ELEN4020 LAB 3 - MapReduce

School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2050

Ntladi Mohajane-1599953, **Phetolo Malele**-1412331, **Zwavhudi Mulelu**-1110574

I. Introduction

This lab exercise illustrates the processing of big data through the use of MapReduce framework. The MapReduce framework is used for processing large amounts of data and it works in two phases, namely, Map and Reduce[1]. The *mapping* phase includes a decomposition of big data processing problem into sub-tasks arranged by a master task and executed by workers. The *reducing* phase then merges results obtained from workers performing sub-tasks into a final result. The common data processing tasks used for illustration are: word-count, Top-K query and inverted index text.

II. MapReduce Framework

The MapReduce framework follows 4 steps (Splitting, Mapping, Shuffling and Reducing) as shown in Figure 1. The **Splitting** phase takes an input and splits it into manageable fixed-size chunks called input splits. **Mapping** is the initial phase in the execution of the map-reduce program. Here, the *map* function takes in each input split and maps it to a value of 1 - this is done to count the frequency of each word. **Shuffling** takes words from Mapping phase and groups equivalent words with their respective value. **Reducing** phase aggregates results from the Shuffling phase. All values of the same key(word) from the Shuffling phase are amalgamated to produce a single output value.

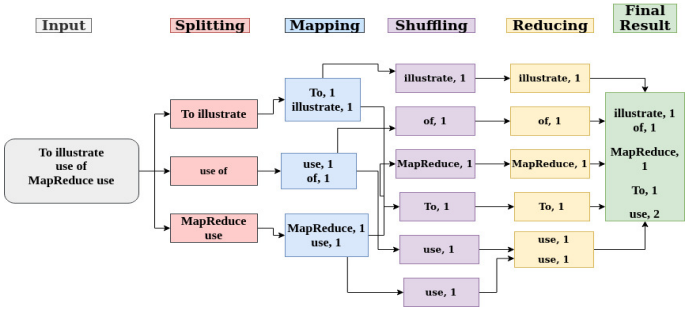

Figure 1. MapReduce architecture

This lab exercise demonstrates the use of a python-based implementation - Mrs MapReduce(version 0.9). This type of MapReduce framework can be installed via pip. The next section illustrates the use of MapReduce framework through various algorithms.

III. MapReduce algorithms

A. Word-Count

This is a simple algorithm which returns the frequency of each unique word contained in a text file. This however excludes common English words called Stop Words. The Stop Words used in this algorithm are specified in the *stop_words.py* file. It follows the architecture described in Section II. All words are considered case-insensitive and to enable this, the algorithm converts all words to lower case prior to counting their frequency. This algorithm uses the *map* function to map each word(key) to a value of 1 and then shuffles and assembles words that are alike. This algorithm also uses the *reduce* function which locates all identical words returned by the *map* function and then adds their values to obtain the sum of each unique word. Algorithm 1 represents the psuedocode for the described Word-Count algorithm.

Algorithm 1: `Word_Count`

```
import mrs, re, string, stop_words
wordRe = create regEx to match all word characters
    including alphanumeric
stopWords = get_stop_words() from stop_words
Class Word_Count(mrs.MapReduce)
    Function map(self, lineNumber,text)
        for word in wordRe.findall(text) do
            if word not in stopWords and word is not a
        digit then
                strip all punctuations from word
                convert word to lowerCase
                return word, 1
    Function reduce(self,word,count)
        group same words and sum all values
    associated(1s)
        return word, totalSum
```

B. Top-K

This algorithm implements a functionality that finds the top K (K=10 and 20 in this lab) most frequent words in a text. The *compare_by_occurrence*, *sort_words* and *get_top_k_words* functions are used to achieve this functionality. The *compare_by_occurrence* function is used to sort the list of words in descending order and returns counted pairs, *sort_words* uses the counted pairs returned by *compare_by_occurrence* and it continuously sorts the words and frequencies in descending order when a new word is read from the file. *sort_words* returns a key value pair result. The *get_top_k_words* function simply writes to an output file the top K words obtained from the input text file. A psuedocode for this algorithm is shown in Algorithm 2.

