Public Notice

This notice is a Manifest about the plagiarism case of the work, *jury*.

Date: Jul 30, 2023
Last Revision: Aug 10, 2023
Authors: Devrim Çavuşoğlu

This document is intended to introduce the case of the plagiarism of the package *jury*, exhibit the evidence and build a document whose current version or later versions (which will be tracked by the edit date of the document) may be used in potential phases of communication with either public, within the parties or commissions as a supplementary material via a private channel.

In this document, there are two sides that are going to be stated regarding the plagiarism case. Although this file does not strictly need to constitute as a component of a legal case, the parties in this document will be referred in the following scheme:

- The first party, the copyright holder of the original work, codebase, components, documents, methodology and any other material related to the aforementioned work will be referred to as PLAINTIFF.
- The second party, the alleged plagiarizer of the original work and the authors of the accused work that is aforementioned will be referred as DEFENDANT.

Collection of Evidence

This chapter is composed of related documents, codes, code snippets, statements, or any evidence that could yield plagiarism of a copyrighted work, Jury.

Background of Jury

This section is devoted to the background of the original work, *jury*. Jury is a comprehensive evaluation toolkit that aims to standardize the evaluation procedures in the deep learning field. Its contributions include, but are not limited to, a unified interface, task mapping among metrics, concurrency, support for multiple predictions and multiple references.

It was originally developed by the OBSS AI team (within the company OBSS, Open Business Software Solutions) for the use of internal projects. It was then released for public use and for the contribution to the open-source community. The publication of the work was through the Github repository under `obss` organization. The work is licensed under MIT License.

The first commit to the public codebase and as a package first PyPi release were simultaneously on `Jul 14, 2021`.
The first commit to the public codebase for the second party's work was on `Feb 10, 2022`.

The latest release of Jury (v2.2.4 as of `Jul 20, 2023`) was in `Jun 15, 2023`. Any publicly available code, documentation, any artifact, material or component that are either product of the use of a third party tool for the development of the aforementioned work or material (e.g. documentation, study reports) outcome of any discussion, brainstorming or meeting will be referred to as the WORK.

For convenience we also include the MIT License that *jury* use below:

```
None
MIT License

Copyright (c) 2021 Open Business Software Solutions

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

The Bad Conduct

Plagiarism can be viewed in two folds; unintentional and intentional plagiarism, and as it is described commonly intentional plagiarism is committed when the sources left out deliberately while unintentional plagiarism is committed generally due to lack of knowledge or attentiveness[1].

We are going to present two parts of the work from the second party. Although it could be seen as from the codebase the case is an unintentional plagiarism, it can be clearly understood that this is not the case when one reviews the paper of the same work.

1. Codebase

We have described the background of WORK, mentioned license content, and the coverage of WORK. Clearly "MIT License" is a basic, easy to understand and simple license which allows actions such as commercial use, modification and distribution of the software with a condition that to include the copyright notice in the works that use the aforementioned software. The following statement is the only requirement of the MIT License.

```
None
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
```

The alleged plagiarizer extended and utilized WORK extensively in their variant. However, the second party failed to put the copyright notices of WORK in their following work. There does not appear for their publicly available codebase that no file or content includes the required copyright notice by the licensor of the copyright holder. This appears to be a clear violation of the use of the licensed work.

As of their latest release (as of the time of writing this very document) v0.9.6 there is no inclusion of copyright notices in their work. The copyright notice for the copyright holder is not present in their repository, markdown files or any other package files.

Furthermore, Article 6bis of the Berne Convention for the Protection of Literary and Artistic Rights ("Berne Convention") sets forth "the right of paternity", which grants authors the right to be recognized as the creator of their work; and, ensures that their work is not falsely attributed to someone else. DEFENDANT's paper was

[1] *Source:* https://www.htbibl.lu.se/en/student/writing-referencing/plagiarism/

indexed at ACL Anthology, which is accessible throughout the states that are contracting parties to the Berne Convention. The fact that there is no attribution to PLAINTIFF as an author is an infringement of the right of paternity in such territories.

Internet Archive

- Archived v0.9.6 release
- Archived Release page/Github tag
- Archived Github Hompage of the repository

For a detailed view on the codebase, we first pose the directory structures of both works. Directory structures including the detailed view on differences can be accessed at https://www.diffchecker.com/bDF5OzGa/.

2. Paper

The second party also wrote a paper based on their work and submitted to a conference, where the paper was accepted and published later in 2022. The paper was submitted to COLING'22. The paper is indexed on ACL Anthology and can be accessed here.

In the paper, they actually cited[2]/mentioned our framework *jury* only in two places:

- the last paragraph of the "Background and Related Work" section as follows:

