Automated Rule-Based Data Cleaning Using NLP

Konstantinos Mavrogiorgos
University of Piraeus
Piraeus, Greece
komav@unipi.gr

Argyro Mavrogiorgou
University of Piraeus
Piraeus, Greece
margy@unipi.gr

Athanasios Kiourtis
University of Piraeus
Piraeus, Greece
kiourtis@unipi.gr

Nikolaos Zafeiropoulos
University of Piraeus
Piraeus, Greece
nikolaszaf@unipi.gr

Spyridon Kleftakis
University of Piraeus
Piraeus, Greece
spiroskleft@unipi.gr

Dimosthenis Kyriazis
University of Piraeus
Piraeus, Greece
dimos@unipi.gr

***Abstract*—Data Cleaning is a subfield of Data Mining that is thriving in the recent years. Ensuring the reliability of data, either when generated or received, is of vital importance to provide the best services possible to users. Accomplishing the aforementioned task is easier said than done, since data are complex, generated at an extremely high rate and are of enormous size. A variety of techniques and methods that are part of other subfields from the domain of the Computer Science have been invoked to assist in making Data Cleaning the most efficient and effective possible. Those subfields include, among others, Natural Language Processing (NLP), which in essence refers to the interaction among computers and human language, seeking to find a way to program computers to be able to process and analyze huge volumes of human language data. NLP is a concept that exists for a long time, but, as time goes by, it is proposed that it can be applied to a variety of concepts that are not solely NLP-related. In this paper, a rule-based data cleaning mechanism is proposed, which utilizes NLP to ensure data reliability. Making use of NLP enabled the mechanism not only to be extremely effective but also to be a lot more efficient compared to other corresponding mechanisms that do not utilize NLP. The mechanism was evaluated upon diverse healthcare datasets, not however being limited to the healthcare domain, but supporting a generalized data cleaning concept.**

I. Introduction

It is widely accepted that, nowadays, data play a crucial role on every aspect of daily life. There exist an extensive range of devices that are capable of generating data, which are the so-called data sources. Indicative examples of such devices are sensors, wearables, and smartphones. One characteristic that most of these sources have in common is the absence of performing actions related to the cleaning of data in order to ensure their reliability. Thus, every mechanism and platform that retrieves data from those sources should, somehow, effectively perform the aforementioned task. Recently, a survey [1] highlighted the importance of data in decision making, and as a result the importance of performing efficient data cleaning. It stated that the global value of data used for solely marketing purposes from 2017 to 2021 is 52 billion USD from which 30 billion USD refer to the value of that kind of data in the United States. In contrast to the past, the amount of the generated data is now ferocious and is estimated to reach the outstanding size of 181 zettabytes by 2025 [2]. This fact indicates that a data cleaning mechanism should not only just perform data cleaning per se, but also utilize techniques that enable it to manage huge amount of data at a very fast and efficient way.

The above have led the research community to seek alternative solutions from other domains of the Data Mining that can be integrated into data cleaning mechanisms in order to support them and extend their usage and potential. For example, several ML algorithms are now being widely used in data cleaning, as for example the algorithms of k-Nearest Neighbors (KNN), Decision Trees (DT), and Random Forest (RF), so as to predict missing values that are present into a dataset. At the same time, another subfield of the Data Mining domain that has recently been introduced to be utilized in data cleaning is NLP. NLP is already being used in a variety of use cases and its global marketing revenue is expected to reach 43 billion USD by 2025 [3]. The concept of NLP exists since the early ‘50s, but lately there is an exponential growth regarding its use due to the advancements in ML and Deep Learning (DP).

Towards this direction, NLP can be used in data cleaning mechanisms as well, especially in the cases where those mechanisms’ cleaning actions are based on rules. In essence, in rule-based data cleaning, a mechanism has the ability to apply the cleaning actions based on a set of rules that have been developed and stored in a specific structure (i.e., schema), where most of the times, these schemas consist of rules that have been developed by a data expert. For example, when it comes to health-related data, an expert could be a healthcare professional. Since the schema is ready, the mechanism is able to apply several data cleaning actions, based on the provided rules. However, if some data consist of features that are “unknown” (i.e., they are not included in the schema), then a new schema should be generated. This task would require extra time and effort from the data expert, while it could not take place in real

time (i.e., when the data are received). What is more, this process could not also be redundant, since maybe the difference between a known and unknown dataset could be just the name of the features and not the features themselves, thus creating a new schema that is, practically, the same with an old one. One possible solution to this problem is the application of NLP, where a cleaning mechanism will be able to recognize "similarities" between known and unknown datasets, even when the name of the features differs, thus exploiting a previously generated set of rules and avoiding the creation of a new one.

Based on the above, this work proposes a rule-based data cleaning mechanism that utilizes NLP to confront the existing challenges. To this context, the mechanism consists of three (3) discrete components, namely "Data Collection", "Data Storage" and "Data Cleaning". The "Data Collection" component gathers the unreliable data that are to be cleaned. The "Data Storage" component stores the ingested and the cleaned data into a suitable database, whilst the "Data Cleaning" component applies the required data cleaning and NLP procedures to guarantee the reliability and the high-quality of the ingested data.

The remaining of the current document is structured as shown below. Section II delivers a state-of-the-art analysis regarding data cleaning and NLP. In Section III, the architecture of the mechanism is thoroughly described, accompanied by its components' description. Section IV includes a discussion regarding the results of this paper and a comparison of them and the ground truth. Lastly, Section V recaps the overall performed work and includes insights concerning the future plans.

II. Literature Review

A. Data Cleaning

Data cleaning is a particular subdomain of Data Mining that focuses on correcting abnormalities in erroneous data. These abnormalities may refer to semantic errors, syntactic errors, missing values and duplicate records, just to mention a few [4]. There exists a plethora of techniques that are applied in the literature and aim to solve the aforementioned problems. The applied techniques may vary from basic concepts that are introduced in statistics to more complex concepts that derive from ML, depending on the nature of the underlying data and the problem that is required to be solved [5], [6].

In the recent years, there has been significant progress regarding data cleaning across diverse domains (e.g., healthcare, environment). More specifically, the authors in [7] propose a data cleaning mechanism that is based on local density in order to detect outliers. The mechanism is applied on multidimensional data that refer to wind power and the authors claim that the cleaning results that are based on the local density of samples are more objective and distinguishable. In another paper, the authors propose and evaluate a Python-based and open source module that applies both data cleaning and data profiling tasks called "Openclean" [8]. This module also provides a Graphical User Interface (GUI) and can be integrated into other applications. It is capable of performing basic cleaning actions such as outliers' detection and missing values prediction, whilst it can also integrate custom cleaning actions that have been developed by a developer/user. In another research paper, a data cleaning method that exploits Empirical Wavelet Transform (EWT) as well as Multiscale Fuzzy Entropy (MFE) is introduced, aiming to perform timeseries cleaning of data regarding water quality data in China. According to the paper's results, this method has the ability to improve the performance of the cleaning of the noisy data, not distorting at the same time the existing data that are noisy-free [9]. In another approach, dependency-based cleaning of data is performed, exploiting Ontology Functional Dependencies (OFDs). By doing so, the authors are able to minimize the false positive errors driving from the data cleaning methods that are based on traditional Functional Dependencies (FDs) [10]. The authors in [11] propose a fusion rule, polynomial fitting, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm so as to recognize outliers in data originating from wind turbines. The data are firstly filtered based on rules, then the method of linear fitting and DBSCAN are used to identify and remove abnormal data. The results of the paper suggest that this method can successfully recognize erroneous data. Moreover, the performance of the method is insensitive to the wind turbine parameters, so that it can be used in different wind farms. Finally, in [12] the authors propose a multi-layer mechanism for performing data cleaning to health-related data. The authors also test the mechanism with a wide range of health-related datasets, thus proving the efficacy of their approach.

B. Natural Language Processing

Natural Language Processing (NLP), also widely known as computational linguistics, is a subfield of Computer Science. It refers to the formation of computational models and processes that are capable of solving complex problems that refer to the understanding of the human language. NLP is currently a data-driven field that utilizes statistical and probabilistic computations, as well as ML [13].

In the literature, a variety of applications regarding NLP have been proposed in order to address issues regarding the processing of natural language, from which the most recent ones are listed below. To begin with, the authors in [14] utilize an NLP method to extract conversations from social media regarding COVID-19 and uncover several issues related to the disease. The method was based on a Linear Discriminant Analysis (LDA) Topic model and Gibbs sampling in order to perform semantic discovery and extraction of COVID-19–related comments. In another approach [15], an NLP model that relies on Bidirectional Encoder Representations from Transformers (BERT) is proposed, named AlBERTo. AlBERTo is focused on social media-related language and is used for performing sentiment analysis on data. The results of the aforementioned analysis regarded subjectivity, polarity and irony detection on Italian tweets. NLP can also be used so as to detect fake news and misinformation, as illustrated in [16], [17]. In another recent paper [18], the authors have implemented a multi-label classifier that is able to group conspiratorial content, and have tested it by searching YouTube for misinformation regarding COVID-19. This approach is based on transfer learning pre-trained models that are used to train the classifier, which is then applied to several YouTube comments. In a completely different approach [19], NLP is utilized for retrosynthesis tasks, which reflects the generation of organic compounds. In that case, the authors use text-like representation of chemical reactions (Simplified Molecular-Input Line-Entry

System) and an NLP neural network transformer architecture in order to predict the retrosynthesis of chemical compounds. NLP can also be used in text-to-speech implementations such as the one in [20]. In that paper, an Artificial Neural Network (ANN) is developed so as to perform the aforementioned task (i.e., speech synthesis, which is a quite common element that is usually utilized in several NLP-related operations.

C. Data Cleaning & NLP

As mentioned above, NLP methods can be used to deal with a variety of challenges in several research domains, including data cleaning. There have recently been a few research initiatives that make an effort to propose a way to integrate NLP in data cleaning. More specifically, the authors in [21] propose an NLP algorithm that is based on ANNs in order to encode character sequences. This method can only be used in the data preparation phase, while it has been tested in both English and Bulgarian. In another research [22], the authors propose Rotom framework so as to make use of NLP in data cleaning. The framework is capable of performing error detection and error correction based on Entity Matching (EM). However, the data that are used by this framework are stored in a relational database and have a certain format. DataCLUE [23] is another framework for data-centric NLP, where the authors claim that it can be used in data cleaning as well. The whole idea on which DataCLUE is based is the automation of processes that were previously done manually. The authors in [24] also suggest the use of NLP in order to automate data cleaning processes. Their NLP-based system attempts to simplify the product safety risk assessment process by automating several steps, including data cleaning. In that case, the authors compare their results to the ones that would have occurred by human experts, thus highlighting the need to automate previously manual processes.

Based on the performed research, it seems that currently there exist just a few applications that exploit NLP in successfully accomplishing data cleaning. Based on the literature, NLP can be applied in a variety of problems in order to automate them and solve them more efficiently. As a result, in this paper we propose a mechanism that makes use of NLP in order to automate rule-based data cleaning. In short, in this type of data cleaning there exist specific constraints, known as rules, for every feature. Those constraints clarify the characteristics that each field of the corresponding feature should have, in order to be considered reliable. If a field does not satisfy the aforementioned constraints, then the corresponding cleaning actions should take place [25]. This kind of data cleaning is a process that is mainly a manual one, since it tightly depends on rules that must be generated by data experts. Hence, the automation of the aforementioned process could contribute to decreasing the computational cost of data cleaning mechanisms, as well as increasing their efficacy.

III. Proposed Approach

The proposed mechanism was implemented in Python Programming Language [26] and utilizes MongoDB [27] in order to store all the collected and produced data. With regards to the data storage, a database with efficient performance in Create and Read operations should be chosen. As a result, MongoDB was utilized, since it is the most suitable choice according to [28]. The mechanism also provides several Application Programming Interface (APIs) in order to allow the interaction with other mechanisms and platforms. The

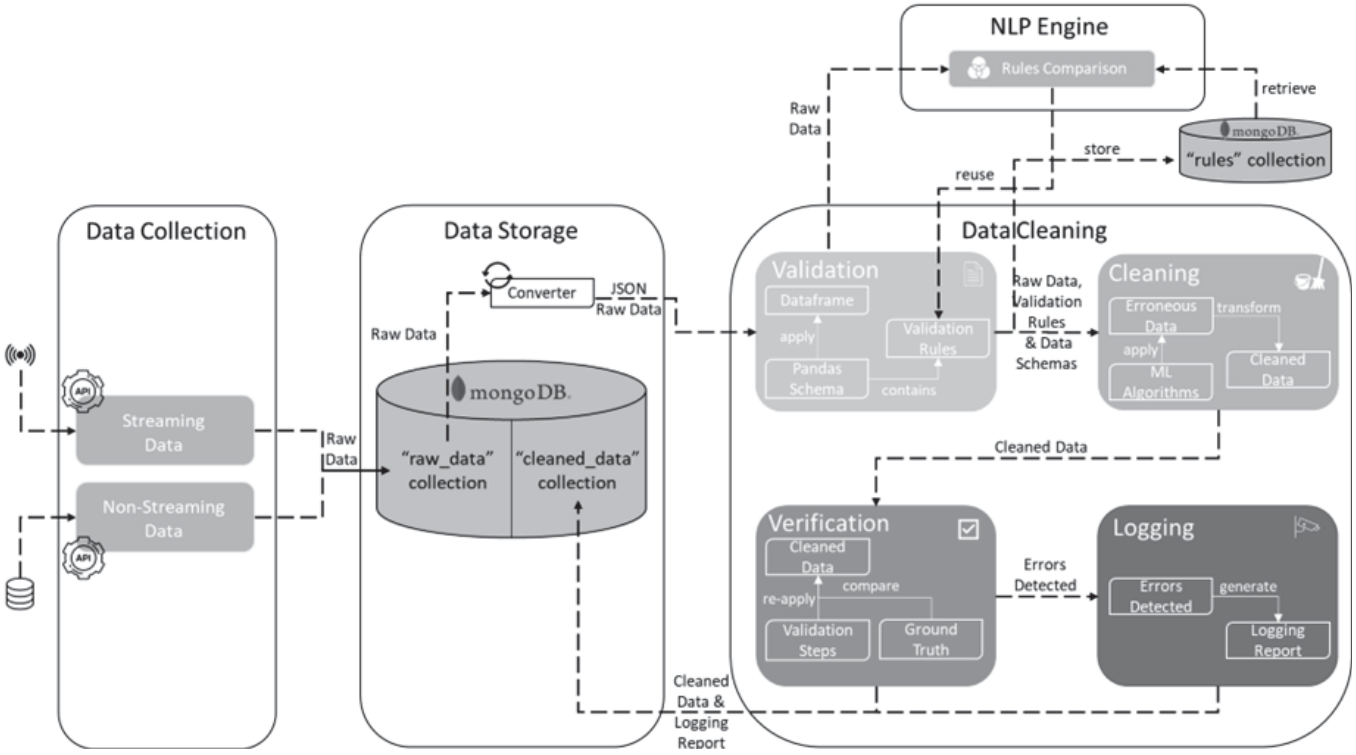

Fig. 1. Conceptual Architecture of Developed Mechanism

developed mechanism's conceptual architecture is shown in Fig. 1 and is described in deep detail in the paragraph below.

To begin with, the mechanism consists of three (3) discrete components. The first one is the "Data Collection" component that, as its name implies, is responsible for acquiring the data that are to be cleaned. The data may be streaming (e.g., deriving from wearable devices, sensors) or non-streaming (e.g., deriving from databases) and are collected by exploiting the corresponding APIs that have been constructed [29], [30]. The second component is the "Data Storage" component, which is responsible for storing the raw data that come into the mechanism into the "raw_data" collection. The third component is called "Data Cleaning", being in charge of performing all the necessary cleaning actions on the data so as to become qualified and reliable. To achieve that, the "Data Cleaning" component consists of four (4) layers, while it also contains the "NLP Engine" that can perform NLP techniques in order to assist the data cleaning process and maximize its performance.

In deeper detail, the first step of this component is called "Validation". In this step the mechanism identifies potential erroneous data by comparing them to specific rules that are included in certain forms called schemas. A rule is designed specifically for a data feature and determines the characteristics of this feature in order to be valid (e.g., range of values, type of variable, measurement unit). Those rules could be generated manually for every dataset that the mechanism receives by a corresponding data expert or can automatically be generated (the dataset and its features' characteristics are used as the ground truth), by making use of the algorithm shown in Fig. 2. In both scenarios, the extracted rules are stored locally. However, the aforementioned process can be avoided with the help of the "NLP Engine". The "NLP Engine" is responsible for comparing a newly uploaded dataset (whose corresponding data schema is not available) to data schemas that were previously generated by the mechanism. Every feature of the new dataset is compared to the features of the old datasets regarding the syntactic and the semantic similarity of the features, as well as the characteristics of the features. In order to compute the syntactic similarity, a variety of techniques are utilized, including Jaccard similarity [31] and Cosine similarity [32]. Regarding the semantic similarity, a more complex technique is used that is based on Transformers [33]. Transformers are a type of Deep Learning models that differentially weights the input data. It is worth mentioning that the semantic similarity is calculated and taken into consideration only if the syntactic similarity is not satisfactory, since calculating the semantic similarity has higher computational cost and is redundant in the cases where the syntactic similarity is high enough. If the "NLP Engine" manages to find a high similarity (syntactic and/or semantic) between the new feature and an old feature (exceeding the set threshold of 70%), then the corresponding rule of the old feature is used in order to validate the new feature, thus eliminating the need for generating a brand-new rule (or even a data schema). In the case that the similarity of the new feature to an old one is not satisfactory, then either the rules of the most similar feature are selected, or a data expect should intervene in order to create new rules for this specific feature/dataset.

The second step of the "Data Cleaning" component is called "Cleaning" and is responsible for correcting all the erroneous data that were identified during the validation process by applying a variety of techniques. The mechanism is capable of dealing with a wide range of data inconsistencies, such as missing values, outliers, duplicate records and syntactic errors. Depending on the inconsistency, the mechanism is capable of removing such erroneous values, predicting them by using ML algorithms and/or fixing it by utilizing NLP.

