LAB4: LARGE SCALE DATA (TEXT) PROCESSING WITH HADOOP MAPREDUCE

Featured Activity1: WordCount on Classical Latin text

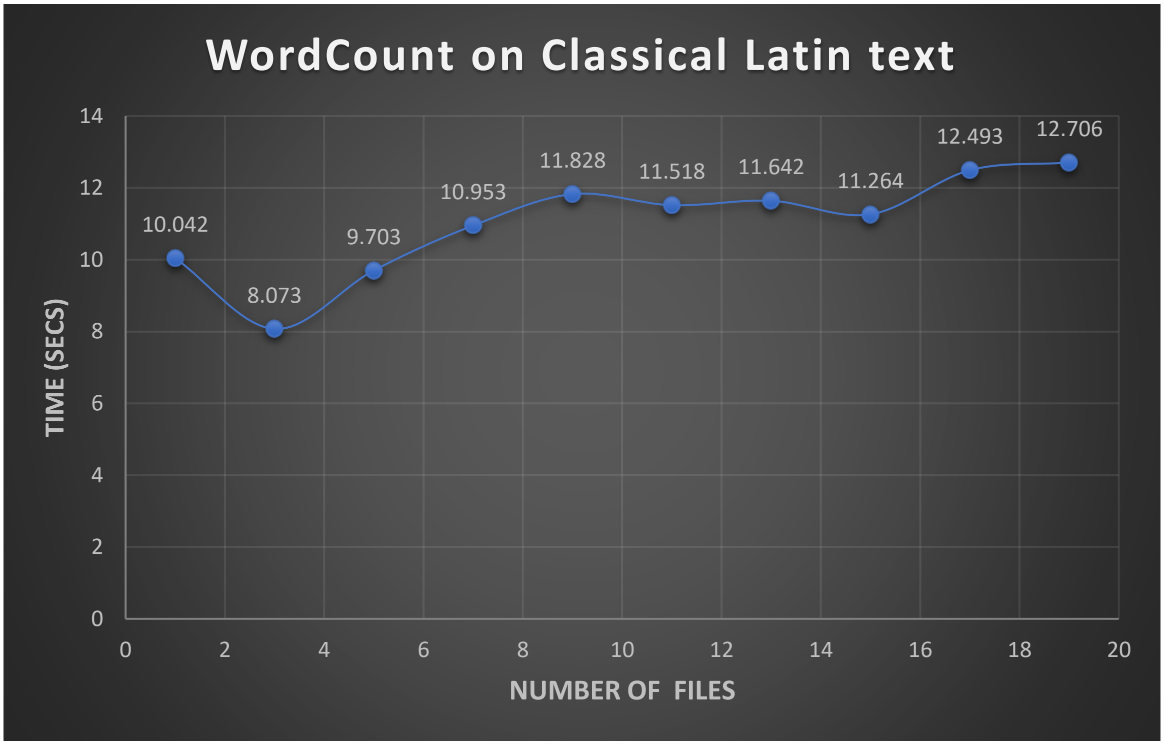
Output Format:

If two words Word1 and Word2 have lemmas Lemma1 and Lemma2 respectively, then the output format is as shown below:

Word1	<Location1><Location2>, Count : 2
Lemma1	<Location1><Location2>, Count : 2
Word2	<Location3><Location4><Location5>, Count : 3
Lemma2	<Location3><Location4><Location5>, Count : 3

*where each location is of the form <DocumentID. ChapterID. Line-number.>

Plot for Number of files Vs Execution Time:



- Seen above is the execution time vs Number of files for WordCount on Classical Latin text using lemmatization.
- As seen, the usual trend is that the time for execution increases as the number of input files to the Mapper increases with very little fluctuations.
- The execution time sometimes drop with increase in the number of files, this can be accounted to the size of the input files.

