Exercise 2: Preprocessing Using Python

Datul, Michael James
College of Computer Studies and Engineering (CSE)
Jose Rizal University
Mandaluyong, Philippines
MichaelJames.Datul@my.jru.edu

***Abstract*— This activity will provide a deep understanding of the author about the importance of pre-processing and data cleaning in unclean data which is common in the environment especially when dealing with a vast amount of data. The introductory phase will give knowledge about the current state of the data produced in today's technology, this will introduce the importance of data preprocessing. The entire preprocessing procedure will provide a detailed explanation of how to deal with different data noises. With the use of Kaggle, a dataset that focuses on the reviews of visitors from Universal Studio is collected. Using this dataset, this paper will show the step-by-step process of cleaning the data by removing special characters, lowercasing, tokenization, and other essential processes that will be helpful to artificial intelligence experts, especially in natural language processing. Text data is one of the most common sources of raw data that can be transformed into valuable insights.**

Keywords- (*pre-processing, data cleaning, data, data noises, dataset, special character, lowercasing, tokenization, Artificial Intelligence, Natural Language Processing, raw data, insights)*

I. INTRODUCTION

Today's technology is widely used, due to its significance, and based on the latest estimation of statistica.com from the published article of explodingtopics.com, around 402.74 million terabytes of data are created every day which equates to around 147 zettabytes of data can be produced, and available per year. These numbers are proven and projected to increase constantly for the following years. [1] Despite the large number of data produced, most of them are considered dirty or uncleaned data, which is not entered or stored correctly. This is due to human error or insufficient standard process in extracting the said data. [2] This data cannot be used or is hard to manipulate if pre-processing is not performed very well.

As for this, data preprocessing is one of the most important phases in creating a model and is traditionally the first step in a successful data mining process. It is also widely adopted in different AI models which provides many advantages when performed well. It modifies the data into a form that can be easily understood by the machine or computer [3] to provide a data-driven output that is useful to humans in resolving certain problems. In connection with the data cleaning process, preprocessing techniques are one of the keys to eliminating the types of unclean data present in a dataset. These techniques will be performed in this paper specifically in a certain dataset that has unclean data, steps like lowercasing, removing special characters, and tokenization will be used to standardize the significant data that can be extracted from the dataset.

A clean and standardized set of data will be seen at the end of this paper ensuring that it can be used to perform any AI techniques. Although there are a lot of steps of cleaning processes available, techniques that will be shown are only limited to some of the cleaning processes because these are the only steps that are applicable in cleaning the given dataset.

II. RELATED WORKS

This section will discuss some applications of artificial intelligence and how the data was gathered and used in the environment.

Existence and big data production became the key factor in focusing on the process of data preprocessing and cleaning. Text data is part of this big data and one of the raw sources of data today, usage of sensors, and inserting information are just some ways of extracting this kind of data. This process takes as the first step before applying to the models that can be used in different sectors to have a standard set of data that can be read and understood by the model.

This became useful in the smart grid concept in promoting efficient operational performance and enhancing reliability and sustainability of power grids in supplying power independently. [4] Data that was collected by the sensors called smart sensors are processed and cleaned which helps in providing prices instantly, predicting possible errors on the grid and the demand that the market requires. [5] This predictive power of the grid became possible because of the data that was collected by the smart sensors.

The demand for clean data is increasing as it was also used in monitoring the condition of a certain structure to ensure its safety to the workers and availability to the users. Specifically in bridges, technologies are created to maintain their structural integrity by assessing different data that was gathered using different kinds of sensors. With the usage of the data captured and the application of artificial intelligence, structural health monitoring systems are produced which is a great help in evaluating the working location and enhances the maintenance of the infrastructure. [6] These provide ease in monitoring and evaluation of bridges and other public infrastructure that ensures safety to the users.

