Addressing the Challenges in Text Preprocessing and Text Representation

Sean Repti Miguel L. Ticzon  
*Computer Studies and Engineering*  
*Jose Rizal University*Mandaluyong City, Philippines  
seanreptimiguel.ticzon@my.jru.edu

***Abstract*—Preprocessing is one of the most critical stages in NLP, as it affects the results of such tasks as text analysis and categorization greatly. Typical operations include tokenization, removal of stop words, lemmatization, and reduction to lowercase; these steps sometimes affect quality depending on the task at hand. Different datasets require different preprocessing techniques; therefore, the best ones would have to be found by methodical experimentation. These are the overcome challenges: the constraints in tool expertise and the handling of noisy data, such as random email addresses and dubious URLs. The problems were solved by applying regular expressions and filtering methods for the cleaning up of datasets, hence having successfully eliminated undesired noise. Moreover, practical experience and good documentation acquainted people with Python tools like CountVectorizer and TfidfVectorizer, which helped improve understanding and application. The results indicated a significant noise reduction and improved efficacy in text representation. The process has underlined the importance of treating data with care and continuous learning of new technologies to ensure effective pretreatment and representation. It showed me how meticulous and detail-oriented real-world management of data is and just how important in-depth knowledge of specific details is to analyze and process data.**

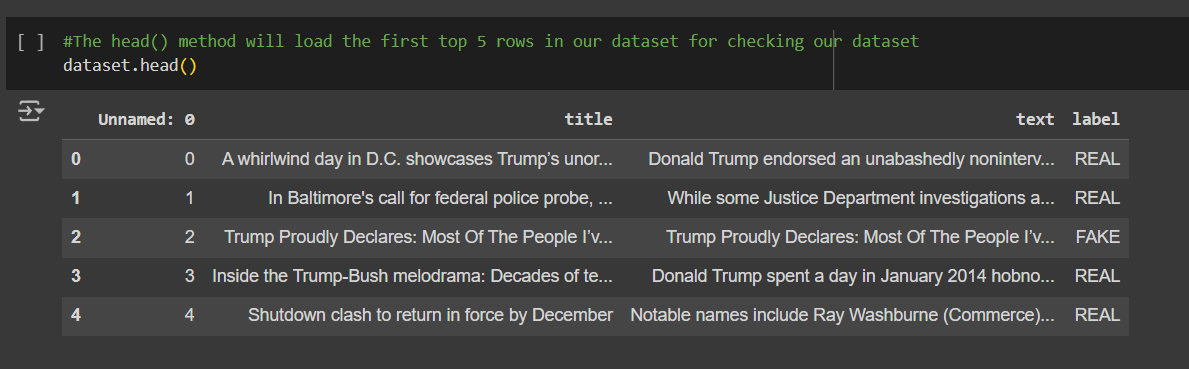
***Keywords—Text Preprocessing, NLP, Text Representation, Vectorization***

# Introduction

Text preprocessing is important phases in NLP, as most of the tasks related to text categorization and text analysis depend on it [1][2]. Tokenization, stop word removal, lemmatization, and lowercase conversion are some of the are common preprocessing techniques [3]. The selection and the order of the preprocessing procedures can affect the quality of the results in the text categorization tasks [1]. In fact, a varied combination of these preprocessing methods may prove the best on different datasets, which is a matter of systematic experimentation to find the real results [1][2]. Representation methods such as bag-of-words find extensive use in a variety of text mining applications including classification and clustering [4]. While preprocessing, the nature of the corpus and the exact NLP application has to be considered while choosing the preprocessing methods, since some datasets may require their own versions of processing to give optimal performance and results [3].