Featured Activity2: Word co-occurrence among multiple documents

a. Bigram

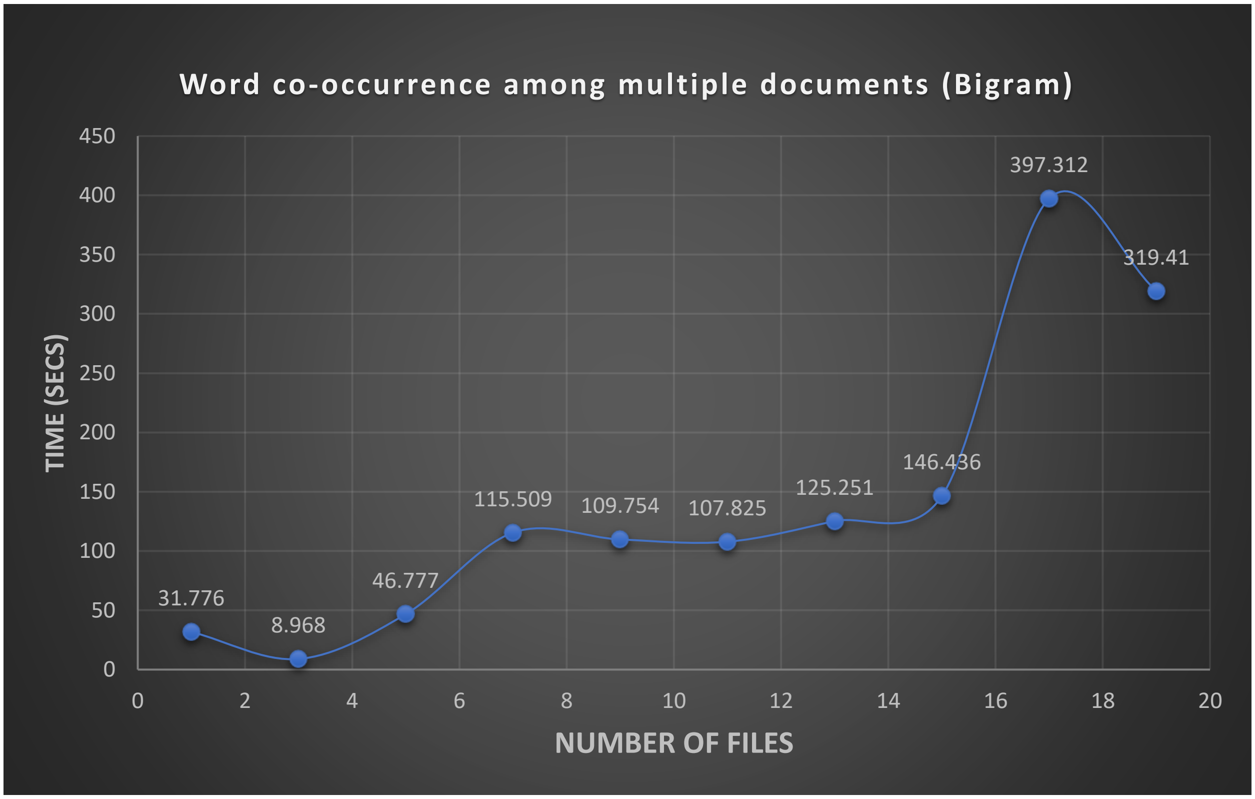
Output Format:

If two words Word1 and Word2 are neighbors and have the following lemmas:

Word1 → Lemma1 and Lemma2
Word2 → Lemma3 and Lemma4

```
Word1  Word2          <Location1><Location2>, Count : 2
Lemma1 Lemma3         <Location1><Location2>, Count : 2
Lemma2 Lemma3         <Location1><Location2>, Count : 2
Lemma1 Lemma4         <Location1><Location2>, Count : 2
Lemma1 Lemma4         <Location1><Location2>, Count : 2
```

Plot for Number of files Vs Execution Time:



- Seen above is the execution time vs Number of files for WordCooccurrence-Bigram on Classical Latin text using lemmatization.
- As with wordcount on classical text, the execution time increases with increase in the number of input files
- Also, we can see that the execution is overall greater for Bigrams compared to the classical wordcount as time is spent for calculating the neighbors for each word.

- Even with this there is a slight drop in the time for 19 files, this can be because of the small size of the 19th file .