Algorithm 2: `get_top_k_words`

input : allCountPairs, K, file
output: top 10 and 20 words in the text

```
countPairs = compare_by_occurence(countPairs)
allCountPairs = sort_words(countpairs)
if length of allCountPairs < K then
    K = length of allCountPairs
topKWords = first element in allCountPairs array
for wordPair in topKWords do
    write the top K words to an output file
```

C. Inverted Index

This algorithm accepts a list of words, then lists the words the line numbers at which the given words occur in a text. The list can go up to 50.Algorithms 3 and 4 below show how this was implemented was implemented.

Algorithm 3: `Mapper algorithm for inverted_indexing`

input : key (word), value(line text)

```
foreach word do
    change word to lowercase if not a STOP_WORD
      or not a digit then
        return word, line number
```

output: word(key) and line number(value) pair

Algorithm 4: `Reducer algorithm for inverted_indexing`

input : key (word), value(line text)

```
Concatenate all line numbers for a word into a single
  list return word, line numbers list
```

output: word(key) and a list of line numbers(value) pair

As shown above, the inverted index algorithm was implemented using a mapper algorithm and a reducer algorithm. The mapper algorithm accepts a word and a text file. it firstly converts the word to lower case. It then checks if the word is not a digit or a stop word. It then maps all page numbers for each word with the same key for the same word. The reducer algorithm then comes and concatenates all page numbers for a given word into a list, and this is done for each and every word.

IV. Results

All three of the algorithms were tested on the Jaguar1 cluster, using python3 and the Mrs MapReduce (version 0.9) implementation from within a virtual environment. Mrs MapReduce is able to take advantage of distributed parallelisation and can run in either serial, master, slave or bypass mode. However, the program took advantage of only one node of the cluster and was tested in serial mode for the sake of simplicity. In addition, inputs of varying word counts are also used in order to see how the program scales with them. Furthermore, the list of the top 20 words that are not stop words, as well as their respective inverted indices for all the inputs tested can be found in the appendix. Table 1 shows the time taken for each of the three algorithms to process small, large and very large inputs.

Table I
TIME TAKEN FOR EACH ALGORITHM TO PROCESS VARIOUS INPUTS

Algorithm	Small Input (Seconds)	Large Input (Seconds)	Very Large Input (Seconds)
Word-Count	2.91	10.61	43.98
Top-K (K = 20)	3.01	10.64	45.06
Inverted Index	3.07	10.85	46.82

Mrs MapReduce is a high level implementation that automatically manages threads on the developer's behalf. The lack of granular control means that the developer has to trust that the implementation will efficiently scale according to the number of available threads, which is not always ideal. Furthermore the results in Table 1 shows that each algorithm will process a given input in roughly the same amount of time. The Top-K and Inverted Index algorithms will always take slightly longer than the Word-Count algorithm, due to addition time needed to sort the list of words. Sorting is handled by the built-in algorithm available in the Python standard library.

V. Recommendations

The program would greatly benefit from the horizontal scaling offered by distributed parallelisation. Therefore, it is recommended that future implementations of the program make use of the master, slave and bypass modes offered Mrs Mapreduce. In addition, other MapReduce implementations that allow for finer control over how many threads are used can also be explored.

VI. conclusion

In conclusion, the implementation and results of the three algorithms that apply MapReduce have been presented. Mrs MapReduce offers a simple, high level approach to implementing MapReduce, with built-in support for distributed parallelisation. However, the lack of granular thread control makes low-level optimizations difficult.

VII. References

[1] https://www.guru99.com/introduction-to-mapreduce.html

APPENDIX

List of the top K words for K = 20 for the small input:
[**not**, 1335], [**upon**, 818], [**one**, 769], [**states**, 718], [**new**, 712],
[**no**, 681], [**american**, 612], [**said**, 584], [**government**, 541], [**holmes**, 466],
[**man**, 404], [**time**, 401], [**war**, 395], [**will**, 394], [**made**, 378],
[**congress**, 366], [**state**, 355], [**great**, 352], [**two**, 351], [**united**, 349]