Doing text preprocessing and text representation sometimes does not go smoothly; there are always problems that we encounter when doing such tasks. Here are some of the examples that I encounter performing the said tasks:

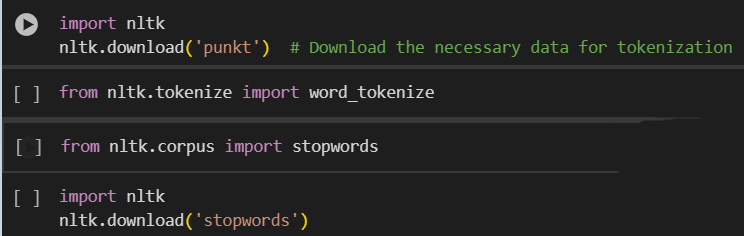
## Noises in the Dataset



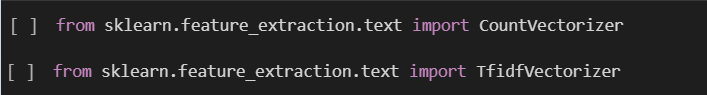
1. Noises in the dataset (Exer 2. About Text Preprocessing)

One challenge that I encountered there are some noises that I encountered while working on my dataset, there are random email addresses scattered around the text and there are fishy website links that could contain malware.

## Lack of Knowledge in Python and Tools used



1. Example tools used for text preprocessing (Exer 2. About Text Preprocessing)



1. Example tools for text representation (Exer 3. About Text Representation)

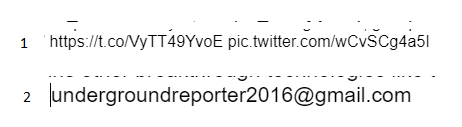
The second challenge I encountered was the lack of knowledge on each of the tools used for text preprocessing and text representation and sometimes messing up the syntax when trying out the code.

# Methodology

This section will cover how I handle the challenges that are encountered in making the text preprocessing and the text representation and what techniques are used to resolve those issues.

## Handling the Noise in the Dataset Text Preprocessing

The first step in processing the noise in the given dataset is its identification; this means text has to be scanned for undesired information such as e-mail addresses and URLs or a form taken by a suspicious pattern unrelated to the investigation. Regular expressions may be used or special filtering techniques to handle and cleanse the data.



1. Number 1 shows the link presented in the dataset and number 2 is the emails presented in the dataset.

This was achieved by building in a text preparation stage that removes unwanted components in the texts, such as email addresses and suspicious URLs. This step cleaned the dataset by filtering out unwanted noise before any further processing or analysis could be done.

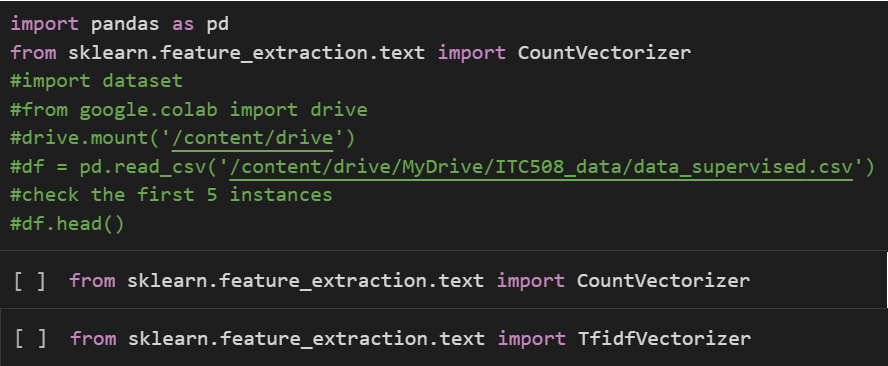


1. Shows a code that removes unwanted emails and links in the dataset.

That was to create a preparation stage of the text in such a way that it automatically removed unwanted components in the form of e-mail addresses and suspicious URLs from the text. It cleaned up the dataset by reducing the presence of noise before any kind of analysis could be performed.

