Implementing BPE tokenizer and applying it to tokenize text. BPE implemented from scratch – without using existing Python package tokenizers where ONLY use re and pandas packages are used

This program accepts three command line arguments: a positive integer **K** representing the number of byte pair merges, a training text file **TRAIN_FILE**, and a test text file **TEST_FILE**.

Key Steps in the Program:

1. **Argument Validation**:
    - Checks if exactly three arguments are provided. If not, it displays an error message and exits.
    - Verifies the existence of the training and test files in the same directory as the script. If either file is missing, it displays an error message and exits.
    - If the K argument is not within the expected range, it defaults to 5.
2. **Training Data Processing**:
    - The training data from TRAIN_FILE is treated as a single string. All punctuation is removed, and non-printable characters are cleaned up, leaving only characters from the initial vocabulary **V** which includes the English alphabet and a stop token.
3. **BPE Training**:
    - The initial vocabulary is augmented with the stop token character.
    - The BPE learner performs K merges based on frequency of adjacent character pairs, updating the vocabulary after each merge.
4. **Test Data Processing**:
    - Similar to training data, the test data is cleaned of punctuation and non-printable characters.
5. **Tokenization**:
    - Using the final vocabulary VV, the test data is tokenized. This involves replacing the most frequent pairs of characters or character sequences with single tokens.
6. **Output**:
    - Saves the final vocabulary to a text file named VOCAB.txt.
    - Saves the tokenized test data to a file named RESULT.txt.

    - Prints the tokenization results on screen, showing up to 20 tokens, and notes if the tokenized text exceeds this length.

Performance Timing:

- The time taken to generate the final vocabulary and to tokenize the test data is measured separately and displayed along with the tokenization results.

Final Display:

- Outputs the BPE results on the console, including the number of merges, file names used, and the actual tokenization result, followed by the performance timings.

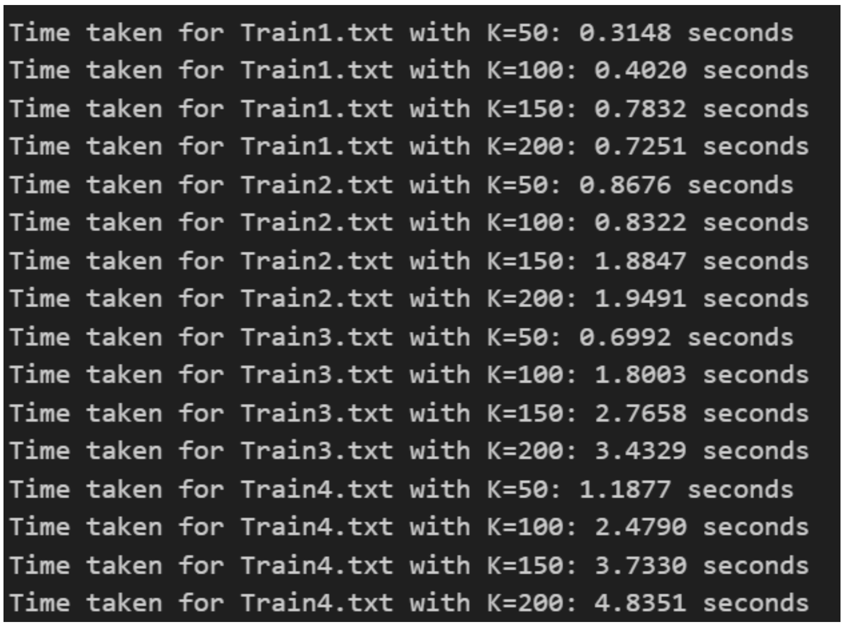
This program automates the process of BPE tokenization from training through testing, ensuring efficient handling of text data while providing informative outputs for evaluation.

Summary / observations / conclusions

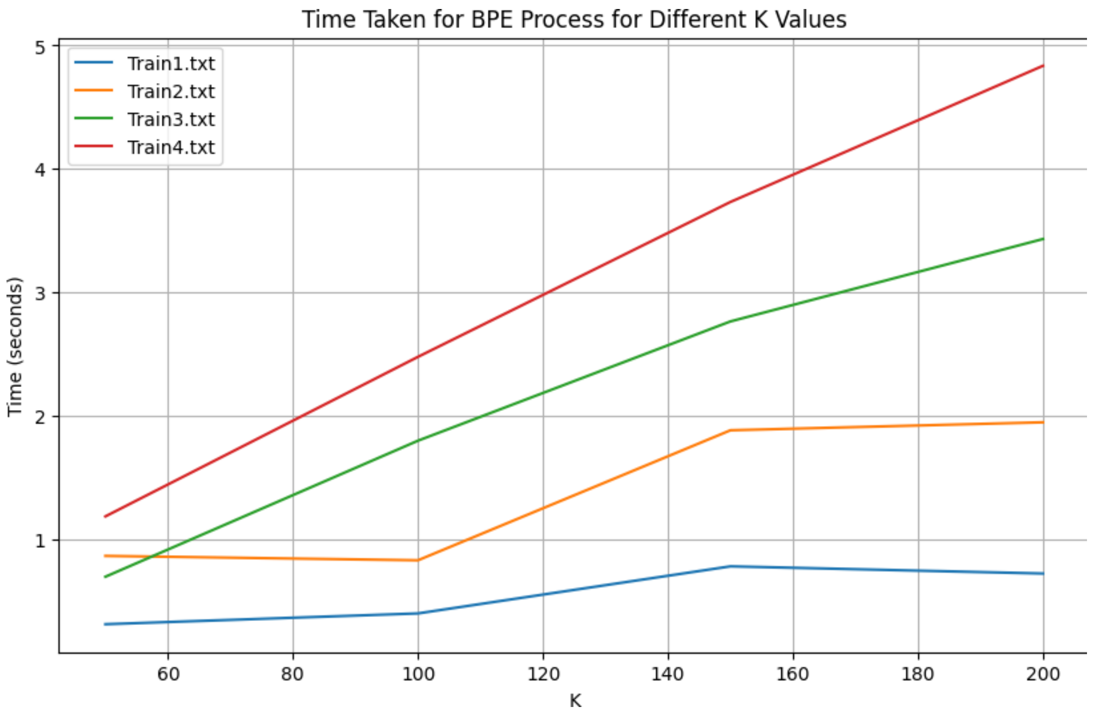
Run Time Analysis:

As we look at time taken for BPE process for different K values (merges) and for different Training file (size) its clear that as the number of merges (K) and the size of the training files increase, the time it takes for the algorithm to run also increases. This shows that BPE process is sensitive to both the K merge operation and the size of data it handles (Training file size).

For larger training file size, there's a significant spike in run times, as we move from K=150 to K=200, showing that the algorithm really starts to slow down when dealing with more data and more complex merges. Interestingly, for smaller file size its actually opposite, run times actually drop when moving from K=150 to K=200. From this we can say that the algorithm handles fewer data more efficiently even with more merges, possibly due to less data overhead or more effective optimization at these merge levels.



```
Time taken for Train1.txt with K=50: 0.3148 seconds
Time taken for Train1.txt with K=100: 0.4020 seconds
Time taken for Train1.txt with K=150: 0.7832 seconds
Time taken for Train1.txt with K=200: 0.7251 seconds
Time taken for Train2.txt with K=50: 0.8676 seconds
Time taken for Train2.txt with K=100: 0.8322 seconds
Time taken for Train2.txt with K=150: 1.8847 seconds
Time taken for Train2.txt with K=200: 1.9491 seconds
Time taken for Train3.txt with K=50: 0.6992 seconds
Time taken for Train3.txt with K=100: 1.8003 seconds
Time taken for Train3.txt with K=150: 2.7658 seconds
Time taken for Train3.txt with K=200: 3.4329 seconds
Time taken for Train4.txt with K=50: 1.1877 seconds
Time taken for Train4.txt with K=100: 2.4790 seconds
Time taken for Train4.txt with K=150: 3.7330 seconds
Time taken for Train4.txt with K=200: 4.8351 seconds
```



Observations: Effect of Number of merges on tokenization results

Test Input: in the state of Republic, the city Department of civic Engagement has been overseeing the Hartsfield Memorial project, //00which aim to preserve the historical sites affected by past wars. This initiative, strongly supported by both Republicans and other ' political factions, may help unify the community after divisions that564 have persisted far into the present day. The city anticipates that this project, by promoting a common\\00 heritage, will bolster a sense of unity and pride. The funds allocated for this purpose are crucial, as they ensure that the sites@ are maintained and accessible for air and land tours, giving everyone a chance to connect with their history.

- As K increases, the algorithm typically forms more complex tokens.
- With small K value (merges), we are getting **common suffixes or prefixes**. When K=50,

Tokenization Result: (Result of only 20 tokens displayed)

in, the, st,ate, of, R,ep,u,bl,ic, the, c,it,y, D,ep,ar,t,men,t, of, c,iv,ic, E,n,g,a,g,e,men,t, h,a,s, b,e,en, o,v,er,s,e,e,ing, the, H,ar,t,s,f,i,e,l,d, M,e,m,or,i,a,l, pro,j,ec,t, w,h,ic,h, ai,m

For moderate merges, we can see, it starts combining these into frequent morpheme, words or sub-word units. Here is the tokenization result of K=100 and 150 (Result of only 20 tokens displayed)

<u>K=100 Result:</u>

in, the, state, of, R,ep,ubl,ic, the, c,ity, D,epart,ment, of, civic, E,n,g,ag,e,ment, h,as, b,e,en, o,v,er,s,e,e,ing, the, H,art,s,f,i,el,d, M,em,or,i,al, pro,j,ec,t, w,h,ic,h, ai,m

<u>K=150 result:</u>

in, the, state, of, R,ep,ublic, the, city, Department, of, civic, Engag,e,ment, has, b,e,en, o,v,er,s,e,e,ing, the, H,art,s,f,i,el,d, M,em,or,ial, pro,ject, wh,ic,h, ai,m,

<u>K=200 result</u>: With a higher number of mergers, we can see words and sub-words forming.

in, the, state, of, Republic, the, city, Department, of, civic , Engage,ment, has, been, ov,er,s,ee,ing, the, H,art,s,fi,el,d, M,eorial, project, which, aim