The first 50 line numbers of the top 20 words in the small input calculated with inverted index:
not [9, 9, 76, 76, 82, 88, 98, 100, 102, 114, 118, 126, 128, 136, 136, 140, 140, 148, 164, 168, 174, 188, 190, 190, 196, 244, 272, 272, 274, 319, 333, 347, 353, 375, 379, 389, 393, 397, 419, 427, 431, 435, 439, 447, 463, 463, 463, 479, 491, 499],
upon [76, 80, 80, 98, 114, 118, 118, 124, 140, 154, 158, 168, 186, 196, 304, 321, 431, 431, 449, 463, 463, 465, 512, 550, 584, 603, 625, 625, 627, 629, 629, 633, 633, 645, 663, 715, 727, 737, 759, 775, 775, 801, 829, 853, 867, 877, 885, 885, 909, 913],
one [76, 76, 76, 80, 114, 118, 118, 122, 122, 160, 178, 256, 282, 296, 310, 331, 333, 337, 361, 393, 449, 449, 449, 457, 465, 491, 570, 603, 625, 645, 659, 669, 697, 717, 723, 727, 777, 817, 823, 831, 885, 909, 935, 949, 951, 955, 963, 967, 983, 993],
states [1361, 1826, 1852, 2022, 2022, 2094, 3976, 5281, 5284, 5285, 5291, 5295, 5297, 5319, 5320, 5368, 5370, 5415, 5502, 5511, 5521, 5539, 5693, 5694, 5791, 5844, 5890, 5891, 5923, 5937, 5943, 5945, 5947, 5949, 5984, 6499, 6513, 6517, 6518, 6523, 6529, 6586, 6595, 6981, 7087, 7136, 7694, 7754, 7757, 7981],
new [80, 198, 417, 566, 737, 803, 811, 815, 1007, 1096, 1206, 1274, 1729, 1842, 1854, 2008, 2219, 2900, 3037, 3197, 3496, 3506, 3554, 3572, 3574, 3873, 4168, 4262, 4798, 4943, 5186, 5187, 5211, 5238, 5275, 5276, 5276, 5276, 5553, 5611, 5682, 5710, 5745, 5757, 5774, 5791, 5797, 5798, 5844, 5877],
no [122, 134, 168, 182, 194, 206, 248, 250, 272, 365, 455, 455, 465, 526, 546, 596, 613, 677, 711, 763, 783, 801, 801, 805, 817, 829, 841, 853, 911, 913, 915, 967, 979, 999, 999, 1017, 1021, 1031, 1052, 1068, 1082, 1096, 1098, 1108, 1114, 1118, 1120, 1136, 1156, 1160],
american [707, 711, 1988, 2098, 4050, 4064, 4268, 4284, 4290, 4290, 4298, 5583, 5611, 5688, 5699, 5707, 5722, 5754, 5756, 5800, 5856, 5858, 5951, 5966, 5967, 5967, 5975, 6014, 6024, 6027, 6105, 6113, 6154, 6220, 6310, 6491, 6533, 6570, 6688, 6733, 6772, 6952, 6962, 6965, 6966, 6980, 6988, 6998, 7004, 7005],
said [94, 98, 114, 118, 128, 138, 144, 156, 162, 166, 168, 170, 176, 190, 192, 198, 254, 280, 282, 302, 310, 351, 363, 365, 389, 409, 457, 479, 501, 507, 552, 578, 586, 605, 617, 651, 653, 663, 669, 677, 705, 707, 709, 715, 719, 721, 725, 725, 727, 733],
government [2022, 5181, 5739, 5751, 5757, 5774, 5822, 6005, 6016, 6019, 6054, 6058, 6088, 6098, 6107, 6128, 6135, 6156, 6161, 6186, 6297, 6453, 6575, 6871, 6883, 6885, 7169, 7231, 7255, 7315, 7320, 7321, 7446, 7453, 7470, 7522, 7533, 7534, 7538, 7558, 7583, 7590, 7601, 7659, 7669, 7673, 7682, 7727, 7744, 7748],
holmes [0, 26, 39, 49, 76, 78, 78, 80, 94, 128, 132, 142, 146, 156, 162, 166, 170, 176, 180, 182, 188, 192, 196, 198, 254, 282, 304, 310, 319, 419, 421, 423, 449, 463, 463, 465, 465, 503, 507, 514, 530, 552, 554, 558, 564, 564, 566, 568, 570, 578],
man [62, 78, 126, 140, 158, 158, 164, 182, 186, 337, 343, 343, 343, 347, 353, 359, 361, 383, 431, 449, 463, 465, 570, 627, 629, 685, 689, 691, 713, 715, 719, 719, 727, 737, 747, 807, 877, 891, 913, 933, 933, 935, 939, 947, 949, 949, 949, 1021, 1044, 1068],
time [78, 78, 168, 389, 405, 417, 505, 605, 625, 631, 759, 759, 775, 775, 777, 907, 911, 913, 913, 923, 925, 951, 967, 975, 989, 1021, 1100, 1104, 1122, 1248, 1280, 1298, 1335, 1359, 1361, 1397, 1491, 1527, 1533, 1545, 1561, 1581, 1591, 1591, 1591, 1723, 1727, 1727, 1758, 1824],
war [1826, 1852, 2022, 4132, 5627, 5653, 5781, 5804, 5820, 5822, 5823, 5831, 5857, 5862, 5887, 5890, 5891, 5893, 5941, 5955, 7169, 7337, 7538, 7567, 7824, 7831, 7872, 7873, 7873, 7900, 7901, 7902, 7915, 7917, 7922, 7970, 8006, 8016, 8029, 8037, 8291, 8307, 8314, 8423, 8456, 8565, 8790, 9314, 9319, 9339],
will [118, 168, 174, 244, 272, 272, 280, 282, 284, 314, 331, 331, 389, 397, 397, 397, 401, 405, 405, 409, 409, 437, 439, 443, 445, 447, 449, 495, 497, 499, 528, 570, 570, 570, 613, 617, 621, 653, 663, 725, 759, 763, 767, 773, 801, 867, 919, 927, 947, 947],
made [138, 194, 248, 319, 409, 427, 465, 548, 566, 568, 576, 649, 703, 877, 885, 935, 969, 1015, 1025, 1082, 1082, 1090, 1102, 1148, 1198, 1266, 1268, 1294, 1298, 1335, 1355, 1397, 1509, 1549, 1583, 1583, 1729, 1760, 1826, 1842, 1844, 1852, 1880, 2119, 2139, 2143, 2171, 2229, 2237, 2237],
congress [6836, 7950, 8166, 8755, 8756, 8768, 8771, 8774, 9227, 9252, 9258, 9261, 9268, 9276, 9281, 9289, 9291, 9294, 9314, 9315, 9325, 9333, 9339, 9444, 9453, 9461, 9468, 9471, 9544, 9554, 9612, 9614, 9618, 9625, 9633, 9682, 9689, 9692, 9712, 9716, 9735, 9888, 9975, 9978, 10038, 10078, 10088, 10104, 10115, 10117],
state [184, 1353, 1984, 2153, 2225, 2642, 2999, 3011, 3457, 3616, 4728, 5073, 5286, 5291, 6018, 6154, 6418, 7195, 7204, 7207, 7237, 7317, 7337, 7475, 7585, 8530, 8920, 9067, 9195, 9283, 9284, 9328, 9443, 9451, 9549, 9566, 9583, 9592, 9607, 9615, 9625, 9626, 9638, 10369, 10584, 10585, 10585, 10588, 10599, 10637],