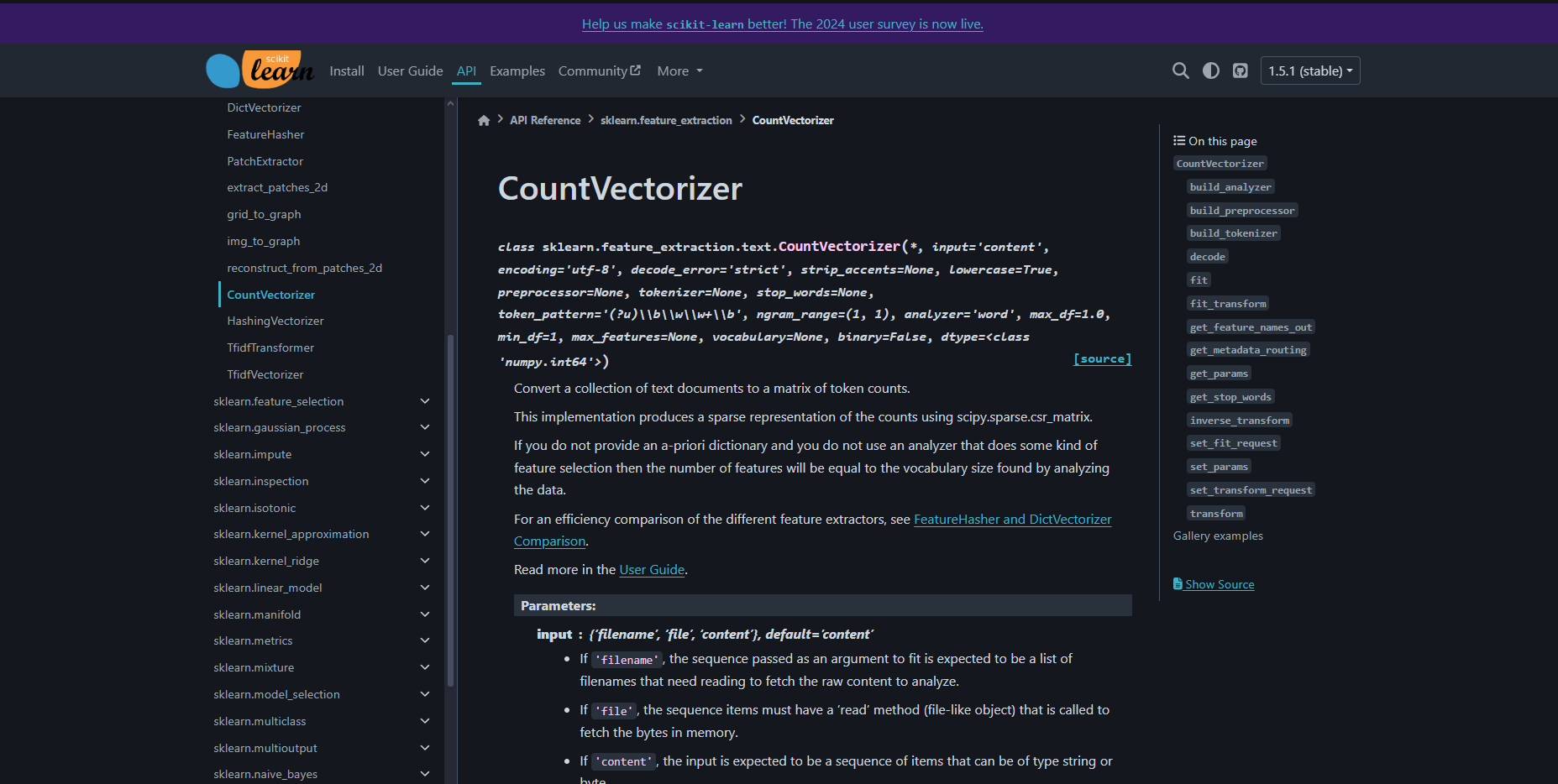
## Handling the Lack of Knowledge in Python Text Representation

First, I did scan and identify and familiarize myself with the new unfamiliar functions of Python in text representation that are used. Understanding these technologies effectively was required in the interest of understanding these technologies and making full potential use of them in my work on data processing.

