online]. Available: https://aclanthology.org/W03-0419

[18] C. Lignos and C. Palen-Michel, "Seqscore: Addressing barriers to reproducible named entity recognition evaluation," in *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*. Association for Computational Linguistics, 2021, pp. 40–50.

[19] N. Chinchor and P. Robinson, "MUC-7 named entity task definition," in *Proceedings of the 7th Conference on Message Understanding*, vol. 29, 1997, pp. 1–21.

[20] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[21] A. Fokkens, M. Van Erp, M. Postma, T. Pedersen, P. Vossen, and N. Freire, "Offspring from reproduction problems: What replication

failure teaches us," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1691–1701.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] B. K. Olorisade, P. Brereton, and P. Andras, "Reproducibility of studies on text mining for citation screening in systematic reviews: evaluation and checklist," *Journal of biomedical informatics*, vol. 73, pp. 1–13, 2017.

[24] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[25] T. Kang, S. Zou, and C. Weng, "Pretraining to recognize PICO elements from randomized controlled trial literature," *Studies in health technology and informatics*, vol. 264, p. 188, 2019.

[26] B. K. Olorisade, P. Brereton, and P. Andras, "Reproducibility in machine learning-based studies: An example of text mining," in *Proceedings of the Reproducibility in Machine Learning Workshop, ICML*, 2017.

[27] S. Casola, I. Lauriola, and A. Lavelli, "Pre-trained transformers: an empirical comparison," *Machine Learning with Applications*, vol. 9, p. 100334, 2022.

[28] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché Buc, E. Fox, and H. Larochelle, "Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program)," *Journal of Machine Learning Research*, vol. 22, no. 164, pp. 1–20, 2021.

[29] M. Baker, "Reproducibility crisis," *Nature*, vol. 533, no. 26, pp. 353–66, 2016.

[30] B. A. Plale, T. Malik, and L. C. Pouchard, "Reproducibility practice in high-performance computing: Community survey results," *Computing in Science & Engineering*, vol. 23, no. 5, pp. 55–60, 2021.

[31] M. Mieskes, K. Fort, A. Névéol, C. Grouin, and K. B. Cohen, "Community perspective on replicability in natural language processing," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 768–775.

[32] National Academies of Sciences and Policy and Global Affairs and Board on Research Data and Information and Division on Engineering and Physical Sciences and Committee on Applied and Theoretical Statistics and Board on Mathematical Sciences and others, *Reproducibility and replicability in science*. National Academies Press, 2019.

[33] A. Belz, S. Agarwal, A. Shimorina, and E. Reiter, "A systematic review of reproducibility research in natural language processing," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 381–393.

[34] K. B. Cohen, J. Xia, P. Zweigenbaum, T. J. Callahan, O. Hargraves, F. Goss, N. Ide, A. Névéol, C. Grouin, and L. E. Hunter, "Three dimensions of reproducibility in natural language processing," in *Proceedings of the LREC International conference on language resources & evaluation*, vol. 2018. NIH Public Access, 2018, pp. 156–165.

[35] S. Cohen-Boulakia, K. Belhajjame, O. Collin, J. Chopard, C. Froidevaux, A. Gaignard, K. Hinsen, P. Larmande, Y. L. Bras, F. Lemoine, F. Mareuil, H. Ménager, C. Pradal, and C. Blanchet, "Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities," *Future Generation Computer Systems*, vol. 75, pp. 284–298, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17300316

[36] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, "A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 338–343.

[37] M. Van Erp and L. Van der Meij, "Reusable research? a case study in named entity recognition," 2013.

[38] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*, 2005, pp. 363–370.

[39] W. Digan, A. Névéol, A. Neuraz, M. Wack, D. Baudoin, A. Burgun, and B. Rance, "Can reproducibility be improved in clinical natural language processing? a study of 7 clinical nlp suites," *Journal of the American Medical Informatics Association*, vol. 28, no. 3, pp. 504–515, 2021.

[40] B. J. Heil, M. M. Hoffman, F. Markowetz, S.-I. Lee, C. S. Greene, and S. C. Hicks, "Reproducibility standards for machine learning in the life sciences," *Nature Methods*, vol. 18, no. 10, pp. 1132–1135, 2021.

[41] R. Tatman, J. VanderPlas, and S. Dane, "A practical taxonomy of reproducibility for machine learning research," 2018.

[42] T. Cokelaer, S. Cohen-Boulakia, and F. Lemoine, "Reprohackathons: promoting reproducibility in bioinformatics through training," *Bioinformatics*, vol. 39, no. Supplement_1, pp. i11–i20, 2023.