TextPreprocessing Assignment

Yuhuan Huang, Aug 2025

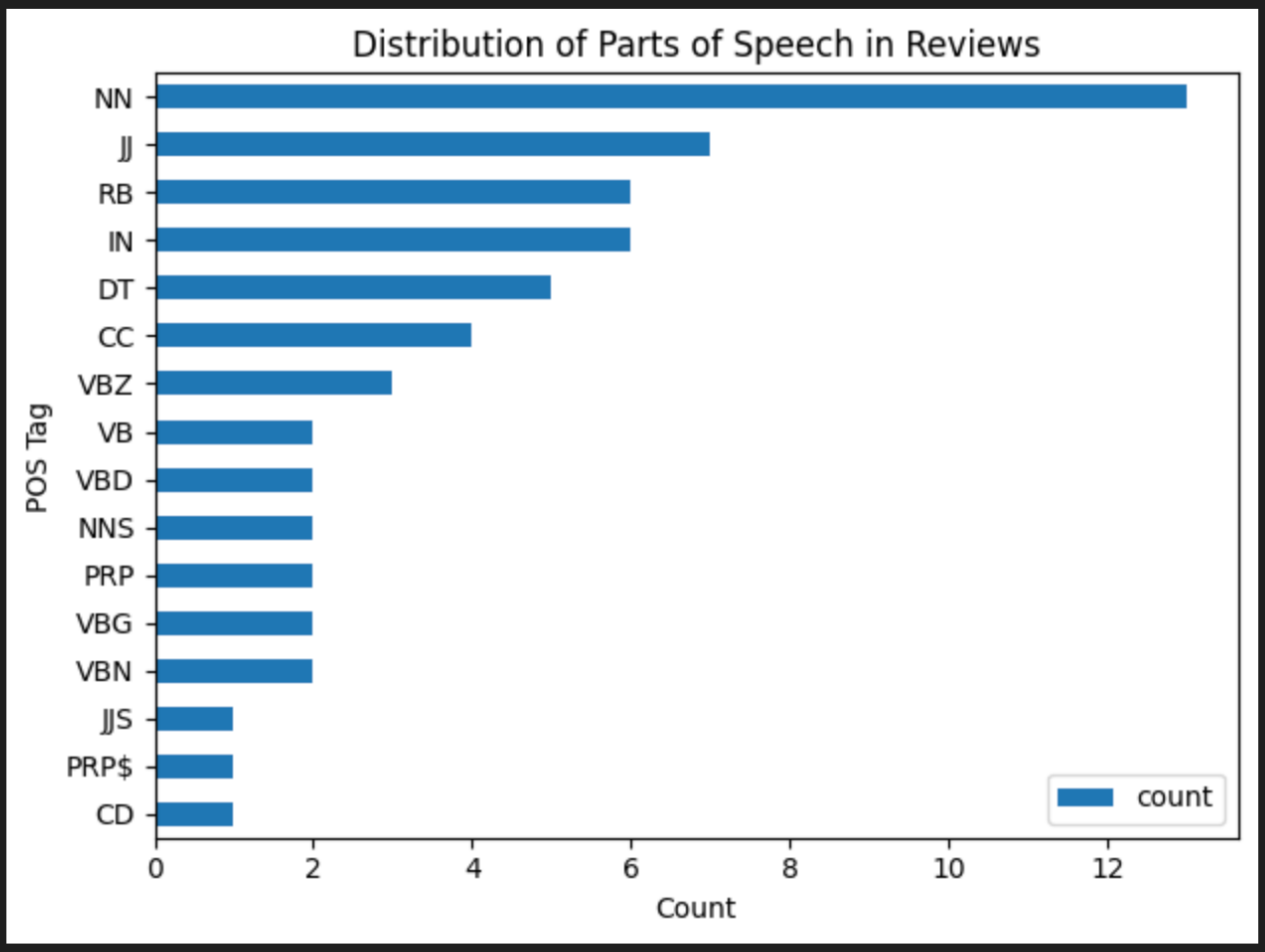
Output Console

1.Example Usage and Visualization:

```
[3]  ✓ 1.7s

...  POS counts:
            count
     CD         1
     PRP$       1
     JJS        1
     VBN        2
     VBG        2
     PRP        2
     NNS        2
     VBD        2
     VB         2
     VBZ        3
     CC         4
     DT         5
     IN         6
     RB         6
     JJ         7
     NN        13

...  <Figure size 1200x600 with 0 Axes>
```




```
TestResults(failed=0, attempted=8)
```

The result shows the distribution of POSs in the review text, it seems that nouns appear most frequently.

2. **Processing Multiple Reviews:**

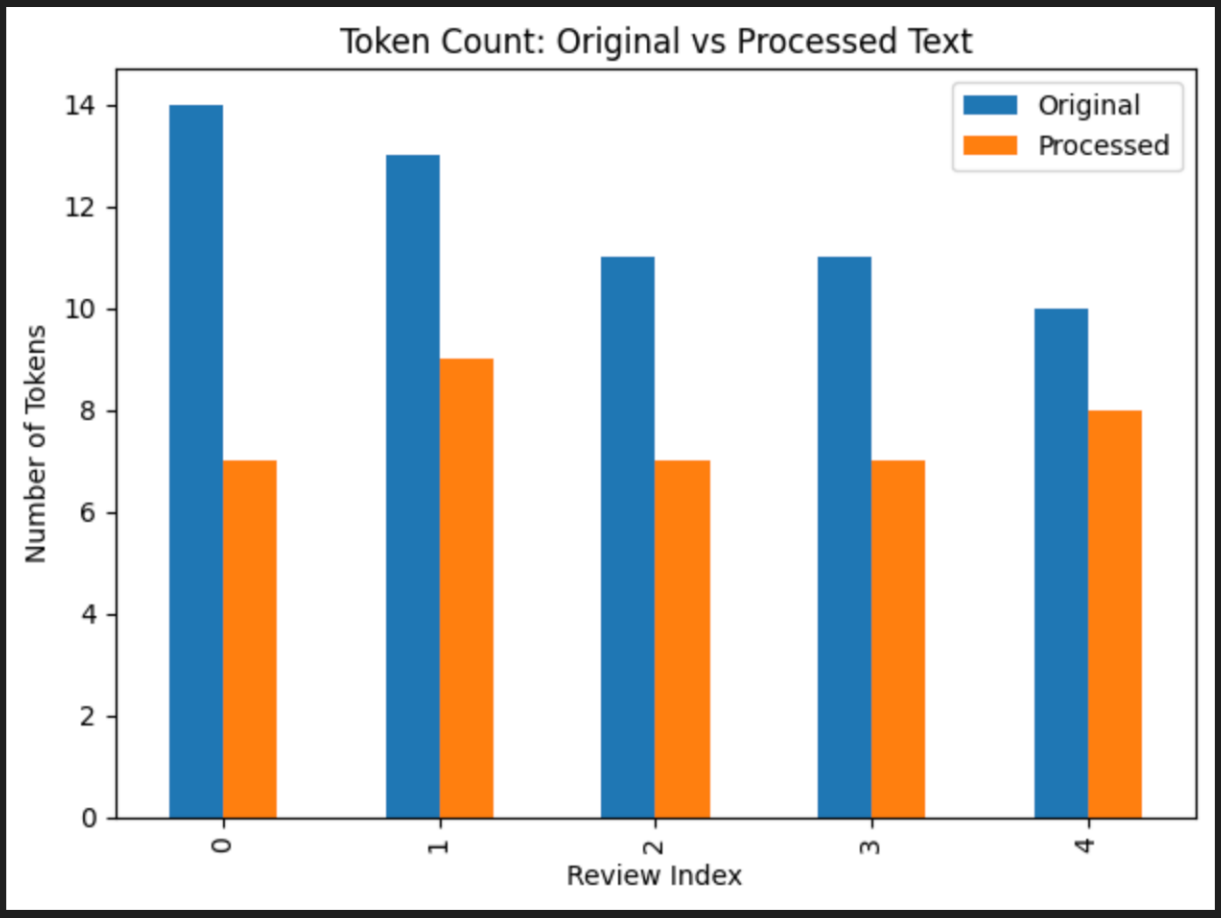
```
[8]
    # Display results
    print("Sample of processed reviews:")
    print(df[['review_text', 'processed_stemming', 'processed_lemmatization']].head())
```

```
...  Sample of processed reviews:
                                          review_text  \
    0  This product is amazing! I've been using it fo...
    1  Terrible experience. The product broke after t...
    2  Good product but a bit expensive. The quality ...
    3  Not worth the money. Save your cash and buy so...
    4  Best purchase ever! Highly recommended for any...

                                    processed_stemming  \
    0         product amaz ive use month work perfectli
    1  terribl experi product broke two day custom se...
    2       good product bit expens qualiti great though
    3               worth money save cash buy someth els
    4  best purchas ever highli recommend anyon look ...

                              processed_lemmatization
    0        product amaze ive use month work perfectly
    1  terrible experience product break two day cust...
    2    good product bit expensive quality great though
    3          worth money save cash buy something else
    4  best purchase ever highly recommended anyone l...
```

```
[9]
...   <Figure size 1000x600 with 0 Axes>
```




```
...
    Preprocessing Statistics:
    Average original length: 11.80 tokens
    Average processed length: 7.60 tokens
    Average reduction: 35.59%
```