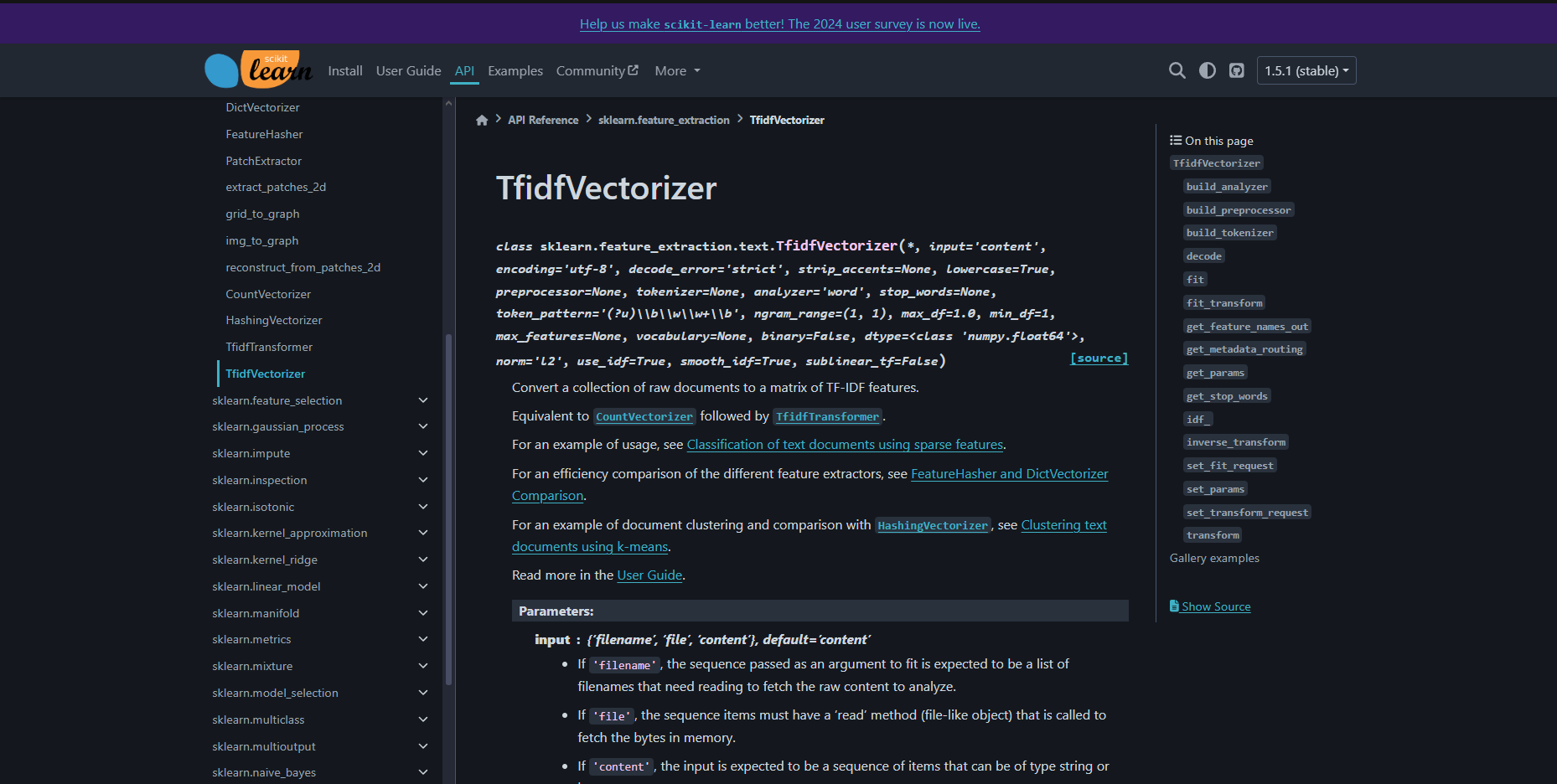


1. Showing the used tools in text representation and familiarizing it.

Familiarity with the tools and looking at how it is structured will help me to understand how it is syntax and what letters should be capitalized and not. This will help me on applying it easily throughout the task.



1. A documentation on how to use the CountVectorizer.



1. A documentation about the TfidfVectorizer

By the help of the documentations found online the tools used like the CountVectorize and TfidfVectorizer, I gained some knowledge on how the tools worked. For the CounterVectorize, a tool used for converting text into a matrix of token count, and as for the TfidfVectorizer does so into the matrix of term frequencies corrected for the so-called inverse document frequency. Since these tools and their functions are understood, one can make better use of them in text representation.

