Open in Colab

HW 1 Submission

Ross Chu

In this homework, you'll compare the different tokenizations that result from different clases of tokenizers. This homework is also for you to check in yourself on your Python proficiency; for all of the operations below (downloading a file, reading it in, counting objects), you should either be comfortable implementing them already or know how to find out how to do so yourself (if you find yourself struggling with them, we encourage you to take this class at a later date, with more Python experience under your belt).

Q1. Tokenize the following document with each of these models. Feel free to use the documentation linked (and AI Assistance) to do so for this low-level operation (but again remember that you have to be able to explain what your code is doing). For each of the tokenizers above, we want to see a list of tokens for this document (not numeric token IDs, but legible words) -- e.g., ["London", ".", ...]

- NLTK `word_tokenize` (https://www.nltk.org/book/ch03.html)
- Spacy `tokenize` (https://spacy.io/usage/spacy-101#annotations-token)
- Tiktoken BPE tokenization (https://github.com/openai/tiktoken) -- cl100k_base (GPT-3.5, GPT-4).

In [1]:

```python
import nltk
import spacy
import tiktoken
```

In [2]:

```python
# Doc string
document="London. Michaelmas term lately over, and the Lord Chancellor sitting in Lincoln’s Inn Hall. Implacable Novemb

# Initialize dictionary to store tokens
tokens = {}
```

In [3]:

```python
# NLTK tokenizer
# Reference: https://www.nltk.org/book/ch03.html
tokens['nltk'] = nltk.word_tokenize(document)
```

In [4]:

```python
# Spacy tokenizer
# Reference: https://spacy.io/usage/spacy-101#annotations-token
nlp = spacy.load("en_core_web_sm")
tokens['spacy'] = [token.text for token in nlp(document)]
```

In [5]:

```
# Tiktoken BPE tokenizer
# Reference: https://github.com/openai/tiktoken
enc = tiktoken.get_encoding("cl100k_base")
tokens['bpe'] = [enc.decode([token_id]) for token_id in enc.encode(document)]
```