As artificial intelligence evolves and gets involved in different sectors present in our environment, it is crucial to provide clean and standard data to ensure the efficiency of each test result. The data format greatly affects the performance of each algorithm that will applied which is why

data pre-processing and cleaning dispute being the first, is also the most important phase in the data modeling process. It is the one who will set the tone of the entire process because of its power to control the performance. Aside from the data itself, it is also important to focus on the medium in which the data is being inserted and extracted. Sensors, input boxes, and other mediums need to be formatted in the sense that it is standard and uniform to avoid unclean data and provide ease to the experts.

III. METHODOLOGY

This section of this paper will focus on the dataset that the author used in practicing and applying the discussed steps in data cleaning and pre-processing. Tools that will be used are defined and will provide their role in completing the application of the process. Lastly, as data is already present, the second phase of natural language processing will be outlined.

A. Dataset Description

- *Source*

The dataset that will be used is gathered from the 'Kaggle' website which has the description of '*Reviews of Universal Studio'* which focuses on the text reviews sent by the visitors of the Universal Studio in its branches in Florida, Singapore, and Japan.

- *Columns*

The reviews are gathered by the 'Trip Advisor Website' and contain 6 columns. These columns are named 'reviewer' which is the name of the person who posted the review, 'rating' is the numeric rating experience of the reviewer which which ranges from 1 as unsatisfactory and 5 is described as satisfactory. The third column is the 'written_date' which specifies the date of the review when it was written followed by the 'title' column which is a short description of the entire review. The fifth column is the 'review_text', which is the actual review of the reviewer that shows the text-based experience, emotion, or comment of the visitor, lastly, the 'branch' column shows in which location the visitor addresses the review.

1	reviewer	rating	written_date	title	review_text	branch
	Kelly B	2	30-May-21	Universal is a complete Disaster - stick with Disney!	We went to Universal over Memorial Day weekend and it was a total train wreck. We waited to get in the parking lot for about forty minutes. We paid for prime parking to make up for all the wasted time. Then we paid extra for the express pass 2-park tickets only to be turned away and sent to guest services bc the app didnâ€™t show the bar code. The line at guest services took forever. They are clearly understaffed. We were sent to yet another guest services line because we had the express passes! Also took ages! We spent nearly 2 hours just trying to enter the park! When we shared this with Jackie at guest services she smirked, didnâ€™t	Universal Studios Florida

Figure 3.1 Dataset First Record

- *Rows*

Figure 3.1 shows the actual data that is present in the dataset which also shows the column name. The dataset contains a total of 50,905 records and as per checking each row does not contain any null values but the name is not in standardized form giving messy data.

- *Data Format and Language*

The dataset contains reviews and rows that were written in the English language, and it is downloaded as a CSV file making it easier to access and be checked using Microsoft Excel.

- *Existing Noises*

Based on the initial analysis of the existing data, it contains different noises similar to the discussion and example from the previous meeting. It did contain random noises that can affect the data in providing accurate and reliable insights [7] if the algorithm is applied.

B. Tools

- *Microsoft Excel*

After downloading the dataset 'Kaggle' website the author reviews the state of the data to have an overview of it in Microsoft Excel, it became a strategy to have ample time searching for another dataset if it does not satisfy the requirement needed. This may not be necessary in some cases as the website provides some information in their website. Still, in this paper, Microsoft Excel helps the author identify the best dataset that can be used throughout the process.

- Google Colab's

The next step after gathering the required data from the website, requires to be imported to Google Colab to manipulate and take advantage of the already existing libraries. Saving it to Google Drive works the best in this activity but Google Colab offers different importing options which can be an option for problems happening in mounting the drive.

- Python

Once the dataset is ready and mounted in Google Colab. It can be manipulated and cleaned using the programming language, and since the author will be working with artificial intelligence, python offers tons of packages that can be helpful in different modeling processes.

C. Data Cleaning Process

This section of the methodology shows the initial data cleaning process which will help in identifying the noises that the dataset contains. Figure 3.2 shows the overview of the different data cleaning processes used in this data set.