From the results of stemming and lemmatization, I find that the lemmatization result may be slightly more "readable" than the stemming result. But for both cases, the "senitimental" words are preserved, which would be helpful for the analysis.

From the bar chart of the original vs the processed text, I find that the processing reduced the text significantly by length, which may preserve the most "useful" information, while throw away the less "useful" ones.

Reflections

1. Q: Analysis of Text Reduction: Examine the bar chart comparing original versus processed text lengths. In our example, we saw an average reduction of token count by approximately 40%. Explain why this reduction occurs and discuss whether this level of reduction might help or hinder sentiment analysis. Consider specific words that

were removed and their potential importance to sentiment.

My Ans: I think it is because we have removed the stopwords and punctuations. This would be helpful since we don't need these stopwords for analysis. It "contracts" information and preserves the most useful ones. However, I find that some mistakes happen. For example, The origin text goes "**not** worth money", while after removing the stop words, the review becomes "worth money", which may hinder our analysis. So we may need to reset the stop words, instead of directly using the NLTK stop words.

2. Q: Impact of Processing Choices : Compare the results of stemming versus lemmatization in our processed reviews. Looking at specific examples from the output, identify cases where one method might be more appropriate than the other for sentiment analysis. What are the trade-offs between these approaches in terms of preserving meaning while standardizing words?

My Ans: From the above results, I find that lemmatization has a slightly "better" result in preserving the textual information than stemming. However, the trade-off is that lemmatization may preserve larger amount of data than stemming, and lemmatization might be less "normalized" than stemming.

3. Q: Critical Evaluation of Pipeline : Our preprocessing pipeline includes normalization, tokenization, POS tagging, stopword removal, and stemming/lemmatization. Looking at the final processed texts, identify any sentiment-bearing words that might have been lost or altered during preprocessing. How might you modify the pipeline to better preserve sentiment information while still maintaining the benefits of preprocessing?

   My Ans: Firstly, in order not to remove the negation words like "not" from the abbreviation, we can expand words like "don't" to "do not" at first. Secondly, in the example, the punctuations like "!" are removed. However, these might also preserves sentiments. Therefore, we can revise the step and preserve some punctuations such like the "!".