great [136, 178, 178, 343, 598, 625, 707, 773, 909, 995, 1044, 1054, 1066, 1196, 1210, 1220, 1248, 1561, 1581, 1591, 1760, 1760, 1774, 1826, 1852, 2030, 2143, 2175, 2221, 2267, 2499, 2507, 2515, 2672, 2834, 3045, 3077, 3101, 3483, 3498, 3518, 3686, 3832, 3902, 3968, 4054, 4212, 4266, 4329, 4373],
two [114, 168, 282, 319, 331, 333, 353, 389, 421, 427, 449, 665, 669, 745, 775, 777, 829, 831, 867, 885, 885, 935, 935, 951, 951, 985, 1019, 1031, 1060, 1086, 1096, 1156, 1206, 1220, 1242, 1274, 1304, 1310, 1337, 1355, 1359, 1361, 1361, 1365, 1373, 1393, 1549, 1553, 1581, 1583],
united [2022, 4736, 5119, 5368, 5370, 5502, 5511, 5521, 5539, 5693, 5694, 5890, 5891, 5923, 5943, 5945, 5947, 5949, 5984, 6116, 6517, 6523, 6529, 6586, 6595, 7087, 7136, 7694, 7754, 7757, 7771, 7914, 7947, 7980, 8166, 8355, 8457, 8463, 8772, 9073, 9171, 9239, 9271, 9372, 9449, 9472, 9546, 9835, 10235, 10356]