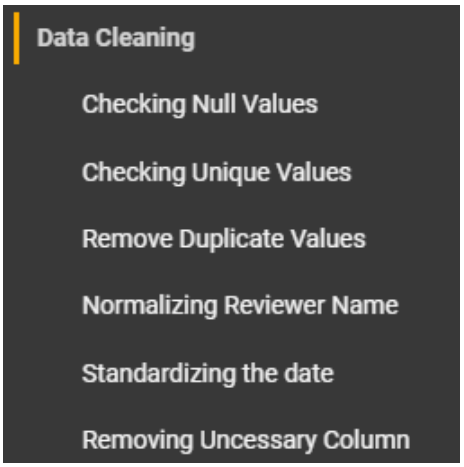

Figure 3.2 Data Cleaning Outline

A brief discussion of each technique will be provided in the fourth part of this paper as it will show the steps and exact numbers of each data. Each data cleaning procedure focuses on all of the rows in the dataset to ensure that the distribution of data is correct and will not affect the performance of a model if in a later part will be applied in this specific dataset.

D. *Pre-Processing Techniques*

In this part of the Google Colab's notebook, the author will provide the preliminary tasks that need to be completed to provide consistent and usable data that can be used in Artificial Intelligence, since the dataset is composed of almost text, this can be used and be taken advantage using Natural Language Processing.

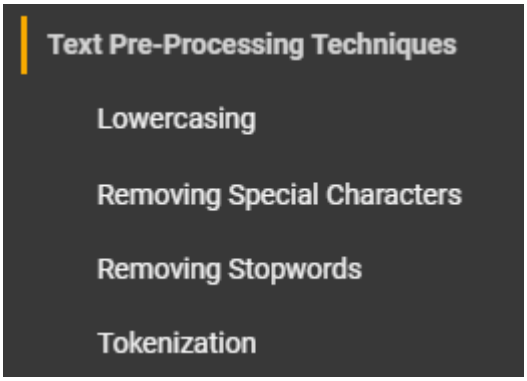

Figure 3.2 Pre-processing Outline

Figure 3.2 shows the different pre-processing techniques that were used in the selected dataset. It is composed of five techniques which are completed one by one to ensure that the final output will provide an essential insight. It is important to ensure that pre-processing techniques are applied to provide high-quality data that can be used in identifying patterns using different algorithms [8], these techniques will be discussed further in the later part of this activity.

After the data collection, it is important to analyze the data gathered to have an overview of what is needed to do next. The next section will show the actual cleaning and data processing techniques that were performed in the dataset.

IV. RESULTS

This section will show a series of images and a narrative explanation of the data cleaning and pre-processing techniques performed to create clean and reliable data that can be used in Natural Language Processing.

A. *Data Cleaning*

- Checking for Null Values

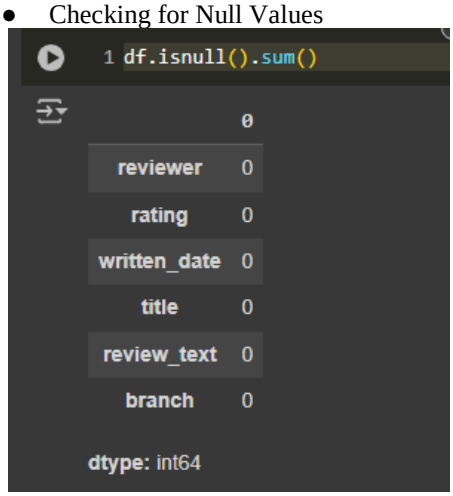

Figure 4.1 Null Value Information

Figure 4.1 shows the first step taken in cleaning the data in the dataset. Identifying null values is important as it has multiple advantages, it will prevent errors associated with null values which are the common errors that programmers encounter of having null values, and it reduces the error that may appear once you run the code. [9] In this dataset, it is visible that there are no null values that can be located in every row of every column which gives us a way to move on to the next part of the data-cleaning process.

