THE MOST FREQUENT WORDS
Project Three (3)

ABLORDEPPEY Prosper

***Abstract* –The goal of this project is to determine the most frequent words of each one of some literary works and compare the results obtained with the exact counts. The algorithm developed by Misra & Gries for determining at most $(k-1)$ frequent items in a data stream was adopted in this presentation, finding the words that occur more than $\left(\frac{n}{k}\right)$ times in the data stream.**

***Keywords* – n -: total number of unique elements in the stream, k -: number of counters used by the misra-greis algorithm.**

PROJECT OUTLINE

The outline of the project in order is given as.

- Introduction
- Project Directory Structure
- A Midsummer Nights Dream
- The Hamlet
- Conclusion

I. INTRODUCTION

The literary works considered in this project are two (2) works of **Shakespeare** with each, translated in three (3) different languages *English, French* and *German*. The titles of these works in English are; **A Midsummer Nights Dream** and **The Hamlet**

II. PROJECT DIRECTORY STRUCTURE

The folder structure for this project is graphically given in Fig. (1). From the representation, the project has two (2) directories and the *FrequentCounter* python notebook file. Detailed description is presented below;

- **shakespeare directory** :- This folder hosts two (2) literary works of Shakespeare considered in this project. In total, the folder has $2 \cdot 3 = 6$ literary works since each work has translations in English, French and German. For instance, Hamlet has variations, *hamlet-english, hamlet-french* and *hamlet-german*.

- **output directory** :- This directory contains the output register (in csv) for all literary works under consideration. Each register has one-hundred and four (104) columns. The first column corresponds to the *rank* of each unique word presented by the counters. The first counter considered was the exact counter which counts the exact number of occurrences of each word in the literary work ranked in descending order. The remaining one hundred and two (102) columns after the exact prefixed with a *k* holds the most frequent words in the literary work presented in descending order computed using the **Misra & Reis** [1] frequency counter for one-hundred *k*'s. The domain of the first 100 *k*'s are even numbers ranging from $(2-200)$. In addition to this range, $k = \lfloor \frac{n}{2} \rfloor$ and $k = n$ were considered as presented in each register. The goal for considering varying values of *k* is to clearly understand the trend of output for every different *k*. The word with rank of one (1) for any counter corresponds to the first most frequent word for that counter, rank two (2) corresponds to the second most frequent and so on for that counter under study.

  There are 6 csv registers/files in this directory. For instance, the files as shown for the output directory in Fig. 1 is the output registers for hamlet-english literary work only.

```
Project-Three(3)
├─ shakespeare
│   ├─ hamlet-english.txt
│   ├─ hamlet-french.txt
│   ├─ hamlet-german.txt
│   └─ ...
├─ output
│   ├─ hamlet-english.csv
│   ├─ hamlet-french.csv
│   ├─ hamlet-german.csv
│   └─ ...
└─ FrequentCounter.ipynb
```

Fig. 1 - Project Directory Structure

- **FrequentCounter.ipynb**
  This is the python notebook file generating the *output* and *combined* registers as detailed above. This file implements three (3) classes. Namely
  – FrequentCounter (Counter Implementation)
  – StreamAlg (Consider Input as Stream)
  – filePreprocessor (Filter Stop-words and Punctuation)

III. A Midsummer Nights Dream

A. English

There are $9,060$ tokens in this version of the literary work. On implementing the exact counter, it took about $11ms$ to fetch all 2883 unique tokens. In addition, for the one-hundred and two (102) k values, it took about $16.5secs$ to complete the Misra and Greis algorithm.

For $k < 8$, we observed no result for the Misra and Greis counters. After $k = 90$, result from the Misra and Greis tend to approach rankings from obtained from the exact counter. At $k = n = 2884$, the result from the Misra and Greis converged to that of the exact counters.

B. French

There are $14,278$ valid tokens in this version of the literary work which took about $6ms$ to fetch all 5078 unique tokens. For the one-hundred and two (102) k values, it took about $35secs$ to execute the Misra and Greis implementation on all k values.

There was no Misra and Greis counter result for $k = 2$. After $k = 160$, counter results from the Misra and Greis tend to approach frequent values obtained using the exact counter. At $k = n = 5079$, the result from the Misra and Greis converged to that of the exact counters.

C. German

The literary work contains $8,477$ valid tokens of which $3,851$ are unique. It took about $6ms$ to obtain the unique words. For all one-hundred and two (102) k values, it took about $17secs$ to finish running the Misra and Greis method.

Counter estimates started reporting correct values after $k = 108$. Again, $k = n = 3852$ recorded convergence of the the Misra and Greis method to the results from the exact counter.

IV. Hamlet

A. English

$4,616$ unique words were observed from the total $16,121$ tokens. Entire search took about $11ms$. The algorithm took about $35secs$ to finish executing Misra and Greis estimates for the one-hundred and two (102) k values.

After about $k = 62$, the top first five (5) most frequent words were correctly estimated. Where $k = n = 4617$, we had the convergence of Misra and Greis results to the exact counter's results.

B. French

$8,522$ unique words were found from the total $29,553$ tokens, which lasted about $11ms$. The algorithm took about $91secs$ to execute Misra and Greis estimates for the one-hundred and two (102) k values.

Counter estimates started reporting correct values after $k = 130$. Again, $k = n = 8523$ recorded convergence of the the Misra and Greis method to the results from the exact counter.

C. German

There are $16,472$ valid tokens in this version of the literary work which took about $6ms$ to fetch all $5,754$ unique tokens. For the one-hundred and two (102) k values, it took about $43secs$ to execute the Misra and Greis implementation on all k values.

There was no Misra and Greis counter result for $k < 6$. After $k = 90$, top five (5) counter results from the Misra and Greis tend to approach frequent values obtained using the exact counter. At $k = n = 5755$, the result from the Misra and Greis converged to that of the exact counters.

V. Conclusion

From the above results and analysis, we arrive at the following conclusions;

- For smaller values of k, we record very bad estimates for the most frequent words in the literary works. For larger values, our estimates get better. We can infer that, the parameter *k* controls the quality of the results obtained.
- At most $(k - 1)$ counters are obtained at any time.

References

[1] J. Misra and David Gries, "Finding repeated elements", *Science of Computer Programming*, vol. 2, no. 2, pp. 143–152, 1982.
URL: https://www.sciencedirect.com/science/article/pii/0167642382900120