Sequentially, in the third step namely "Verification", the cleaned data are once again checked to ensure that they are indeed "clean" and no errors still exist, based on the corresponding schema. During this step, a score is also generated that represents the success rate of the whole process.

Finally, in the fourth step called "Logging", a report (i.e., logging files) is generated that provides insights regarding all the actions that the "Cleaning" component performed so that the users of the mechanism are able to retrieve it in any given time. Both the cleaned data and the corresponding logging files are stored in the "cleaned_data" collection for further use, if needed.

IV. Experimental results

The efficacy and efficiency of the proposed mechanism were evaluated by utilizing three (3) datasets from Kaggle, all of them referring to the healthcare domain. The inconsistencies of each dataset varied so that it could be ensured that the mechanism can eradicate all the data inconsistencies. Moreover, some of the features of the data had high similarity between them in order to also test the functionality of the "NLP Engine". The latter is one of the core novelties of the proposed mechanism, since it promotes the reusability of validation rules and schemas when performing data cleaning on unknown data, thus enhancing the

```
training_feature, feature //the field used to create the validation rule
characteristics_list, characteristic //list containing all possible characteristics of a feature (e.g., type, range etc.)
validation_rule_list, validation //list that contains all the validation rules generated for a feature
while characteristics_list has next, do
    for each characteristic in characteristics, do
        if feature.characteristic = characteristic
            add characteristic in validation
    end
end
```

Fig. 2. Algorithm used for generating validation rules

performance of the whole mechanism. As for the contents of the datasets, the first dataset is patients suffering from diabetes [34] and includes 101766 records and 50 features. The second one refers to patients who may deal with a cardiovascular disease [35], containing 299 records and 12 features. The final one refers to stroke-related patients [36], consisted of 5110 records and 13 features. The ground truth (i.e., the complete dataset without any errors) of those datasets was available and, as a result, it was possible to evaluate the performance of the mechanism.

To begin the whole experimental process, the mechanism firstly collected the aforementioned data using the "Data Collection" component. The latter were then stored in the "raw_data" collection through the "Data Storage" component and then retrieved by the "Data Cleaning" component, so that the rule-based data cleaning could take place. Then, the cleaned data, alongside with the generated logging report were stored in the "cleaned_data" collection. The performance was measured in terms of the rate of fields that were successfully cleaned, which in the scope of this paper is referred to as "success rate". The datasets generated by the mechanism were identical to the ones anticipated based on the ground truth, thus the success rate of the mechanism was one hundred percent (100%).

Indicative examples of the mechanism's usage on the "Diabetes" (Dataset I), "Heart" (Dataset II) and "Stroke" (Dataset III) datasets are respectively shown in Fig. 3., Fig. 4. and Fig. 5. In each of the three (3) figures shown below, there exist three (3) JSON files. The: (a) JSON file that is an indicative example of the corresponding dataset, the (b) JSON file that is an example of the corresponding logging report that is generated and includes information about the erroneous data and the actions performed, and the (c) JSON file that is an instance of the results of the mechanism that includes information about the dataset, the performed cleaning actions, and the success rate of the mechanism.

As shown below, the mechanism is capable of dealing with a variety of inconsistencies that occur in the data, referring to syntactic errors, missing values, duplicate records as well as outliers. To deal with such errors it is capable of performing a variety of actions such as the removal of those values, or their prediction based on ML algorithms such as KNN. Regarding the usage of the "NLP Engine", this took place in certain features of the datasets that were, on purpose, similar to each other.


```
{
    "patient_nbr": "100654011",
    "race": "Caucasian",
    "gender":"Female",
    "age":"[70-80)",
    "weight":"?"
}
```
(a)

```
{
    "info": "row: 0, column: weight: ? is not string.",
    "logs_id": "DUMHMSG9"
}
```
(b)

```
{
    "initial_fields": 5088300,
    "initial_columns": 50,
    "initial_rows": 101766,
    "number_of_errors": 192849,
    "number_of_columns_dropped": 0,
    "number_of_rows_dropped": 0,
    "number_of_duplicates": 0,
    "number_of_outliers_detected": 0,
    "number_of_fields_replaced_with_mode": 50000,
    "number_of_columns_knn_was_applied": 1,
    "success_rate": 100,
    "logs_id": "GH53SKX0",
    "dataset": "diabetes"
}
```
(c)