In [6]:

```
# Print tokens for each tokenizer
for key in tokens.keys():
    print(f"{key} tokenizer:")
    print(tokens[key])
    print("\n")
```

```
nltk tokenizer:
['London', '.', 'Michaelmas', 'term', 'lately', 'over', ',', 'and', 'the', 'Lord', 'Chancellor', 'sitting', 'in', 'Linc
oln', '’', 's', 'Inn', 'Hall', '.', 'Implacable', 'November', 'weather', '.', 'As', 'much', 'mud', 'in', 'the', 'street
s', 'as', 'if', 'the', 'waters', 'had', 'but', 'newly', 'retired', 'from', 'the', 'face', 'of', 'the', 'earth', ',', 'a
nd', 'it', 'would', 'not', 'be', 'wonderful', 'to', 'meet', 'a', 'Megalosaurus', ',', 'forty', 'feet', 'long', 'or', 's
o', ',', 'waddling', 'like', 'an', 'elephantine', 'lizard', 'up', 'Holborn', 'Hill', '.', 'Smoke', 'lowering', 'down',
'from', 'chimney-pots', ',', 'making', 'a', 'soft', 'black', 'drizzle', ',', 'with', 'flakes', 'of', 'soot', 'in', 'i
t', 'as', 'big', 'as', 'full-grown', 'snowflakes—gone', 'into', 'mourning', ',', 'one', 'might', 'imagine', ',', 'for',
'the', 'death', 'of', 'the', 'sun', '.', 'Dogs', ',', 'undistinguishable', 'in', 'mire', '.', 'Horses', ',', 'scarcel
y', 'better', ';', 'splashed', 'to', 'their', 'very', 'blinkers', '.', 'Foot', 'passengers', ',', 'jostling', 'one', 'a
nother', '’', 's', 'umbrellas', 'in', 'a', 'general', 'infection', 'of', 'ill', 'temper', ',', 'and', 'losing', 'thei
r', 'foot-hold', 'at', 'street-corners', ',', 'where', 'tens', 'of', 'thousands', 'of', 'other', 'foot', 'passengers',
'have', 'been', 'slipping', 'and', 'sliding', 'since', 'the', 'day', 'broke', '(', 'if', 'this', 'day', 'ever', 'brok
e', ')', ',', 'adding', 'new', 'deposits', 'to', 'the', 'crust', 'upon', 'crust', 'of', 'mud', ',', 'sticking', 'at',
'those', 'points', 'tenaciously', 'to', 'the', 'pavement', ',', 'and', 'accumulating', 'at', 'compound', 'interest',
'.']


spacy tokenizer:
['London', '.', 'Michaelmas', 'term', 'lately', 'over', ',', 'and', 'the', 'Lord', 'Chancellor', 'sitting', 'in', 'Linc
oln', '’s', 'Inn', 'Hall', '.', 'Implacable', 'November', 'weather', '.', 'As', 'much', 'mud', 'in', 'the', 'streets',
'as', 'if', 'the', 'waters', 'had', 'but', 'newly', 'retired', 'from', 'the', 'face', 'of', 'the', 'earth', ',', 'and',
'it', 'would', 'not', 'be', 'wonderful', 'to', 'meet', 'a', 'Megalosaurus', ',', 'forty', 'feet', 'long', 'or', 'so',
',', 'waddling', 'like', 'an', 'elephantine', 'lizard', 'up', 'Holborn', 'Hill', '.', 'Smoke', 'lowering', 'down', 'fro
m', 'chimney', '-', 'pots', ',', 'making', 'a', 'soft', 'black', 'drizzle', ',', 'with', 'flakes', 'of', 'soot', 'in',
'it', 'as', 'big', 'as', 'full', '-', 'grown', 'snowflakes', '—', 'gone', 'into', 'mourning', ',', 'one', 'might', 'ima
gine', ',', 'for', 'the', 'death', 'of', 'the', 'sun', '.', 'Dogs', ',', 'undistinguishable', 'in', 'mire', '.', 'Horse
s', ',', 'scarcely', 'better', ';', 'splashed', 'to', 'their', 'very', 'blinkers', '.', 'Foot', 'passengers', ',', 'jos
tling', 'one', 'another', '’s', 'umbrellas', 'in', 'a', 'general', 'infection', 'of', 'ill', 'temper', ',', 'and', 'los
ing', 'their', 'foot', '-', 'hold', 'at', 'street', '-', 'corners', ',', 'where', 'tens', 'of', 'thousands', 'of', 'oth
er', 'foot', 'passengers', 'have', 'been', 'slipping', 'and', 'sliding', 'since', 'the', 'day', 'broke', '(', 'if', 'th
is', 'day', 'ever', 'broke', ')', ',', 'adding', 'new', 'deposits', 'to', 'the', 'crust', 'upon', 'crust', 'of', 'mud',
',', 'sticking', 'at', 'those', 'points', 'tenaciously', 'to', 'the', 'pavement', ',', 'and', 'accumulating', 'at', 'co
mpound', 'interest', '.']


bpe tokenizer:
['London', '.', ' Michael', 'mas', ' term', ' lately', ' over', ',', ' and', ' the', ' Lord', ' Chancellor', ' sittin
g', ' in', ' Lincoln', '’s', ' Inn', ' Hall', '.', ' Impl', 'ac', 'able', ' November', ' weather', '.', ' As', ' much',
' mud', ' in', ' the', ' streets', ' as', ' if', ' the', ' waters', ' had', ' but', ' newly', ' retired', ' from', ' th
e', ' face', ' of', ' the', ' earth', ',', ' and', ' it', ' would', ' not', ' be', ' wonderful', ' to', ' meet', ' a',
' Meg', 'al', 'os', 'aurus', ',', ' forty', ' feet', ' long', ' or', ' so', ',', ' w', 'add', 'ling', ' like', ' an', '
elephant', 'ine', ' lizard', ' up', ' Hol', 'born', ' Hill', '.', ' Smoke', ' lowering', ' down', ' from', ' chimney',
'-p', 'ots', ',', ' making', ' a', ' soft', ' black', ' dr', 'izzle', ',', ' with', ' flakes', ' of', ' so', 'ot', ' i
n', ' it', ' as', ' big', ' as', ' full', '-g', 'rown', ' snow', 'fl', 'akes', '—', 'gone', ' into', ' mourning', ',',
' one', ' might', ' imagine', ',', ' for', ' the', ' death', ' of', ' the', ' sun', '.', ' Dogs', ',', ' und', 'istingu
ish', 'able', ' in', ' m', 'ire', '.', ' H', 'orses', ',', ' scarcely', ' better', ';', ' spl', 'ashed', ' to', ' thei
r', ' very', ' blink', 'ers', '.', ' Foot', ' passengers', ',', ' j', 'ost', 'ling', ' one', ' another', '’s', ' umb',
'rellas', ' in', ' a', ' general', ' infection', ' of', ' ill', ' temper', ',', ' and', ' losing', ' their', ' foot',
'-h', 'old', ' at', ' street', '-c', 'orners', ',', ' where', ' tens', ' of', ' thousands', ' of', ' other', ' foot', '
passengers', ' have', ' been', ' slipping', ' and', ' sliding', ' since', ' the', ' day', ' broke', ' (', 'if', ' thi
s', ' day', ' ever', ' broke', '),', ' adding', ' new', ' deposits', ' to', ' the', ' crust', ' upon', ' crust', ' of',
' mud', ',', ' sticking', ' at', ' those', ' points', ' ten', 'ac', 'iously', ' to', ' the', ' pavement', ',', ' and',
' accumulating', ' at', ' compound', ' interest', '.']
```

Q2. Examine the different tokenizations for the passage above -- i.e., actually read through them and see how they differ. In a paragraph or two, characterize the salient differences in tokenization between a.) NLTK and Spacy and b.) NLTK and BPE. Reference real examples in the text. (At the end of this homework, you want to be able to discuss the practical differences between tokenization methods).

HW Response:

NLTK vs Spacy

NLTK seems better than Spacy at capturing words that contain punctuation. For example, full-grown, street-corners, and foot-hold are treated as single tokens in NLTK while they are separated by dashes for Spacy.

NLTK vs BPE

BPE splits words into its subwords produced by NLTK. Doing so increases the dimensionality of vocabulary by breaking down each word into many sub words. Examples include implacable VS impl + ac + able, megalosaurus VS meg + al + os + aurus, weddling VS w + add + ling, and elephantine VS elephant + ine.

Q3. Download the full text of *Pride and Prejudice* (https://raw.githubusercontent.com/dbamman/anlp24/main/data/1342_pride_and_prejudice.txt) and tokenize it using each of the methods above. How many word types (in the formal sense we discussed in class) does each tokenization method have for that complete file?

HW Response:

NLTK: 7475 types

Spacy: 6780 types

BPE: 8364 types

In [7]:

```
# Doc string
file_path='/Users/RossChu/GoogleDrive/ANLP2024/anlp24/data/1342_pride_and_prejudice.txt'
with open(file_path, 'r', encoding='utf-8') as file:
    document = file.read()

# Set tokenizers for spacy and tiktoken
nlp = spacy.load("en_core_web_sm")
enc = tiktoken.get_encoding("cl100k_base")

# Tokenize doc with each tokenizer
tokens = {}
tokens['nltk'] = nltk.word_tokenize(document)
tokens['spacy'] = [token.text for token in nlp(document)]
tokens['bpe'] = [enc.decode([token_id]) for token_id in enc.encode(document)]
```

In [8]:

```
# Count word types for each tokenization method
types = {key: len(set(tokens[key])) for key in tokens.keys()}
print(types)
```

```
{'nltk': 7475, 'spacy': 6780, 'bpe': 8364}
```

Q4. Which text has the greater type-token ratio, *Pride and Prejudice* (https://raw.githubusercontent.com/dbamman/anlp24/main/data/1342_pride_and_prejudice.txt) or *Emma* (https://raw.githubusercontent.com/dbamman/anlp24/main/data/158_emma.txt)? Calculate the TTR for both texts using the NLTK tokenizer, but only use the first 1,000 tokens from each text when calculating its TTR.

HW Response:

Emma has more types (410) compared with pride and prejudice (360), resulting in a lower TTR for emma (2.44) compared with pride and prejudice (2.78). This suggests that Emma uses richer vocabulary than pride and prejudice.

In [9]:

```
# Doc strings for two texts
doc = {}
for doc_file in ['1342_pride_and_prejudice','158_emma']:
    file_path=f'/Users/RossChu/GoogleDrive/ANLP2024/anlp24/data/{doc_file}.txt'
    with open(file_path, 'r', encoding='utf-8') as file:
        doc[doc_file] = file.read()

# Get first 1000 tokens from each doc
tokens = {key: nltk.word_tokenize(doc[key])[:1000] for key in doc.keys()}

# Calculate TTR and its numerator / denominator
num = {key: len(tokens[key]) for key in doc.keys()}
den = {key: len(set(tokens[key])) for key in doc.keys()}
ttr = {key: round(num[key]/den[key],2) for key in doc.keys()}

# Print results
```

```
print(ttr)
print(num)
print(den)
```

```
{'1342_pride_and_prejudice': 2.78, '158_emma': 2.44}
{'1342_pride_and_prejudice': 1000, '158_emma': 1000}
{'1342_pride_and_prejudice': 360, '158_emma': 410}
```