- Checking for Unique Values

Another important part of data cleaning is knowing the unique values that are present in the dataset. This characteristic holds as the backbone in terms of the integrity of the dataset and helps the analyst or expert in producing accurate information and formulating data-driven decisions. [10]

```
1 for col in df.columns:
2   print(f"{col}: {df[col].unique().size} unique values")

reviewer: 44086 unique values
rating: 5 unique values
written_date: 4276 unique values
title: 37522 unique values
review_text: 50847 unique values
branch: 3 unique values
```

Figure 4.2 Checking the Unique Values

. As seen in Figure 4.2, the dataset contains 44,086 unique values for the reviewer which is the one who provides their review. The rating has 5 unique values which are rates 1 up to 5 showing the satisfactory rate of the reviewer. Written date only consists of 4,276 unique values as reviews can be done on the same date resulting in a lower count of unique values. The 'review_text' column shows 50,847 unique values since every review did not match the number of rows which is 50,905, it means that it has 57 rows that are duplicated.

```
num_duplicates = df.duplicated(subset=['review_text']).sum()
print("Duplicate Reviews:", num_duplicates)

# Remove rows with duplicate 'review_text' values
df_no_duplicates = df.drop_duplicates(subset=['review_text'])

# Verify the number of rows after removing duplicates
print("Update Rows:", df_no_duplicates.shape[0])

#Apply changes to the dataset
df = df_no_duplicates

Duplicate Reviews: 57
Update Rows: 50847
Updated DataFrame shape: (50847, 6)
```

Figure 4.3 Removing Duplicate Values

To address duplicate data, the removal of these is one of the better options because based on the numbers it will not greatly affect the the dataset. Figure 4.3 shows the process of removing duplicates in which the rows are now updated and the total number of rows becomes 50,847.

- Normalizing Reviewer Name

Another inconsistency that is discovered in the Universal Studio Review Dataset is the inconsistency in the reviewer name. It is important to normalize the data in a dataset to improve the organization and consistency of the entire dataset [11] which will also give a better view of each data.

	reviewer	...	...
0	Kelly B	50899	vinz20
1	Jon	50900	betty l
2	Nerdy P	50901	spoonos65
3	ran101278	50902	HeatSeekerWrexham_UK
4	tammies20132015	50903	sc_myinitial

Figure 4.4 Reviewer Name Actual Data

As depicted in Figure 4.4, it is noticeable that the names of the reviewers are inconsistently recorded. Some names are combined with numbers, and special characters and are in inconsistent cases. To address this problem figure 4.5 shows the transforming of each row into a lowercase.

```
df['reviewer'] = df['reviewer'].str.lower().str.replace('_', ' ')
```

	reviewer		
0	kelly b	50899	vinz20
1	jon	50900	betty l
2	nerdy p	50901	spoonos65
3	ran101278	50902	heatseekerwrexham uk
4	tammies20132015	50903	sc myinitial

Figure 4.4.1 Transforming the 'reviewer' column

With the use of the 'str.replace()' function as illustrated in Figure 4.4.1, changes are visible on names that are composed of underscore(-) signs which are replaced by the whitespace. The next process used is by removing the numbers that are connected to the letters.

```
import re

def remove_numbers(name):
  return re.sub(r'\d+', '', name).strip()

df['reviewer'] = df['reviewer'].apply(remove_numbers)
df['reviewer']
```

Figure 4.4.2 Function to remove the number

The figure above shows the creation of a function that separates the word and number to ensure a consistent name. The output below shows the application of the function to the dataset.

	reviewer
0	kelly b
1	jon
2	nerdy p
3	ran
4	tammies
...	...
50899	vinz
50900	betty l
50901	spoonos
50902	heatseekerwrexham uk
50903	sc myinitial

50847 rows × 1 columns

Figure 4.4.3 Normalized Name

Figure 4.4.3 depicts the effect of applying the function to the dataset which gives a much-accepted form of the reviewer as it has lowercase letters and does not contain any special characters. We can still separate the first and last names but considering that this paper will focus on the reviewer's review visit to Universal Studio this is more acceptable than the original data.