Fig. 3. (a): JSON format of Raw Dataset I, (b): Indicative example of report file, (c): Results of cleaned Dataset I


```
{
    "age":75,
    "anaemia":0,
    "creatinine_phosphokinase":582,
    "diabetes":0
}
```
(a)

```
{
    "info": "OUTLIER",
    "data":
    [
        {
            "age": 75
        }
    ]
}
```
(b)

```
{
    "initial_fields": 3887,
    "initial_columns": 13,
    "initial_rows": 299,
    "number_of_errors": 19,
    "number_of_columns_dropped": 0,
    "number_of_rows_dropped": 19,
    "number_of_duplicates": 0,
    "number_of_outliers_detected": 19,
    "number_of_fields_replaced_with_mode": 0,
    "number_of_columns_knn_was_applied": 0,
    "success_rate": 100,
    "logs_id": "JEWGW4DW",
    "dataset": "heart"
}
```
(c)

Fig. 4. (a): JSON format of Raw Dataset II, (b): Indicative example of report file, (c): Results of cleaned Dataset II


```
{
    "id":9046,
    "gender":"Male",
    "age":67,
    "hypertension":0
}
```
(a)

```
{
    "info": "DUPLICATE",
    "data":
    [
        {
            "id": 9046
        }
    ]
}
```
(b)

```
{
    "initial_fields": 61320,
    "initial_columns": 12,
    "initial_rows": 5110,
    "number_of_errors": 205,
    "number_of_columns_dropped": 0,
    "number_of_rows_dropped": 202,
    "number_of_duplicates": 1,
    "number_of_outliers_detected": 0,
    "number_of_fields_replaced_with_mode": 0,
    "number_of_columns_knn_was_applied": 3,
    "success_rate": 100,
    "logs_id": "ME7MEC4S",
    "dataset": "stroke"
}
```
(c)

Fig. 5. (a): JSON format of Raw Dataset III, (b): Indicative example of report file, (c): Results of cleaned Dataset III

For instance, the feature "Gender" was present in all the three (3) datasets, so the "NLP Engine" reused the rule regarding this feature that was generated for the "Diabetes" dataset, thus avoiding the generation of a new rule. In another example, the mechanism recognized the similarity between the features "patient_nbr" of the "Diabetes" dataset and the "id" feature of the "Heart" dataset and used the same validation rule. More specifically, even if the name of the feature was different, the mechanism found out that both features were some kind of unique identification code, so it utilized the same validation rule. In that case, there are two (2) validation rules. The first one states that every field of the feature should be unique and the second one states that it should be an integer number. A complete list of the features whose validation rules were reused are shown in Table I. The above showcases the fact that for each feature there can be many validation rules. The criterion on which the "NLP Engine" was based, was the similarity of both the name of the features (known and unknown ones) and the context of the corresponding values. If the percentage of similarity was high enough, approximately at 70% (i.e., set threshold), then the suggested validation rule was utilized on the unknown feature. That way the need of generating a new validation rule or even a complete data schema was being avoided, whilst both time and computational resources were conserved, since the generation of a new rule would need the intervention of a data expert.

TABLE I. FEATURES LIST OF REUSED VALIDATION RULES

Feature	***Feature(s) that Reused the Validation Rule(s)***	***Feature Description***
"Gender"	"gender" ("Heart" dataset), "sex" ("Stroke" dataset)	Categorical variable that corresponds to the gender of the observation
"patient_nbr"	"id" ("Heart" dataset), "encounter_id" ("Diabetes" dataset)	Integer number that is a unique identifier of the observation
"smoking"	"smoking_status" ("Stroke" dataset)	Binary variable that indicates the smoking status of an observation
"high_blood_pressure"	"hypertension" ("Stroke" dataset)	Binary variable that indicates whether an observation has hypertension
"time_in_hospital"	"time" ("Stroke" dataset)	Integer variable that corresponds to the number of days
"avg_glucose_level"	"glu_serum" ("Diabetes" dataset)	Float number that corresponds to the glucose level

A summarization of the data inconsistencies that were found and corrected is shown in Table II. Based on the findings of the experiments, the mechanism was able to recognize and fix all the inconsistencies that were present in the data, since the success rate was found to be 100%. This percentage was measured in the "Verification" step, where the ground truth was compared to the cleaned data generate by the mechanism, by making use of a particular syntactic similarity algorithm. What is more, the "NLP Engine" that was proposed in this approach assisted in the reusability of validation rules and data schemas, thus preserving time, and minimizing computing costs. Regarding the actions applied on the data, those were based on the corresponding input data so as to ensure the best cleaning results possible and were not incidental.

