Text Data Preprocessing Documentation Using Regular Expressions

Erfan Mohammadzadeh

er.mohammadzadeh@gmail.com

e-mohammadzadeh

November 27, 2025

Contents

List of Tables

List of Figures

1 Introduction

To analyze datasets with models, we first need to prepare them. Data preparation or preprocessing refers to cleaning data from unwanted elements, such as removing extra words, symbols, or whitespace from the dataset. First, we explain how to read the dataset, then describe in detail the preprocessing steps performed on our datasets. The preprocessing section is divided into three main parts: Replace, Remove, and Expand.

The order of preprocessing steps performed corresponds to the sequence shown in Table 1.

Table 1: Order of Preprocessing Steps

#	**Preprocessing Steps**
1	Remove Null Values
2	Replace Slang words
3	Lowercase
4	Expand Contractions
5	Remove Stopwords
6	Replace URLs
7	Replace Email Addresses
8	Expand Mentions
9	Expand Hashtags
10	Remove Date
11	Replace Currencies
12	Replace Time
13	Remove Tags
14	Remove Cashtags
15	Remove Numbers
16	Remove Titles
17	Remove Punctuations
18	Replace Repeated Letters

2 Reading the Dataset

Initially, the path and names of the desired files are given to the program. Then the files are read sequentially and stored in a dictionary. The key part in the dictionary can be one of train, test, or dev. The value part in the dictionary is the file read corresponding to the key. Then preprocessing steps are applied to each value in this dictionary, and the result is clean data that is stored in another dictionary. In the following sections, each preprocessing step is explained in detail.

First, we divide the dataset columns into two parts based on their content: columns containing numerical data and columns containing textual data. Then preprocessing steps are applied to columns containing textual data.

3 Remove

3.1 Remove Null Values

The first section relates to removing certain characters, words, or special symbols.

If the dataset being examined in this section contains a cell with a null value (NaN[1]), that entire row is deleted, even if other cells in that row contain values. Also, for user information, the row numbers that have null cells and are to be deleted are printed in the output.

[1]Not a Number

English Stopwords.txt

```
until, their, further, can, each, yourself, it, myself, out, were, will,
but, where, ve, should've, above, your, again, up, me, those, an, very,
these, needn, having, he, under, how, m, between, its, about, had, this,
that'll, it's, they, hers, when, any, she, have, of, for, during, we,
while, below, the, she's, through, herself, before, if, you've, other,
now, that, own, off, ourselves, you're,  with, whom, and, has, into, in,
on, so, d, most, them, itself, same, down, you'll, is, should, because,
from, yours, then, themselves, such, i, over, there, being, or, at, been,
her, ours, did, here, a, his, are, you'd, y, just, why, than, yourselves,
our, be, which, am, theirs, doing, was, s, ll, after, more, what, re, my,
both, do, does, all, o, to, himself, as, you, who, only, by, too, t,
once, against, few, ma, him, some
```

Figure 1: List of Non-Meaningful Stopwords in English

3.2 Remove Stopwords

A text file created by the author is placed alongside the code in txt format. This file contains most English stopwords, with the difference that negative words were not included because we don't want negative words to be removed from the dataset. This text file contains 140 English stopwords. Figure 1 shows the list of English stopwords with the difference that negative words have been removed from this list.

3.3 Remove Date

The goal of this section is to extract and remove dates. The way dates are expressed can vary, so we've tried to extract most of the important formats for expressing dates. Table 2 shows short formats and Table 3 shows full formats that we can extract for dates from text. The various cases that can be considered for month names in the Gregorian calendar are also mentioned in Table 4. For extracting dates from text, we can use short and full formats separately, or combine these two formats. Therefore, the number of cases that can be extracted is large, and this results in high accuracy of the program during preprocessing.