- Standardizing the Date

Dates are also one of the data that has different formats, because of this standardizing this data will help experts avoid confusion in data manipulation and provide efficient analysis of data.

```
df['written_date'] = pd.to_datetime(df['written_date']).dt.strftime('%m-%d-%Y')
df['written_date']
```

Figure 4.5 Date Standardization

Figure 4.5 shows a snippet of code that transforms the format of dates in the dataset into a certain format of numbered (month-day-year).

B. *Pre-Processing Techniques*

This section will focus on techniques that were used in the target text to transform it into standard and usable data that can be used in natural language processing. To have a comparison on the steps that will be done, figure 4.6 shows the original content of the 'review_text' column which consists

of upper and lowercase letters, special characters, and stopwords that are not usable in NLP.

```
df1 = df.copy()
df1['review_text']
```

	review_text
0	We went to Universal over Memorial Day weekend...
1	The food service is horrible. I'm not reviewin...
2	I booked this vacation mainly to ride Hagrid m...
3	When a person tries the test seat for the ride...
4	Ok, I can't stress enough to anyone and everyo...
...	...
50899	This is my first visit to a Universal Studio t...
50900	We finally visited Singapore's very first them...
50901	We visited during the first week of its 'soft ...
50902	We visited on the 3rd day of the 'soft' openin...
50903	My group managed to get the tickets for the 16...

Figure 4.6 Original 'review_text' column

- *Lowercasing*

To start the text pre-processing process, Figure 4.7 shows the code in which the transformation of a certain column is also visible.

```
df1['review_text'] = df1['review_text'].str.lower()
df1['review_text']
```

	review_text
0	we went to universal over memorial day weekend...
1	the food service is horrible. i'm not reviewin...
2	i booked this vacation mainly to ride hagrid m...
3	when a person tries the test seat for the ride...
4	ok, i can't stress enough to anyone and everyo...
...	...
50899	this is my first visit to a universal studio t...
50900	we finally visited singapore's very first them...
50901	we visited during the first week of its 'soft ...
50902	we visited on the 3rd day of the 'soft' openin...
50903	my group managed to get the tickets for the 16...

Figure 4.7 Original 'review_text' column

The use of the 'str.lower()' function makes the transformation of text possible as it checks all of the rows in the dataset and changes all of the scanned letters into lowercase if it is not in that letter is not in that state.

- *Removing Special Characters*

The next process in text pre-processing is by removing special characters. Figure 4.8 will show the entire snippet of codes as well as its effect on the record.

```
df1['review_text'] = df1['review_text'].apply(lambda x: re.sub(r'[^a-zA-Z0-9\s]', '', x))
df1['review_text'].iloc[0]
```

```
'we went to universal over memorial day weekend and it was a total train wreck we waited to get in the
parking lot for about forty minutes we paid for prime parking to make up for all the wasted time  then
we paid extra for the express pass 2park tickets only to be turned away and sent to guest services bc
the app didnt show the bar code the line at guest services took forever they are clearly understaffed
we were sent to yet another guest services line because we had the express passes also took ages we sp
ent nearly 2 hours just trying to enter the park when we shared this with jackie at guest services she
smirked didnt even apologize and was patronizing this would never happen at disney  once inside severa
l of the rides didnt work when they reopened they were backed up so the express line on some rides was
still a full hour wait and two hours without the express pass we also saw people jump over and sneak i
n to the express lanes and then convince the workers to just let them on if the...'
```

Figure 4.8 Special Characters Removal

With the use of the 're.sub()' function, it will automatically look for the characters that will not match the given set of characters that serve as the parameter. In this case, the parameter used is '[^a-zA-Z0-9\s]' which means everything that is not in the alphabet and number is considered a special character. When located it will be removed. The second line shows the output after applying the function to the first row.