TABLE II. SUMMARIZATION OF DATA INCONSISTENCIES

Dataset	***"Diabetes"***	***"Heart"***	***"Stroke"***
Records	101766	299	5110
Features	50	13	12
Syntactic Errors	10000	12	100
Missing Values	182849	0	104
Outliers	0	19	0
Duplicates	0	0	1

V. CONCLUSIONS

Nowadays there exist a plethora of data sources. Those sources do not have any control regarding the reliability and the quality of the underlying data. Consequently, the need for data cleaning mechanisms is imperative. Moreover, the exponential growth in the fields of AI and other subfields in ML and NLP has allowed the utilization of methods and techniques from this fields to other fields in the Computer Science. Having said that, this paper presented the main concepts of NLP and Data Cleaning. It also referred to the current developments regarding the application of NLP in data cleaning, which are currently at an early stage. Moreover, in the current paper a data cleaning mechanism was developed and illustrated that combines principles from both the Data Cleaning and the NLP domains whilst it is also capable of performing more complex computations and actions than the other mechanisms that have been recently proposed. The mechanism was tested on health-related datasets, achieved 100% success rate, whilst it minimized the execution time, and the computational costs of the data cleaning tasks by utilizing the "NLP Engine", since it enabled the reusage of validation rules and data schemas, as explained in Section III.

Considering abovementioned, we expect to examine the developed mechanism by making use of datasets from additional areas except for healthcare (e.g., environment, transportations). Moreover, we hope to generalize the proposed mechanism in order to be self-adopted, regardless of the dataset's domain [37]. We also aim to add more ML algorithms for missing values' prediction, so that the mechanism is able to choose the most proper one in each case. Furthermore, we hope to stress the limits of the "NLP Engine" by using even more complex data and improve it, if necessary. We also have the intention of applying data anonymization on the data, thus addressing data privacy issues as well [38]. Last but not least, one of our main goals is to evaluate the applicability and the efficiency of the developed data cleaning mechanism in different existing healthcare platforms, referring to the CrowdHEALTH platform that introduces a new paradigm of Holistic Health Records (HHRs) that include all health determinants defining a person's health status by using Big Data management mechanisms [39], as well as the beHEALTHIER platform that constructs health policies out of data of collective knowledge, being managed through a variety of data management methods [40].

ACKNOWLEDGMENT

The research leading to the results presented in this paper has received funding from the European Union's funded Project INFINITECH under Grant Agreement No. 856632, and the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation under the call RESEARCH-CREATE-INNOVATE (project code: BeHEALTHIER-T2EDK-04207).

REFERENCES

[1] Statista - Data usage in marketing and advertising - Statistics & Facts, https://www.statista.com/topics/4654/data-usage-in-marketing-and-advertising/#dossierKeyfigures, last accessed: October 2022.

[2] Statista - Big data - Statistics & Facts, https://www.statista.com/topics/1464/big-data/#dossierKeyfigures, last accessed: October 2022.

[3] Statista - Revenues from the natural language processing (NLP) market worldwide from 2017 to 2025, https://www.statista.com/statistics/607891/worldwide-natural-language-processing-market-revenues/, last accessed: October 2022.

[4] Mavrogiorgou, A., et. al. (2019). Analyzing data and data sources towards a unified approach for ensuring end-to-end data and data sources quality in healthcare 4.0. Computer methods and programs in biomedicine, 181, 104967.

[5] Chu, X., et. al. (2016). Data cleaning: Overview and emerging challenges. In Proceedings of the 2016 international conference on management of data (pp. 2201-2206).

[6] Mavrogiorgou, A., et. al. (2021). An Optimized KDD Process for Collecting and Processing Ingested and Streaming Healthcare Data. In 2021 12th International Conference on Information and Communication Systems (ICICS) (pp. 49-56). IEEE.

[7] Wang, S., et. al. (2021). Short-term wind power prediction based on multidimensional data cleaning and feature reconfiguration. Applied Energy, 292, 116851.

[8] Müller, H., et. al (2021). From papers to practice: the openclean open-source data cleaning library. Proceedings of the VLDB Endowment, 14(12), 2763-2766.

[9] Chen, Z., et. al. (2022). An adaptive data cleaning framework: a case study of the water quality monitoring system in China. Hydrological Sciences Journal, 1-16.

[10] Zheng, Z., et. al. (2021). Discovery and contextual data cleaning with ontology functional dependencies. arXiv preprint arXiv:2105.08105.

[11] Guan, D., et. al. (2021). Abnormal Data Identification and Cleaning Method of Wind Turbine Based on Multi-model Fusion. In 2021 China Automation Congress (CAC) (pp. 5223-5227). IEEE.

[12] Mavrogiorgos, K., et. al. (2022). A Multi-layer Approach for Data Cleaning in the Healthcare Domain. In 2022 The 8th International Conference on Computing and Data Engineering (pp. 22-28).

[13] Otter, D. W., et. al. (2020). A survey of the usages of deep learning for natural language processing. IEEE transactions on neural networks and learning systems, 32(2), 604-624.