```
None
"... To our best knowledge, NLGEval (Sharma et al., 2017), HugginFace Datasets
(Lhoest et al., 2021), Evalaute3 , TorchMetrics (Detlefsen et al., 2022), and
Jury (Cavusoglu et al., 2022) are the only resources available."
```

- In the caption of *Table 1* as:

```
None
Table 1: Comparison of our library (v1.0.0) with existing NLG evaluation
packages: NLGEval (v2.3.0), Datasets (v2.4.0), Evaluate (v0.2.2), TorchMetrics
(v0.8.2), Jury (v2.2). "+ Datasets" stands for an automatic fallback towards
HuggingFace Datasets in case of unsupported metrics (lower bound).
```

[2] Citation here naturally and correctly refers to the online source as a package *jury* since there was no publication available. A notice for citation is also given in the readme of the homepage of the repository at https://github.com/obss/jury/tree/main#-citation-

There is nowhere else that citation is included or a mention is made. However, it is important to note that they cited and mentioned *jury* in their paper meaning that they are aware of WORK. Thus, the following exemplar of bad conduct strongly suggest that this plagiarism case is potentially intentional.

We will be exhibiting the related chunks of the paper as how the authors of the alleged plagiarized work introduce or present their work in their own words. All of the chunks represented can have direct quotations, references to sections, appendix or material used in the aforementioned COLING'22 paper of DEFENDANT.

The direct quotations in the following chunks will be represented in the following format.

> "To facilitate researchers to judge the effectiveness of their models broadly, we introduce NLG-METRICVERSE—an end-to-end open-source library for NLG evaluation based on Python."

 # Chunk - 1 (§5 Main Modules)
In section 5, where they state the foreword and briefly introduce their package. They mention the open-source works that they used for their package's as a building block.

> "NLGMETRICVERSE is in turn built on top of opensource libraries, including Datasets (Lhoest et al., 2021), NumPy (van der Walt et al., 2011), SciPy (Virtanen et al., 2020), and Matplotlib (Hunter, 2007)."

As they stated, *jury* is not included in the list of open source libraries which their work is built on top of.

 # Chunk - 2 (§5.1 Metrics)
As the authors stated in this section, they described as "they built" automated tests in their codebase to ensure the integrity of metrics.

> "We ensure the integrity of each metric within the codebase through automated tests."

While this may be practically correct, the tests are copied/transferred with small or no changes without including copyright notice from the public repository of WORK.

 # Chunk - 3 (§5.1 Metrics/Input Format)
As the authors stated in this section, they described as "they built" the design of a unified metric input type which allows passing multiple instances for both candidates and references.

> "We design a unified metric input type, also handling n-arity for candidate and reference texts (Table 3)—a feature as vital as neglected by current systems."

This design was the primary focus of PLAINTIFF's work, *jury,* and it is already mentioned in the README.md file that how the authors built a unified interface around metric input types for candidates/predictions and references.

 # Chunk - 4 (§5.1 Metrics/Metrics Application)
As the authors stated in this section, they described how one could use their work to evaluate the generated texts by system.

> "Evaluating artificial text requires just two lines of code: (i) create a Scorer object with the desired metrics; (ii) apply the Scorer object to the input data. So, many metrics may be executed in one go."

PLAINTIFF designed a "Scorer" object in WORK which functions the same way, and interestingly enough this object and the related code is "copied" over to the second party's work as is.

 # Chunk - 5 (§5.1 Metrics/Metrics Application)
As the authors stated in this section, they described how they added parameters to handle multiple instances of predictions/references by "reduce_fn"

> "If a prediction needs to be compared against multiple references, the user is left with the possibility to specify the aggregation strategy of preference through the reduce_fn parameter. For example, reduce_fn="max" considers only the prediction-reference pair with the highest score for each dataset instance. Inherently, NLGMETRICVERSE allows all NumPy function names and custom aggregation functions as well."

Here, in WORK both the names of the arguments/parameters are originally constructed by PLAINTIFF in their work/package as is. The functionality is the same as well as the structure due to copy of the work without copyright notice.

 # Chunk - 6 (§5.1 Metrics/Metrics Application)
As the authors stated in this section, they described how one could use their work to evaluate the generated texts by system.

> "An asynchronous execution with a separate process for each metric can be specified to push efficiency and scalability (run_concurrent), bringing parallelism to the evaluation loop. "

Concurrent execution for performing metric computations (intended for parallel computing of different metrics on the same input) was originally built and developed by PLAINTIFF and it is in the aforementioned repository of the package, *jury.*

Chunk - 7 (§5.1 Metrics/Metrics Application)
As the authors stated in this section, they described their work, NLG-METRICVERSE, falls back to implementation of *datasets* package if it's not yet supported.

> "By employing the load_metric() function for step (i), NLG-METRICVERSE falls back to the Datasets implementation in case of metrics not yet supported. Consequently, our library englobes at least any metrics that the Datasets package has."

This behavior mentioned here was genuinely developed and added as a built-in feature of the original work, *jury* by PLAINTIFF in WORK.

Chunk - 8 (§5.1 Metrics/Metrics Application)
As the authors stated in this section, they described their work "Scorer" component is built such that it allows for users to pass different hyperparameters or arguments for different metrics

> "When defining the Scorer, a maximum degree of freedom is retained to allow the setting of metric-specific hyperparameters and different instantiations of the same metric (Figure 4). Further, since metrics generally involve several hyperparameters and results can deviate significantly for other choices, we accompany the output with a config report (hyperparams setting, hardware setup, etc.) for increasing comparability and replicability."

This behavior mentioned here was genuinely developed and added as a built-in feature of the original work, *jury* by PLAINTIFF in WORK.

Chunk - 9 (§5.1 Metrics/Custom Metric)
As the authors stated in this section, they described their work "Scorer" component is built such that it allows for users to pass different hyperparameters or arguments for different metrics

> "NLG-METRICVERSE offers a flexible and uniform API for easily creating custom user-defined metrics. It only requires inheriting the MetricForNLG class (i.e., the common base
> class for each metric) and implementing the abstract functions linked to the possible input formats
> (Figure 6). We pursue the idea of enabling the user to create complex setups without superimposing constraints that may not suit future research."