Table 2: Short Date Formats

#	**Short Date**		
1	01/01/2024	01-01-2024	01.01.2024
2	27/5/24	27-5-24	27.5.24
3	01/2024	01-2024	01.2024

Table 3: Full Date Formats

Full Date			
#	First Part	Second Part	Third Part
1	Month-name[b]	of	Any number {4 digit}
2	Month-name	of	Any number {4 digit},
3	Any number {1-4 digit}	Month-name	
4	Any number {1-4 digit},	Month-name,	
5	Month-name	Any number {2 digit}{st, nd, rd, n-th}	Any number {1-4 digit}
6	Month-name,	Any number {2 digit}{st, nd, rd, n-th},	Any number {1-4 digit}
7	Month-name	Any number {1-4 digit}	Any number {2 digit}{st, nd, rd, n-th}
8	Month-name,	Any number {1-4 digit}	Any number {2 digit}{st, nd, rd, n-th},
9	Any number {1-4 digit}	Month-name	Any number {2 digit}{st, nd, rd, n-th}
10	Any number {1-4 digit},	Month-name,	Any number {2 digit}{st, nd, rd, n-th}

[b] Each month name (table 4) can be substituted

Table 4: Various Formats for Month Names

Month Names			
Jan	January	jan	january
Feb	February	feb	february
Mar	March	mar	march
Apr	April	apr	april
-	May	-	may
Jun	June	jun	june
Jul	July	jul	july
Aug	August	aug	august
Sept	September	sept	september
Oct	October	oct	october
Nov	November	nov	november
Dec	December	dec	december

3.4 Remove Tag

In this section, we identify and remove HTML opening and closing tags and other tags. For example, the following tags are removed in this section:

```
1: <head>
2: <url>
3: </body>
4: </div>
5: <img src="image.jpg">
```

3.5 Remove Cashtag

According to X[1] company's definition: "A cashtag is a unique and short symbol for a cryptocurrency or company that is preceded by a dollar sign" [1]. Cashtags are used in the X app. Now in this section, we intend to extract and remove these cashtags from the text if they exist. For example, the following cashtags and similar ones are removed in this section:

1 $\$GOOG \longmapsto$ Google company cashtag

2 $\$TSLA \longmapsto$ Tesla company cashtag

3 $\$PEP \longmapsto$ Pepsi company cashtag

4 $\$BTC \longmapsto$ Bitcoin cryptocurrency cashtag

5 $\$ETH \longmapsto$ Ethereum cryptocurrency cashtag

[1]Formerly Twitter

3.6 Remove Number

Removing numbers present in text is another preprocessing step. Numbers that are removed can have the following formats:

- Positive or negative numbers
- Fractional numbers
- Decimal numbers
- Numbers separated every three digits with **.** or **,** characters
- Scientific notation in the form A$E \pm$ B or A$e \pm$ B (where A and B are both numbers)

When cleaning text from numbers, we faced two challenges, each briefly mentioned below:

- Removing all numbers present in the text such that numbers within the structure
$$\rhd \{L/M/T/E\}_{counter} \lhd$$
remain untouched. To solve this problem, we first identify this structure, then extract the text between the two characters $\lhd$ and $\rhd$, and after that perform the operation of removing numbers. Finally, the extracted text is inserted back into its place (between the two characters).

- The second challenge is removing numbers such that numbers within Unicode characters are not deleted. When there is a non-English sentence or word in the dataset, there is a possibility of Unicode characters in that word or sentence. Unicode characters usually have the following structure: \u0000 \u10FFFF

After the letter ”u”, six characters can be written, where each character can be ”digits from zero to nine - lowercase English letters from a to f - and uppercase English letters from A to F”. Here are several examples of Unicode characters:

K\u00f6lsch $\longmapsto$ Kölsch

pur\u00e9ed $\longmapsto$ puréed

jalape\u00f1o $\longmapsto$ jalapeño

Gar\u00e7on $\longmapsto$ Garçon

Using commands, we first identify Unicode characters, then ignore six characters and remove the remaining numbers, thus solving the second challenge when removing numbers.

3.7 Remove Title

Removing titles used in text can also be part of preprocessing tasks, titles such as:
Mr. Mrs. Miss. Dr. Prof. Pres. (president) ...
that frequently occur in English texts. Seventeen English words were considered for this section, the complete list of which can be seen in Figure 2. For removing titles, these points were considered: first, case insensitivity; second, the presence or absence of a period after the title.

3.8 Remove Punctuation

Removing punctuation marks and additional visual symbols from text is also part of preprocessing steps. In this section, characters such as:

‘ - = [ ] \ ; ’ , . / ! @ # $ % ˆ& * ( ) _ + — : ” ¡ ¿ ?

are found and removed from the dataset.