List of the top K words for K = 20 for the large input:
[**not**, 6626], [**said**, 3464], [**one**, 3307], [**may**, 2551], [**no**, 2348],
[**prince**, 1898], [**pierre**, 1797], [**now**, 1697], [**will**, 1577], [**time**, 1529],
[**man**, 1525], [**new**, 1211], [**old**, 1180], [**first**, 1177], [**two**, 1138],
[**face**, 1125], [**men**, 1118], [**upon**, 1111], [**see**, 1101], [**natasha**, 1097]

The first 50 line numbers of the top 20 words in the large input calculated with inverted index:
not [9, 9, 76, 76, 82, 88, 98, 100, 102, 114, 118, 126, 128, 136, 136, 140, 140, 148, 164, 168, 174, 188, 190, 190, 196, 244, 272, 272, 274, 319, 333, 347, 353, 375, 379, 389, 393, 397, 419, 427, 431, 435, 439, 447, 463, 463, 463, 479, 491, 499],
said [94, 98, 114, 118, 128, 138, 144, 156, 162, 166, 168, 170, 176, 190, 192, 198, 254, 280, 282, 302, 310, 351, 363, 365, 389, 409, 457, 479, 501, 507, 552, 578, 586, 605, 617, 651, 653, 663, 669, 677, 705, 707, 709, 715, 719, 721, 725, 725, 727, 733],
one [76, 76, 76, 80, 114, 118, 118, 122, 122, 160, 178, 256, 282, 296, 310, 331, 333, 337, 361, 393, 449, 449, 449, 457, 465, 491, 570, 603, 625, 645, 659, 669, 697, 717, 723, 727, 777, 817, 823, 831, 885, 909, 935, 949, 951, 955, 963, 967, 983, 993],
may [15, 114, 118, 152, 152, 164, 164, 166, 166, 168, 174, 268, 393, 409, 415, 427, 459, 491, 495, 499, 570, 570, 713, 867, 931, 941, 943, 951, 979, 1011, 1064, 1068, 1068, 1210, 1234, 1242, 1268, 1298, 1302, 1302, 1318, 1365, 1365, 1365, 1369, 1369, 1373, 1393, 1415, 1477],
no [122, 134, 168, 182, 194, 206, 248, 250, 272, 365, 455, 455, 465, 526, 546, 596, 613, 677, 711, 763, 783, 801, 801, 805, 817, 829, 841, 853, 911, 913, 915, 967, 979, 999, 999, 1017, 1021, 1031, 1052, 1068, 1082, 1096, 1098, 1108, 1114, 1118, 1120, 1136, 1156, 1160],
prince [236, 8094, 15825, 25738, 26923, 54331, 54341, 54350, 54355, 54376, 54383, 54389, 54394, 54434, 54446, 54453, 54470, 54480, 54483, 54494, 54502, 54530, 54541, 54548, 54573, 54579, 54637, 54640, 54725, 54800, 54809, 54830, 54900, 54913, 54929, 54930, 54932, 54943, 54945, 54948, 54959, 54963, 54971, 54973, 54977, 54982, 54991, 54992, 54999, 55001],
pierre [54654, 54660, 54664, 54669, 54678, 54697, 54702, 54863, 54876, 54879, 54889, 54929, 54937, 54939, 54956, 54961, 54970, 55132, 55149, 55154, 55175, 55179, 55186, 55195, 55205, 55220, 55245, 55253, 55266, 55281, 55343, 55355, 55425, 55446, 55456, 55471, 55478, 55483, 55486, 55508, 55514, 55529, 55538, 55541, 55560, 55568, 55575, 55591, 55608, 55611],
now [80, 114, 136, 198, 236, 361, 389, 389, 423, 423, 447, 457, 463, 491, 493, 507, 580, 625, 655, 659, 663, 667, 669, 711, 713, 747, 907, 911, 955, 975, 987, 1062, 1074, 1102, 1176, 1178, 1210, 1220, 1242, 1274, 1278, 1290, 1294, 1300, 1302, 1308, 1355, 1459, 1483, 1519],
will [118, 168, 174, 244, 272, 272, 280, 282, 284, 314, 331, 331, 389, 397, 397, 397, 401, 405, 405, 409, 409, 437, 439, 443, 445, 447, 449, 495, 497, 499, 528, 570, 570, 570, 613, 617, 621, 653, 663, 725, 759, 763, 767, 773, 801, 867, 919, 927, 947, 947],
time [78, 78, 168, 389, 405, 417, 505, 605, 625, 631, 759, 759, 775, 775, 777, 907, 911, 913, 913, 923, 925, 951, 967, 975, 989, 1021, 1100, 1104, 1122, 1248, 1280, 1298, 1335, 1359, 1361, 1397, 1491, 1527, 1533, 1545, 1561, 1581, 1591, 1591, 1591, 1723, 1727, 1727, 1758, 1824],
man [62, 78, 126, 140, 158, 158, 164, 182, 186, 337, 343, 343, 343, 347, 353, 359, 361, 383, 431, 449, 463, 465, 570, 627, 629, 685, 689, 691, 713, 715, 719, 719, 727, 737, 747, 807, 877, 891, 913, 933, 933, 935, 939, 947, 949, 949, 949, 1021, 1044, 1068],
well [80, 108, 126, 194, 323, 331, 347, 365, 421, 465, 548, 566, 568, 639, 647, 651, 669, 673, 703, 707, 715, 733, 739, 747, 749, 759, 773, 775, 777, 805, 869, 913, 917, 931, 955, 975, 1013, 1031, 1039, 1090, 1100, 1108, 1116, 1120, 1144, 1146, 1212, 1212, 1216, 1242],
new [80, 198, 417, 566, 737, 803, 811, 815, 1007, 1096, 1206, 1274, 1729, 1842, 1854, 2008, 2219, 2900, 3037, 3197, 3496, 3506, 3554, 3572, 3574, 3873, 4168, 4262, 4798, 4943, 5186, 5187, 5211, 5238, 5275, 5276, 5276, 5276, 5553, 5611, 5682, 5710, 5745, 5757, 5774, 5791, 5797, 5798, 5844, 5877],
old [78, 321, 483, 566, 711, 939, 1054, 1204, 1206, 1298, 1318, 1355, 1359, 1503, 1561, 1707, 1719, 1723, 1729, 1741, 1747, 1751, 1760, 1828, 1834, 1884, 1930, 2094, 2137, 2145, 2171, 2293, 2429, 2491, 2564, 2602, 2630, 2700, 3037, 3039, 3039, 3041, 3055, 3093, 3101, 3143, 3169, 3179, 3193, 3219],