```
1  class CustomMetric(MetricForNLG):
2    def _compute_single_pred_single_ref(
3        self, preds, refs, reduce_fn=None,
         ↪  **kwargs
4    ): ...
5    def _compute_single_pred_multi_ref ...
6    def _compute_multi_pred_multi_ref ...
```

Figure 6: Custom metric implementation.

PLAINTIFF has clearly stated in the repository (README.md) that mentioned about the custom metric implementation. As one can apparently see, the quoted section from the paper of DEFENDANT and WORK of PLAINTIFF is almost absolutely the same, where they only renamed/modified the base class as "MetricForNLG" from "MetricForTask" which is intended to be a generic base class for many tasks in *jury*.

Conclusion

In this document, we have introduced the background of the genuine and original work, *jury*, and how plagiarism is committed by the authors of the accused work, *nlg-metricverse*. We gave detailed evidence that strongly suggests there has been plagiarism committed. Furthermore, by section 1 of the chapter "The Bad Conduct", which lays the examples from the code part of the case one may not be certain at whether the case is intentional or unintentional. However, when we look at the part related to the published paper, one can conclude that the case is intentional plagiarism. In the paper, the authors clearly posed the components of the original work, *jury*, as they built and developed it.

To sum up, after reading this document carefully and examining the evidence we laid out, we leave the readers to their inner selves.

Also, we here acknowledge that DEFENDANT's work may have improvements and genuine extension on top of WORK (e.g. visualization). However, this does not imply by any means that our claim presented in this very document is false or mistargeted. While we appreciate the extensions and derivative works, the plagiarism case is still there.

Contact

For further information, questions, raising issues and any other information that may be related to this case, please reach the author via
devrim.cavusoglu@obss.tech