In this section, there was also a challenge related to Unicode characters; a Unicode

English Titles

```
Mr, Ms, Mrs, Miss, Dr, Prof, Sir, Ma'am, Madam, Madame,
Rev,  # Reverend, used for Christian clergy
Fr,  # Father, used for Catholic priests
Sr,  # Sister, used for Catholic nuns
Capt,  # Captain, used in the military and for pilots
Gen,  # General, used in the military
Hon,  # Honorable, used for judges and certain politicians
Pres,  # President, used for presidents of companies or organizations
```

Figure 2: List of English Titles

character without the "\" symbol has no meaning and cannot be converted to the desired form. Therefore, when removing punctuation marks, we must be careful not to remove the "\" symbol in Unicode characters.

4 Replace

The second section relates to replacing certain characters, words, or special symbols with other words or unique keys.

4.1 Slang Words

The first task in the replacement section relates to replacing English slang words with their complete and formal form. In writing English text, to perform typing and sending messages as quickly as possible, shortened and slang words are commonly used. To increase model accuracy, we tried to identify these shortened and slang words to some extent and replace them with the complete word for each.

For this purpose, we made some modifications to a previously created file [2];

```
English Slang Words.json
{"07734": "hello",
"2day": "today",
"2ge4": "Together",
"2morrow": "tommorrow",
"4ever": "forever",
"Onoe": "Oh No",
"Ovr": "over",
"10q": "thank you",
"5n": "fine",
"absnt": "absent",
"bc": "because",
"c@": "cat",
"dw": "don’t worry",
...}
```

Figure 3: Sample of English Slang Words Along with Complete Form of Each

ultimately, the file contains a dictionary where the key is the abbreviated word and the value is the word in complete form. Figure 3 shows a sample of this file. To access the complete file, refer to the author’s GitHub.

4.2 Lowercase

One of the tasks performed on datasets is converting uppercase English letters to lowercase English letters.

4.3 Contraction

A file created by the author (available on GitHub) has JSON format. This file contains most contraction words in English. Figure 4 shows a sample of this file. The general format of the file is a dictionary where the key is the abbreviated word and the value is the complete form of that abbreviated word. In this preprocessing section, we replace abbreviated words present in the dataset with their complete form.

English Contractions.json

```
{"ain't": "are not",
"aren't": "are not",
"can't've": "cannot have",
"can't": "cannot",
"'cause": "because",
"could've": "could have",
"couldn't've": "could not have",
"couldn't": "could not",
"didn't": "did not",
"doesn't": "does not",
"don't": "do not",
"dunno": "do not know",
...}
```

Figure 4: Sample of English Contraction Words Along with Complete Form of Each

4.4 URL

The next task is extracting Uniform Resource Locators (URLs) and replacing them with a unique key that can later be used to retrieve the desired web address if needed. The key considered for this section has the following structure:

$$\triangleright L_{counter} \triangleleft$$

For example, the first web address is replaced with $\triangleright L1 \triangleleft$, the second web address with $\triangleright L2 \triangleleft$, and so on. The letter L is taken from the word Link. After the replacement operation, the web address along with its corresponding key is stored in a file. With this approach, the web address can be retrieved if needed.

4.5 Email Address

Next is extracting email addresses and replacing them with a unique key that can later be used to retrieve the desired email address if needed. For the email domain, no restrictions were considered, so we identify any email address with a valid username and domain. The key considered for this section has the following structure:

$$\rhd E_{counter} \lhd$$

For example, the first email address is replaced with $\rhd E1 \lhd$, the second email address with $\rhd E2 \lhd$, and so on. The letter E is taken from the word Email. After the replacement operation, the email address along with its corresponding key is stored in a file. With this approach, it can be retrieved if needed.

4.6 Currency

Another preprocessing step is extracting money and currency units and replacing them with a unique key that can later be used to retrieve the desired money if needed. The key considered for this section has the following structure:

$$\rhd M_{counter} \lhd$$

For example, the first currency unit is replaced with $\rhd M1 \lhd$, the second currency unit with $\rhd M2 \lhd$, and so on. The letter M is taken from the word Money. After the replacement operation, the currency unit along with its corresponding key is stored in a file. With this approach, it can be retrieved if needed. Examples of numbers and words that are identified as money and replaced with a specific key are mentioned below.

- Currency abbreviation symbols considered during extraction are mentioned in Table 5. Whether the symbol appears before or after the number, both cases are extracted from text.

Table 5: Currency Abbreviation Symbols of Countries

#	**Currency Symbol**	**Currency**	**Country**
1	$	Dollar	USA
2	€	Euro	Euro Member Countries
3	£	Pound Sterling	UK
4	¥	Yen	Japan
5	₹	Rupee	India
6	¢	Cent	USA

- Usually when writing money numerically, numbers are separated every three digits. Here the separator character can be period (.), comma (,), or single quotation (’).

- Usually when expressing money, terms like ”thousand”, ”million”, and ”billion” are used. In Table 6, we defined the unit counting system. For example, if in text the letter m or word Mill appears with a number (it doesn’t matter whether it appears to the right or left of the number), then we consider that letter or word as million and replace the number along with that letter or word with a special key.

By combining the three previous items, various cases for extracting money from

Table 6: Money Counting System

#	**Unit Symbol**	**Unit**
1	K k	Thousand
2	M, Mill, Million m, mill, million	Million
3	B, Bill, Billion b, bill, billion	Billion
4	T, Trill,Trillion t, trill, trillion	Trillion

text are obtained. Here are several examples:

€200 $800.00 $1,222 8$ $60k £300,75 ¢400'85 €600.90m €1000.10mill 2000.20Million€ $1800'12b

4.7 Time

Extracting and replacing time (hours) is another preprocessing step. The unique key considered for this section has the following structure:

$$\triangleright T_{counter} \triangleleft$$

For example, the first time is replaced with $\triangleright T1 \triangleleft$, the second time with $\triangleright T2 \triangleleft$, and so on. The letter T is taken from the word Time. After the replacement operation, the time along with its corresponding key is stored in a file. With this approach, it can be retrieved if needed. Examples of times that are identified and replaced with a specific key are mentioned in Table 7.

Table 7: Time Recognition System

Type	**Numbers**	**Postfix**
Short Time	Any number separated by dot(**.**)	AM PM am pm A.M. P.M. a.m. p.m.
Full Time	Numbers [0:23] **:** Numbers [0:59] **:** Numbers [0:59]	

4.8 Repeated Letter

Usually when chatting, some people repeatedly write certain letters, which can have various reasons, such as:

- Michael Erard: "Compensating for the lack of vocal cues when writing instead of speaking"
- Expressing sarcasm or negative emotions
- Persuasion: It has been scientifically proven that repeating information increases the likelihood of changing people's minds

Another preprocessing step is finding and removing these extra letters from text. In English, we have words where two consecutive letters are repeated and that word is meaningful (like Good), so the basis for the number of consecutive letters is a maximum of two. So if a letter is repeated more than twice consecutively, we replace all those letters with two letters, for example:

Goooooood $\longmapsto$ Good

Note that in this section, not only English letters are checked, but also the Space character is checked, and if there are more than two consecutive spaces in the dataset text, they are all replaced with two spaces. As a result, empty spaces present in the text are also filtered in this section. For example:

`This␣␣is␣a␣␣␣␣big␣␣␣␣␣␣␣window.` $\longmapsto$ `This␣␣is␣a␣␣big␣␣window.`

5 Expand

The third section relates to expanding or developing certain words or characters.

5.1 Mention

According to X company's definition: "A Twitter mention includes the username of another account preceded by the at sign (@)" [1]. In this preprocessing section, the goal is to expand the username. Usually usernames can be a single word or a combination of several words separated by underscores. The goal in this section is first to identify the username using the @ symbol. Then we check if it uses multiple words and underscores, separating the words by underscores and replacing the original username. Finally, we remove the @ symbol.

Hey, @dark_web This sentence is just for test. $\rotatebox{-90}{$\curvearrowright$}$
Hey, dark web This sentence is just for test.

5.2 Hashtag

A hashtag is usually formed from one or several words that are combined with underscores to form a single word. The special symbol for hashtags is "#". In this preprocessing section, the goal is to find hashtags and separate the words (if possible) and then replace them with the original word and remove the hashtag symbol.

Hey, #dark_web_2024 This sentence is just for test. $\rotatebox{-90}{$\curvearrowright$}$
Hey, dark web 2024 This sentence is just for test.

6 Saving the Dataset

After performing the steps mentioned in the previous sections in the order shown in Table 1, we will have a clean and organized dataset. The preprocessed files are stored separately in CSV file format, to be used in later stages.

7 References

[1] X (formerly Twitter). Glossary – X help center. https://help.x.com/en/resources/glossary, 2025. Accessed: 2025-04-05.

[2] S. J. Whitmore. tweet-collector – GitHub repository. https://github.com/sjwhitmore/tweet-collector, 2025.