- *Removing Stop Words*

Removing stop words is one of the most common steps as well in pre-processing text data as it will enhance the text's ability to be analyzed and provide a relevant way to help a model find patterns in the data.

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
```

Figure 4.9.0 Importing NLTK package

In these lines of code, nltk is imported which is the library used in natural language processing. The second line loads the existing common stopwords which are already collected. The fourth line finally converts and retrieves specific English stopwords which is in list form for easier and faster comparison.

```
def remove_stopwords(text):
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    return ' '.join(filtered_words)
```

Figure 4.9.1 Creating 'remove_stopword' function

Figure 4.9.1 is essential because it will be the function that will be used to automatically locate and remove the stopwords that are present in each row of the 'review_text' column. Each text that will be fed to the function will be separated using the 'split()' function based on whitespaces in each. After splitting the text, the 'filtered_words' variable for a list is created which stores all of the non-stop word texts and will ensure that the text that is going to be stored is in lowercase by the use of the '.lower()' function. Finally using the '.join()' function, all text that has been stored in filtered words is joined into a single string.

The figure below will illustrate the implementation of the created function in removing the stop words. Alongside it is the result that it has created in which the sentence became shorter and somehow incomplete because words without meaning are removed.

Figure 4.9.8 Aplying 'remove_stopword' function

- *Tokenization*

The last step that was used in this paper is tokenization which the string that was cleaned and combined from the start of the pre-processing will be converted into a smaller group of text called tokens. [12] The following images will show the process taken in adopting and applying tokenization in the selected dataset.

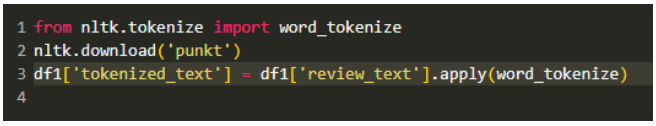

Figure 4.10.0 Importing Tokenization Packages

Python has a ready-made package that can be used in performing different tasks related to natural language processing, one of which is tokenization which tokenizes strings that are fed into. In this code, the 'apply' function is responsible for tokenizing the 'review_text' column.

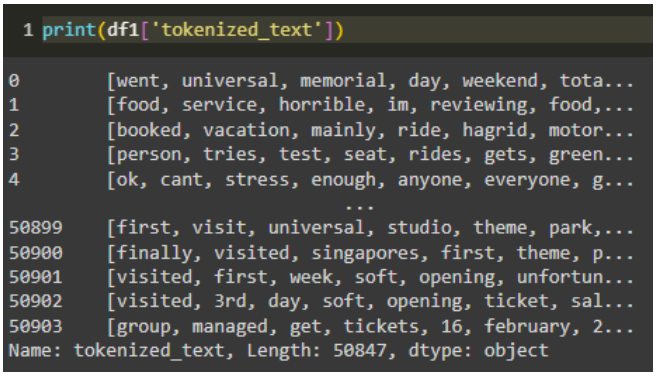

Figure 4.10.1 Tokenized 'review_text' column

Showing in figure 4.10.1 is the final result of the pre-processing procedure that was completed.

V. Discussion

The final output of the activity in the Google Colab notebook shows the tokenization of each relevant text in the dataset. Relating to the main objective of this paper which is to perform text-processing techniques that will allow the data to be used in natural language processing it is also important to maximize the dataset by performing several data cleaning processes for an efficient workflow.

Provided that data that are available on the internet are not clean and already usable, analyzing and getting rid of every aspect that may affect the performance of any algorithm can be the best practice. This activity starts in the first phase of natural language processing which is gathering the specific data that will be used. In this case, the researcher used a website that provides free datasets and chose a dataset that covers the review of visitors to one of the known entertainment studios, which is Universal Studio. The dataset is composed of columns and rows which are essential in handling records that are required to be analyzed and transformed according to the specification requirements. The gathered data is then imported into an environment that allows its manipulation, data that are present are then cleaned. Processes like removing null and duplicated values, and standardizing inconsistent data. This will help reduce errors that will be encountered throughout the process.

After completing the data cleaning process, pre-processing of the text is the next step. In this activity, lowercase the letters of each row that is selected becomes the preliminary step to create a consistent data format. It is followed by removing the special characters and stopwords that do not have significant value or meaning to the data. The steps ended in tokenizing each word in the rows which will help in analyzing word frequency and structure.

Conclusion

The steps and techniques that were shown and discussed during the class are essential information in understanding how natural language processing works. As it is already present in the late and today's technology, data cleaning and processing are a must-do routine to maintain the reliability and performance of every algorithm. Since NLP is widely used in different tasks like email filtering, translation, and other tasks, it is crucial to reduce the errors in providing results since most of us rely on its output.

And since technology keeps on evolving every time it is necessary to be informed about things that may keep the processes updated and correct including things like how we get data. It cannot be assumed a perfect set of data,

Acknowledgment

The author would like to express their sincere appreciation to Mr. Rodolfo Raga for providing information and his knowledge on creating this paper and sharing useful resources where to get data in this specific research.

References

[1] F. Duarte, "Amount of Data Created Daily (2024)," Exploding Topics, Mar. 16, 2023. https://explodingtopics.com/blog/data-generated-per-day

[2] S. Couwenbergh, "The Importance of Cleaning Dirty Data for Improved Operations and Customer Success," Validity, Aug. 24, 2022. https://www.validity.com/blog/dirty-data/

[3] G. Lawton, "Data Preprocessing: Definition, Key Steps and Concepts," SearchDataManagement, Jan. 2022. https://www.techtarget.com/searchdatamanagement/definition/data-preprocessing

[4] [5] Fanidhar Dewangan, A. Y. Abdelaziz, and Monalisa Biswal, "Load Forecasting Models in Smart Grid Using Smart Meter Information: A Review," vol. 16, no. 3, pp. 1404–1404, Jan. 2023, doi: https://doi.org/10.3390/en16031404.

[6] Z. He, W. Li, H. Salehi, H. Zhang, H. Zhou, and P. Jiao, "Integrated structural health monitoring in bridge engineering," Automation in Construction, vol. 136, p. 104168, Apr. 2022, doi: https://doi.org/10.1016/j.autcon.2022.104168.

[7] "What is noisy data? - Definition from WhatIs.com," SearchBusinessAnalytics. https://www.techtarget.com/searchbusinessanalytics/definition/noisy-data

[8] A. Aydin, "1 — Text Preprocessing Techniques for NLP," Medium, Oct. 04, 2023. https://ayselaydin.medium.com/1-text-preprocessing-techniques-for-nlp-37544483c007

[9] H. Ali, "Checking for is a good practice in programming for several reasons: Prevents Null-Reference Errors: Null-checking helps prevent null-reference errors, which occur when you try to access properties or

methods of an object that is or. Such errors can lead to program crashes.," Linkedin.com, Jan. 10, 2024. https://www.linkedin.com/pulse/why-checking-null-good-practice-haider-ali-56a1e#:~:text=Checking%20for%20null%20is%20a (accessed Aug. 19, 2024).

[10] Leon, "How to Find Unique Values in R," sqlpad.io, May 08, 2024. https://sqlpad.io/tutorial/unique-values/ (accessed Aug. 19, 2024)

[11] S. Morris, "Data Normalization: Definition, Importance, and Advantages," *coresignal.com*, Jun. 02, 2022. https://coresignal.com/blog/data-normalization/

[12] A. A. Awan, "What is Tokenization?," datacamp.com, Sep. 2023. https://www.datacamp.com/blog/what-is-tokenization (accessed Aug. 19, 2024).