# The Imperfect RegEx-based Tokenizer

Natural Language Processing (NLP) relies heavily on the accurate tokenization of text data. This process is particularly challenging when dealing with social media content, which often contains non-standard language use, platform-specific elements, and multilingual text. This analysis explores the development and application of a RegEx-based tokenizer designed specifically for processing tweets.

A corpus is a large collection of texts used for linguistic analysis. Tokenization is the process of breaking down text into individual units (tokens) for further processing and analysis. As stated from the PDF slides provided to us, it is up to the students to define what a "word" truly is in NLP1000. From this tokenization approach, a "word" was defined to include various elements found in the twitter social media text. Let's take a look at my regular expression pattern:

```
pattern = r"https?://[^\s]+|#\w+|@\w+|\b[\w']+\b|[!?.,:;]"
```

Each component of the RegEx pattern serves a specific purpose:

- **The URL pattern (https?://[^\s]+)** captures web links, which are common in tweets and often carry significant meaning;
- **Hashtags (#\w+)** and **mentions (@\w+)** are crucial for understanding tweet topics and user interactions;
- **The word pattern (\b[\w']+\b)** allows for contractions and possessives while maintaining word boundaries;
- **Individual punctuation marks ([!?.,:;])** are captured separately to preserve sentence structure and potential sentiment indicators.

The definition accommodates various elements that are not so surprisingly common in a social media text. Unlike some tokenizers that'll split contractions *( e.g. "isn't" into "is" "n't" )*, my approach kept such contractions intact. Aside from the sample input/output given to us, I have slightly modified it to include some of the patterns listed above, let's take a look at it:

**Name:** Homssi, Yazan M.
**Course || Section:** NLP1000 || S15

```
# Testing with Sample Corpus given to us
sample_corpus = [
    "This is an example of a corpus for processing.",
    "There isn't much to it, but here it is.",
    "Also, here is a number, 55. Is this a word?",
    "#Hashtags, @userHandles, links.co/wow",
    "Kilig, Dctionary isn't",
    "http://example.com"
]
```

Figure 1.1 (Sample Corpus- Input)

```
== Testing with sample corpus: ==
Sample Total tokens: 48
Sample Total vocabulary: 31
All words + counts:
, : 6
is : 4
. : 4
a : 3
this : 2
isn't : 2
it : 2
here : 2
an : 1
example : 1
of : 1
corpus : 1
for : 1
processing : 1
there : 1
much : 1
to : 1
but : 1
also : 1
number : 1
55 : 1
word : 1
? : 1
#hashtags : 1
@userhandles : 1
links : 1
co : 1
wow : 1
kilig : 1
dctionary : 1
http://example.com : 1
```

*Figure 1.2 (Sample Corpus - Output)*

The sample corpus output demonstrates the tokenizer's ability to handle various tweet elements. It correctly identifies hashtags and mentions as single tokens, preserves contractions as stated earlier, and separates punctuation. The high frequency of commas and periods suggests that even in a small sample, punctuation plays a significant role in tweet structure.

**Name:** Homssi, Yazan M.
**Course || Section:** NLP1000 || S15

As noted, it is up to the developer to choose whether or not they would want to include the punctuations, and that is exactly what I did ( since our good awesome professor did so in the sample ). The decision to include punctuation as separate tokens increases the overall token count and influences the frequency distribution. This means that commas and periods are among the most frequent tokens. Including the question mark "?" as a separate token allows for potential analysis of question frequency or sentence types in the dataset. As a matter of fact, in the Python Notebook submitted, if you were to run the sample corpus cell, you will notice a +1 in the total number of tokens and vocabularies due to the inclusion of the question mark “?”.

However, I would assume that this choice would be considered “noise” to other developers’ analyses if they did not want/need such tokens.

With regards to my actual output from the `tweets_for_mp2.csv` dataset:

- **Total tokens:** 2,400,086
- **Total vocabulary:** 254,478
- **Top tokens include punctuation** (".", "!", ","), **Filipino words** ("na", "ko", "sa"), **English words** ("the", "to", "you").

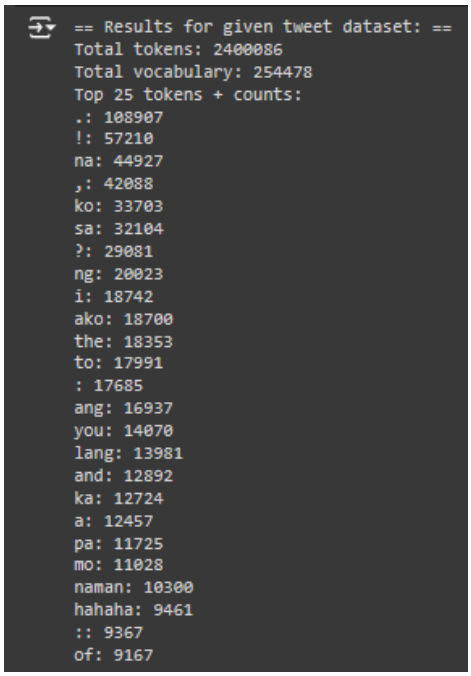


*Figure 1.3 (Actual Corpus - Output)*

The most frequent tokens included punctuation marks, common Filipino words like: "na", "ko", "sa", and English words like: "the", "to", "you").

The bilingual nature of the tweets, evident from the mix of Filipino and English words in the top tokens, presents both challenges and opportunities for NLP tasks. It reflects the linguistic reality of the dataset's origin but may require language-specific processing for certain analyses. This bilingualism could be valuable for studying code-switching behaviors or developing multilingual NLP models.

## Limitations and Strengths

Although I'd understand the extreme commonness of punctuation among the top tokens would raise questions about how truly effective this approach is for certain types of analysis. While yes it provides insight into writing styles, it may obscure content-based patterns. This highlights a limitation of the current tokenization method: It may require additional preprocessing steps for analyses focused on semantic content rather than structural elements, but alas, in my defense– Sir Edward did so first in the sample corpus output. 🙂

The tokenizer's handling of social media-specific elements I believe to be is a strength. By preserving hashtags, mentions, and URLs as single tokens, it allows for analyses that consider these elements' unique roles in social media communication. This would be considered particularly valuable for studies focusing on trends, user interactions, or link-sharing behaviors across different apps.

## General Reflection

This tokenization approach provides a relatively good foundation for analyzing the structure & basic content of tweets, it also reveals the complexities involved in processing social media text. As with many aspects of natural language processing, the ideal tokenization approach depends heavily on the specific goals of the analysis and may require constant improvements as those goals evolve.

**Name:** Homssi, Yazan M.
**Course || Section:** NLP1000 || S15