first [8, 78, 345, 651, 663, 683, 767, 823, 831, 831, 883, 945, 951, 979, 991, 1021, 1050, 1074, 1104, 1124, 1148, 1156, 1216, 1216, 1220, 1296, 1302, 1310, 1369, 1725, 1729, 1828, 1950, 1988, 1990, 2024, 2036, 2038, 2088, 2119, 2137, 2143, 2237, 2265, 2525, 2529, 2580, 2682, 3023, 3041],
two [114, 168, 282, 319, 331, 333, 353, 389, 421, 427, 449, 665, 669, 745, 775, 777, 829, 831, 867, 885, 885, 935, 935, 951, 951, 985, 1019, 1031, 1060, 1086, 1096, 1156, 1206, 1220, 1242, 1274, 1304, 1310, 1337, 1355, 1359, 1361, 1361, 1365, 1373, 1393, 1549, 1553, 1581, 1583],
face [140, 158, 158, 186, 272, 321, 347, 449, 483, 514, 731, 735, 797, 853, 873, 913, 965, 993, 993, 1027, 1074, 1082, 1104, 1202, 1204, 1216, 1288, 1294, 1300, 1304, 1495, 1533, 1533, 1583, 1583, 1585, 1591, 1635, 1695, 1707, 1729, 1735, 1774, 1830, 1842, 1844, 1900, 2117, 2119, 2161],
men [196, 272, 331, 421, 421, 449, 449, 659, 689, 689, 693, 707, 707, 709, 711, 935, 979, 1003, 1310, 1355, 1387, 1389, 1583, 1729, 1802, 1988, 2004, 2012, 2024, 2028, 2199, 2237, 2926, 3041, 3500, 4383, 4716, 4716, 5774, 5964, 6034, 6071, 6083, 6095, 6116, 6135, 6173, 6352, 6353, 6375],
upon [76, 80, 80, 98, 114, 118, 118, 124, 140, 154, 158, 168, 186, 196, 304, 321, 431, 431, 449, 463, 463, 465, 512, 550, 584, 603, 625, 625, 627, 629, 629, 633, 633, 645, 663, 715, 727, 737, 759, 775, 775, 801, 829, 853, 867, 877, 885, 885, 909, 913],
see [80, 82, 92, 94, 98, 102, 198, 337, 339, 345, 423, 463, 487, 564, 568, 649, 651, 691, 707, 711, 713, 715, 747, 775, 775, 799, 827, 893, 901, 903, 911, 939, 979, 1001, 1001, 1009, 1021, 1027, 1076, 1082, 1092, 1110, 1118, 1120, 1180, 1220, 1224, 1234, 1238, 1304]