# Results

After I encounter and resolve the problems that are presented on the activities, these are the results that I gathered:

## Handling the Noise in the Dataset Text Preprocessing

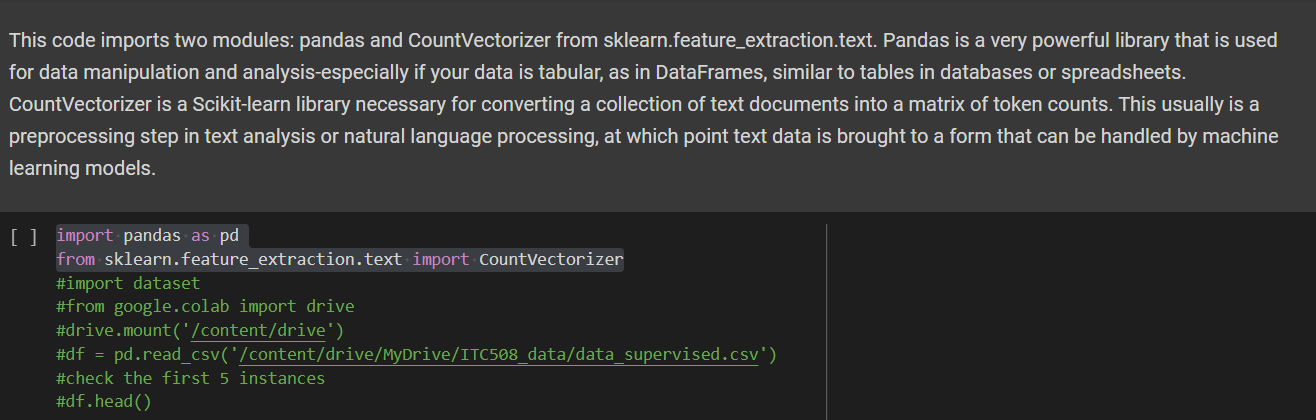
I have used a code specifically to rid the data of noise, regarding e-mails and links within the data. This was quite efficient in removing the noise and therefore reduced the dataset to a much clean dataset. After this preprocessing step, the findings showed a significant reduction of the text pertaining to the noise, of the text data.



1. The results of the removed email and link

## Handling the Lack of Knowledge in Python Text Representation

By researching for documentation online it help me to understand the tools used for text representation. By this it aided me on giving me knowledge on how it worked and how to use it effectively.



1. The results of the removed email and link

# Discussion

By doing this I realize that we need to more active and focus when it comes to handling data, because in the real world you better not make mistakes because you are handling real data that has sensitive information about the company or the stakeholders, it might affect the company that you are working with and the chance of you getting fired is high. This helped me on being more alert when working with datasets and also gain new knowledge on the tools used for data analysis for better efficiency and results.

# Conclusion

Upon completion of this research I gained knowledge on how important to be on being aware of what you are doing with the data you are handling, and being able to focus on this type of works is hard and overwhelming for me but with the right mindset and determination you will be able to complete any task that needs to be done. Being able to advance study and research for unfamiliarities in the future is good for us to easily understand incoming more challenges.

##### References

1. Y. HaCohen-Kerner, D. Miller, and Y. Yigal, “The influence of preprocessing on text classification using a bag-of-words representation,” PLoS ONE, vol. 15, no. 5, p. e0232525, May 2020, doi: 10.1371/journal.pone.0232525.
2. N. M. A. Rajesh and N. D. T. Hiwarkar, “Exploring Preprocessing Techniques for Natural LanguageText: A Comprehensive Study using Python Code,” International Journal of Engineering Technology and Management Sciences, vol. 7, no. 5, pp. 390–399, Jan. 2023, doi: 10.46647/ijetms.2023.v07i05.047.
3. C. P. Chai, “Comparison of text preprocessing methods,” Natural Language Engineering, vol. 29, no. 3, pp. 509–553, Jun. 2022, doi: 10.1017/s1351324922000213.
4. K. N. Singh, H. M. Devi, K. Robindro, and A. K. Mahanta, "A systematic study on textual data processing in text mining," in 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2019, pp. 24–29.