[14] Jelodar, H., et. al. (2020). Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. IEEE Journal of Biomedical and Health Informatics, 24(10), 2733-2742.

[15] Polignano, M., et. al. (2019). Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In 6th Italian Conference on Computational Linguistics, CLiC-it 2019 (Vol. 2481, pp. 1-6). CEUR.

[16] Busioc, C., et. al. (2020). A Literature Review of NLP Approaches to Fake News Detection and Their Applicability to RomanianLanguage News Analysis. Revista Transilvania, (10).

[17] Hamid, A., et. al. (2020). Fake news detection in social media using graph neural networks and NLP techniques: A COVID-19 use-case. arXiv preprint arXiv:2012.07517.

[18] Serrano, J. C. M., et. al. (2020). NLP-based feature extraction for the detection of COVID-19 misinformation videos on YouTube. In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020.

[19] Tetko, I. V., et. al. (2020). State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. Nature communications, 11(1), 1-11.

[20] Adam, E. E. B. (2020). Deep learning based NLP techniques in text to speech synthesis for communication recognition. Journal of Soft Computing Paradigm (JSCP), 2(04), 209-215.

[21] Marinov, M., & Efremov, A. (2019). Representing character sequences as sets: A simple and intuitive string encoding algorithm for NLP data cleaning. In 2019 IEEE International Conference on Advanced Scientific Computing (ICASC) (pp. 1-6). IEEE.

[22] Miao, Z., et. al. (2021). Rotom: A meta-learned data augmentation framework for entity matching, data cleaning, text classification, and beyond. In Proceedings of the 2021 International Conference on Management of Data (pp. 1303-1316).

[23] Xu, L., et. al. (2021). DataCLUE: A Benchmark Suite for Data-centric NLP. arXiv preprint arXiv:2111.08647.

[24] Chu, X., et. al. (2016). Data cleaning: Overview and emerging challenges. In Proceedings of the 2016 international conference on management of data (pp. 2201-2206).

[25] Hellwig, M., et. al. (2021). NLP for Product Safety Risk Assessment. In International Conference on Intelligent Systems Design and Applications (pp. 266-276). Springer, Cham.

[26] Python Programming Language - Python, https://www.python.org, last accessed: October 2022.

[27] MongoDB – MongoDB, https://www.mongodb.com, last accessed: October 2022.

[28] Mavrogiorgos, K., et. al. (2021). A Comparative Study of MongoDB, ArangoDB and CouchDB for Big Data Storage. In 2021 5th International Conference on Cloud and Big Data Computing (ICCBDC) (pp. 8-14).

[29] Perakis, K., et. al. (2019). Data Sources and Gateways: Design and Open Specification. Acta Informatica Medica, 27(5), 341.

[30] Mavrogiorgou, A., et. al. (2020). A plug 'n'play approach for dynamic data acquisition from heterogeneous IoT medical devices of unknown nature. Evolving Systems, 11(2), 269-289.

[31] Jalal, A. et. al. (2022). A web content mining application for detecting relevant pages using Jaccard similarity. International Journal of Electrical and Computer Engineering (IJECE), 12(6), 6461-6471.

[32] Henderi, H., et. al. (2021). Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient. Journal of Applied Data Sciences, 2(2).

[33] Ormerod, M., Del Rincón, J. M., & Devereux, B. (2021). Predicting Semantic Similarity Between Clinical Sentence Pairs Using Transformer Models: Evaluation and Representational Analysis. JMIR Medical Informatics, 9(5), e23099.

[34] Kaggle – Diabetes Dataset, https://www.kaggle.com/smit1212/diabetic-data-cleaning, last accessed: October 2022.

[35] Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC medical informatics and decision making, 20(1), 1-16.

[36] Kaggle – Stroke Prediction Dataset, https://www.kaggle.com/fedesoriano/stroke-prediction-dataset, last accessed: October 2022.

[37] Mavrogiorgou, A., et. al. (2021). Adjustable Data Cleaning Towards Extracting Statistical Information. In Public Health and Informatics (pp. 1013-1014). IOS Press.

[38] Kiourtis, A., et. al. (2018). Towards a secure semantic knowledge of healthcare data through structural ontological transformations. In Joint Conference on Knowledge-Based Software Engineering (pp. 178-188). Springer, Cham.

[39] Kyriazis, D., et. al. (2019). The CrowdHEALTH project and the hollistic health records: Collective wisdom driving public health policies. Acta Informatica Medica, 27(5), 369.

[40] Mavrogiorgou, A., et. al. (2021). beHEALTHIER: A microservices platform for analyzing and exploiting healthcare data. In 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS) (pp. 283-288). IEEE.