Featured Activity2: Word co-occurrence among multiple documents

b. Trigram

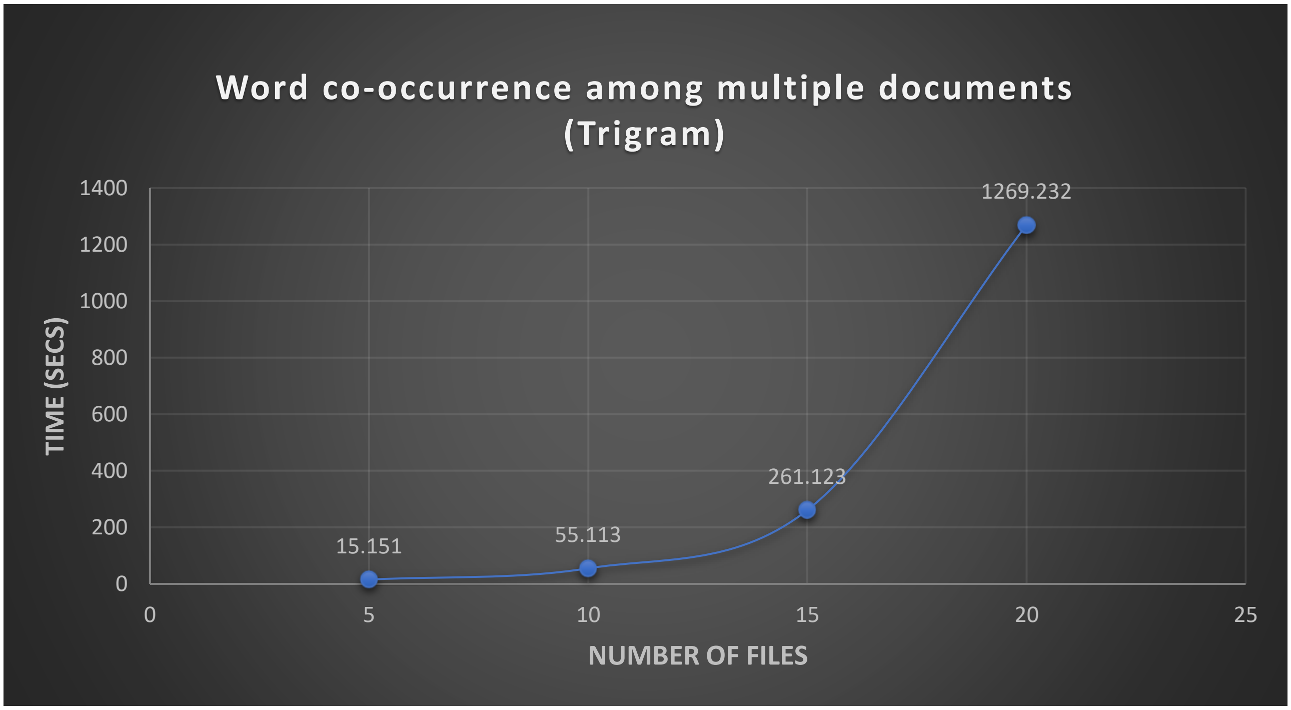
Output Format:

If two words Word1 and Word2 are neighbors and have the following lemmas:

Word1 → Lemma1 and Lemma2
Word2 → Lemma3 and Lemma4
Word3 → Lemma5 and Lemma6

```
Word1  Word2 Word3          <Location1><Location2>, Count : 2
Lemma1 Lemma3 Lemma5        <Location1><Location2>, Count : 2
Lemma2 Lemma3 Lemma5        <Location1><Location2>, Count : 2
Lemma1 Lemma4 Lemma5        <Location1><Location2>, Count : 2
Lemma1 Lemma4 Lemma6        <Location1><Location2>, Count : 2
Lemma2 Lemma4 Lemma5        <Location1><Location2>, Count : 2
Lemma2 Lemma4 Lemma6        <Location1><Location2>, Count : 2
Lemma1 Lemma3 Lemma6        <Location1><Location2>, Count : 2
Lemma2 Lemma3 Lemma6        <Location1><Location2>, Count : 2
```

Plot for Number of files Vs Execution Time:



- Seen above is the execution time vs Number of files for WordCooccurrence-Trigram on Classical Latin text using lemmatization.
- I have performed the trigram cooccurrence using small files, as bigger files take a lot of time to execute even with 1 or 2 files at the input.
- As with wordcount on classical text, the execution time increases with increase in the number of input files
- Also, we can see that the execution is overall greater for Trigrams compared to the classical wordcount as well as for Bigrams as time is spent for calculating the two neighbors for each word.
- Also, as the number of input files increases, the execution time shoots up to a large number. This is as expected, because more number of files increases the computation drastically.