List of the top K words for K = 20 for the very large input:
[**not**, 20448], [**one**, 14804], [**also**, 13442], [**can**, 11935], [**many**, 9645],
[**used**, 8709], [**first**, 8602], [**two**, 7982], [**may**, 7587], [**time**, 7310],
[**new**, 6637], [**however**, 6633], [**no**, 6501], [**will**, 6238], [**use**, 5676]
[**often**, 5586], [**years**, 5009], [**called**, 4975], [**people**, 4965], [**world**, 4504]

The first 50 line numbers of the top 20 words in the very large input calculated with inverted index:
one [22, 39, 50, 54, 59, 83, 90, 92, 93, 94, 96, 98, 101, 108, 109, 111, 124, 124, 126, 126, 138, 176, 182, 208, 209, 215, 232, 237, 239, 244, 263, 264, 267, 283, 292, 293, 296, 313, 326, 343, 346, 346, 347, 348, 348, 348, 348, 356, 356, 359],
also [0, 2, 2, 44, 47, 48, 55, 57, 61, 63, 65, 66, 68, 76, 87, 94, 96, 96, 100, 100, 119, 130, 137, 153, 157, 162, 192, 215, 248, 251, 253, 261, 280, 293, 324, 331, 336, 339, 344, 349, 355, 356, 383, 391, 397, 399, 405, 433, 433, 465],
can [8, 36, 49, 69, 77, 82, 83, 84, 88, 94, 94, 95, 96, 96, 96, 98, 99, 99, 99, 100, 100, 101, 102, 102, 113, 114, 114, 122, 131, 131, 133, 136, 150, 152, 156, 157, 160, 173, 176, 198, 239, 261, 270, 286, 348, 356, 356, 357, 360, 388],
many [7, 11, 17, 19, 25, 27, 29, 40, 45, 46, 51, 51, 54, 55, 60, 68, 70, 77, 81, 85, 98, 99, 101, 114, 115, 124, 124, 126, 127, 127, 128, 128, 128, 130, 130, 131, 132, 148, 157, 174, 176, 207, 225, 263, 287, 289, 300, 325, 372, 391],
used [0, 0, 0, 50, 55, 76, 78, 82, 82, 94, 111, 113, 134, 150, 176, 184, 184, 234, 286, 374, 381, 391, 435, 455, 522, 537, 537, 555, 559, 576, 616, 659, 700, 719, 720, 739, 753, 793, 804, 812, 815, 833, 856, 857, 861, 861, 863, 866, 877, 921],
first [0, 7, 12, 12, 14, 15, 18, 78, 79, 85, 92, 167, 171, 189, 229, 236, 246, 249, 291, 335, 340, 351, 354, 363, 381, 386, 435, 440, 442, 494, 521, 543, 546, 546, 557, 570, 571, 576, 584, 599, 614, 614, 643, 644, 662, 687, 729, 779, 789, 801],
two [15, 16, 50, 113, 119, 127, 135, 147, 177, 184, 200, 205, 208, 249, 273, 274, 280, 297, 357, 385, 385, 391, 399, 420, 425, 447, 463, 495, 495, 521, 600, 643, 648, 656, 668, 673, 712, 763, 785, 785, 792, 796, 797, 816, 827, 831, 884, 899, 899, 913],
may [18, 42, 75, 86, 88, 88, 90, 93, 94, 95, 95, 95, 96, 96, 97, 97, 99, 100, 112, 114, 114, 123, 137, 138, 157, 173, 173, 212, 233, 272, 274, 275, 294, 331, 349, 351, 356, 359, 375, 384, 385, 385, 407, 414, 416, 419, 419, 420, 422, 424],
time [5, 15, 23, 80, 96, 96, 100, 167, 208, 233, 233, 233, 234, 246, 254, 254, 261, 261, 265, 276, 286, 326, 332, 342, 359, 370, 372, 389, 390, 390, 391, 397, 489, 490, 541, 560, 560, 562, 636, 641, 641, 657, 658, 662, 663, 684, 687, 710, 711, 711],
new [13, 42, 46, 54, 102, 129, 159, 159, 168, 233, 259, 261, 313, 314, 314, 319, 320, 321, 379, 441, 468, 476, 495, 495, 518, 528, 539, 539, 580, 585, 585, 592, 594, 594, 594, 603, 611, 664, 664, 664, 693, 695, 707, 709, 730, 731, 745, 746, 751, 778],
however [2, 17, 23, 34, 60, 60, 77, 79, 82, 84, 87, 90, 131, 136, 151, 160, 206, 208, 208, 226, 229, 230, 254, 254, 268, 278, 339, 359, 360, 365, 381, 385, 390, 398, 483, 492, 534, 550, 577, 614, 614, 634, 639, 661, 672, 693, 706, 711, 731, 776],
no [6, 7, 23, 92, 93, 114, 116, 117, 123, 138, 139, 145, 175, 184, 188, 240, 246, 254, 254, 262, 276, 281, 293, 393, 405, 419, 421, 426, 438, 443, 508, 508, 557, 557, 569, 593, 603, 604, 604, 617, 618, 623, 634, 638, 650, 681, 682, 682, 691, 695],
will [5, 22, 39, 59, 60, 71, 75, 83, 98, 99, 101, 128, 137, 156, 157, 244, 244, 261, 261, 270, 270, 270, 280, 335, 358, 364, 366, 399, 406, 406, 408, 413, 413, 413, 419, 421, 424, 426, 463, 470, 481, 484, 501, 516, 533, 533, 555, 573, 573, 573],
use [1, 7, 8, 8, 50, 53, 55, 59, 81, 81, 101, 103, 173, 179, 184, 184, 185, 187, 288, 339, 390, 391, 463, 536, 538, 548, 557, 614, 682, 715, 724, 759, 805, 813, 816, 821, 822, 823, 856, 858, 863, 867, 888, 893, 921, 973, 986, 995, 998, 1001],
often [19, 21, 38, 39, 39, 58, 60, 82, 83, 84, 85, 86, 93, 93, 95, 95, 96, 113, 119, 130, 130, 131, 132, 205, 224, 246, 263, 280, 348, 372, 374, 405, 415, 422, 423, 504, 634, 640, 662, 671, 682, 695, 700, 706, 711, 712, 714, 734, 734, 735],

years [14, 37, 63, 76, 76, 80, 111, 120, 188, 217, 231, 232, 232, 257, 274, 274, 289, 302, 302, 334, 335, 448, 458, 480, 495, 497, 504, 528, 532, 539, 554, 662, 671, 713, 721, 725, 742, 746, 747, 766, 806, 822, 827, 831, 879, 886, 899, 904, 923, 931],
called [8, 11, 12, 14, 17, 47, 49, 234, 255, 339, 363, 371, 415, 419, 424, 439, 456, 477, 507, 509, 531, 632, 632, 634, 695, 788, 795, 801, 832, 854, 858, 874, 899, 913, 978, 1113, 1137, 1172, 1214, 1223, 1238, 1240, 1246, 1248, 1253, 1255, 1266, 1286, 1286, 1296],
people [22, 55, 55, 58, 59, 79, 84, 85, 85, 88, 92, 93, 93, 93, 93, 93, 94, 94, 94, 94, 95, 95, 95, 95, 95, 95, 96, 96, 97, 104, 114, 114, 115, 115, 123, 127, 130, 131, 131, 131, 132, 134, 155, 193, 193, 193, 193, 198, 198, 242],
world [45, 97, 127, 157, 209, 280, 293, 338, 348, 369, 382, 398, 416, 443, 457, 462, 463, 519, 521, 526, 528, 528, 528, 530, 533, 544, 545, 550, 560, 568, 570, 570, 570, 583, 585, 591, 591, 592, 592, 592, 597, 604, 604, 604, 604, 604, 605, 610, 619, 624]