**Knowledge-base for Word Prediction**

Ohad Abramovitz: 307880112

Ori Suchy: 204051114

The task:

Map-Reduce system for calculating the conditional probability of each trigram found in a given corpus.

This system input corpus is Hebrew 3-Gram dataset of Google Books Ngrams.

The output of the system is a list of word trigrams and their conditional probabilities.

The list is ordered:

1. By ascending
2. By the probability , descending.

The probabilities are calculated as suggested by [Thede & Harper](https://dl.acm.org/doi/10.3115/1034678.1034712):

Where:

* is the number of times occurs.
* is the number of times sequence ) occurs.
* is the number of times sequence occurs.
* is the total number of word instances in the corpus.
* is the number of times occurs.
* is the number of times sequence occurs.

Output file: [link](https://oo-dspsp-ass2.s3.amazonaws.com/output4/part-r-00000)

Description

The system is divided to 4 different Map-Reduce steps, the first 3 steps calculate the sums of the probability equation, the last step is for sorting the results:

1. Step1 – First sum:

Input: 3-Gram dataset

* + Map:
    - Parse each of the input's records
    - Write to context:
  + Combiner:

Local aggregation of:

* + Reducer:
    - Count
    - Count s
    - For each Count s related to it
    - Calculate for each triplet
    - Write to context:
      * (all the pairs)

Output – "output1"

1. Step2 – Second sum:

Input: "output1"

* + Map:
    - Parse each of the input's records
    - Write to context:
  + Combiner:

Local aggregation of:

* + Reducer:
    - Count
    - Count s
    - For each Count s related to it
    - Calculate for each triplet
    - Write to context:
      * (all the pairs)

Output – "output2"

1. Step3 – Third sum:

Input: "output2"

* + Map:
    - Parse each of the input's records
    - Write to context:
  + Combiner:

Local aggregation of:

* + Reducer:
    - Count
    - Count s
    - For each Count s related to it
    - Calculate for each triplet
    - Write to context:
      * (all the pairs)

Output – "output3"

1. Step4 – Sort:

Input: "output3"

* + Map:
    - Parse each of the input's records
    - Write to context:
  + Combiner:

Local aggregation of:

* + Reducer:
    - Count
    - Count s
    - For each Count s related to it
    - Calculate for each triplet
    - Write to context:
      * (all the pairs)

Output – "output4"

Statistics:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Local**  **Aggregation** | **#Records** | **Size** |
| Step 1 | With | 8,046,830 | 0.16 GB |
| Without | 277,762,412 | 5.64 GB |
| Step 2 | With | 2,676,065 | 0.07 GB |
| Without | 7,470,120 | 0.21 GB |
| Step 3 | No Combiner | 1,747,596 | 0.15 GB |
| Step 4 | No Combiner | 1,650,268 | 0.07 GB |

Analysis: