Report

Q1:

- I used re to remove symbols and other non-useful characters from the text data.
- I used nltk library to then remove punctuation, stop words and blank spaces from the text.
- 5 examples from different files.

```
File file270.txt content is:
let low price fool incredible device free mixing software years since tracked anything back everything solid state analogue scarlett solo moderate computer large monitor stress large m


File file215.txt content is:
great taylor 150e 12string guitar issue wand came tear microfiber duster


File file701.txt content is:
absolute junk doesnt worth amazon shipment price dont buy circumstances fire hazard 3 4 received dead arrival even missing parts rattle inside one working also something rattling insid


File file693.txt content is:
works perfectly ukuleles guitars simple use instant perfect tuner


File file302.txt content is:
case awesome molded interior fits fender jazz bass like glove feel secured times second case ive owned quality superb balances much better hand carrying last case bit front heavy also
```

Q2:

- I used the default dictionary to store the inverted indexes.
- Then I used a for loop to store all the document ids with corresponding tokens in the dictionary.
- This is an example of the view of invert index

```
way: [1, 108, 13, 139, 144, 152, 157, 194, 214, 228, 249, 269, 279, 293, 305, 326, 349, 356, 373, 383, 384, 386, 406, 413, 428, 43, 438, 469, 483, 487, 49, 512, 537, 546, 549,
great: [1, 100, 101, 102, 103, 104, 105, 108, 109, 115, 119, 120, 121, 122, 124, 127, 129, 13, 131, 132, 134, 137, 139, 140, 147, 148, 152, 156, 158, 167, 176, 183, 187, 188, 1
go: [1, 100, 103, 106, 117, 136, 139, 144, 150, 172, 19, 191, 196, 212, 230, 24, 248, 254, 257, 332, 335, 353, 363, 366, 382, 388, 407, 410, 413, 414, 42, 423, 429, 454, 464, 4
floating: [1, 245, 430]
loving: [1, 254, 391, 723]
good: [1, 103, 106, 110, 111, 115, 118, 13, 137, 141, 143, 154, 155, 157, 159, 16, 160, 162, 163, 164, 166, 172, 174, 175, 176, 179, 18, 189, 19, 2, 204, 207, 210, 217, 220, 23
strat: [1, 149, 163, 197, 241, 245, 25, 253, 345, 353, 380, 396, 400, 422, 440, 455, 457, 469, 519, 529, 559, 565, 579, 611, 626, 650, 652, 691, 801, 838, 853, 90, 940, 978, 99
bridge: [1, 108, 117, 139, 168, 244, 249, 257, 283, 328, 353, 361, 454, 464, 484, 503, 541, 579, 598, 61, 620, 700, 725, 748, 756, 770, 787, 801, 806, 881, 916, 927, 998]
vintage: [1, 150, 197, 278, 422, 439, 494, 51, 597, 638, 674, 725, 737, 827, 847, 895, 907, 936]
tension: [1, 116, 143, 507, 514, 751, 839, 980]
springs: [1, 272, 469, 806, 937]
stability: [1, 115, 382, 521, 759]
want: [1, 102, 114, 121, 129, 143, 191, 213, 216, 228, 23, 232, 249, 293, 299, 307, 320, 325, 33, 343, 353, 359, 366, 39, 390, 396, 399, 419, 429, 43, 439, 490, 5, 519, 529, 54
product: [10, 105, 115, 121, 134, 142, 152, 167, 168, 171, 178, 180, 186, 19, 196, 208, 217, 225, 228, 229, 241, 251, 26, 269, 282, 298, 30, 326, 337, 35, 358, 363, 372, 374, 3
frame: [10, 352, 466, 684, 849]
bend: [10, 212, 70, 833, 994]
making: [10, 111, 143, 158, 172, 212, 249, 336, 338, 425, 444, 5, 744, 769, 871]
stand: [10, 120, 122, 135, 150, 153, 172, 18, 185, 208, 210, 234, 267, 269, 283, 286, 3, 303, 308, 326, 366, 374, 382, 408, 42, 425, 44, 442, 458, 459, 462, 465, 466, 47, 475,
angle: [10, 103, 301, 305, 338, 523, 579, 648, 663, 678, 720, 765, 772, 778, 79, 808, 882]
weird: [10, 499, 690, 721, 794, 886, 923]
photos: [10, 117, 126, 222, 290, 294, 319, 449, 460, 489, 494, 525, 55, 627, 694, 726, 790]
support: [10, 267, 270, 282, 42, 499, 646, 649, 778, 875, 937, 993]
vertical: [10]
becoming: [10, 254, 491]
assembled: [10, 458, 728]
```

```
def andOperator(set1, set2):
    return set1 and set2

def orOperator(set1, set2):
    return set1 or set2

def andNotOperator(set1, set2):
    return set1 - set2

def orNotOperator(set1, set2):
    return set1 or  (vocab - set2)
```

- I further used these self defined functions for doing the given algorithm
- Then according to the example given in the assignment sheet I wrote a code that does the work for the user.

```
Query 1 :  guitar AND player
Number of documents retreieved for query:  13
Documents retreived:  769.txt,131.txt,68.txt,390.txt,174.txt,401.txt,948.txt,854.txt,23.txt,569.txt,539.txt,279.txt,183.txt
```

- Example output for the code

Q3.

- I used a dictionary of dictionaries to store the positional index. Following a similar pattern as Q2 , I got a dictionary that looks like this.

```
loving: defaultdict(<class 'list'>, {1: [0], 254: [19], 391: [6], 723: [6]})
vintage: defaultdict(<class 'list'>, {1: [1, 3], 150: [23], 197: [8, 66], 278: [9], 422: [14], 439: [8, 57], 494: [16], 51: [57], 597: [55], 638: [120], 674: [5
springs: defaultdict(<class 'list'>, {1: [2, 12], 272: [2, 19], 469: [57], 806: [12], 937: [47, 100]})
strat: defaultdict(<class 'list'>, {1: [4], 149: [7], 163: [113, 146], 197: [22], 241: [6], 245: [10], 25: [5, 11], 253: [44], 345: [12], 353: [106], 380: [15,
good: defaultdict(<class 'list'>, {1: [5], 103: [106], 106: [55], 110: [10], 111: [44], 115: [5], 118: [117], 13: [48, 55], 137: [3], 141: [32], 143: [36], 154
tension: defaultdict(<class 'list'>, {1: [6], 116: [24], 143: [25], 507: [110], 514: [42, 168], 751: [3], 839: [138], 980: [24]})
great: defaultdict(<class 'list'>, {1: [7], 100: [3], 101: [2], 102: [40], 103: [115], 104: [10], 105: [82], 108: [11], 109: [4], 115: [2], 119: [2], 120: [9],
stability: defaultdict(<class 'list'>, {1: [8], 115: [67], 382: [19], 521: [140], 759: [20]})
floating: defaultdict(<class 'list'>, {1: [9], 245: [136], 430: [10]})
bridge: defaultdict(<class 'list'>, {1: [10], 108: [136], 117: [30], 139: [23], 168: [38], 244: [64], 249: [106], 257: [52], 283: [37], 328: [93], 353: [52], 3
want: defaultdict(<class 'list'>, {1: [11], 102: [56], 114: [13, 24, 58], 121: [8], 129: [112], 143: [76], 191: [4], 213: [15], 216: [133], 228: [75], 23: [16],
way: defaultdict(<class 'list'>, {1: [13], 108: [66], 13: [49], 139: [31], 144: [69], 152: [11], 157: [70], 194: [13], 214: [111], 228: [1], 249: [10], 269: [15
go: defaultdict(<class 'list'>, {1: [14], 100: [76], 103: [118], 106: [57], 117: [69], 136: [61], 139: [27], 144: [71], 150: [92], 172: [116], 19: [119], 191: [
awesome: defaultdict(<class 'list'>, {10: [0], 105: [87], 108: [120], 152: [61], 178: [32], 218: [0], 239: [128], 241: [27], 279: [0], 302: [3], 307: [6], 314:
stand: defaultdict(<class 'list'>, {10: [1], 120: [4, 37, 66], 122: [5, 23], 135: [23], 150: [1, 60, 101], 153: [21], 172: [26, 47, 85, 99, 139, 170], 18: [150
tip: defaultdict(<class 'list'>, {10: [2], 257: [62], 360: [5], 454: [28], 600: [9], 679: [45], 824: [24], 978: [44], 991: [64]})
bottom: defaultdict(<class 'list'>, {10: [3], 107: [49], 118: [71, 111], 123: [31], 184: [26], 204: [30], 222: [109], 267: [167], 334: [36], 336: [177], 434: [6
part: defaultdict(<class 'list'>, {10: [4], 117: [39], 135: [34], 174: [24], 353: [45], 370: [46], 440: [101], 45: [31], 482: [23], 483: [141], 484: [40], 492:
supports: defaultdict(<class 'list'>, {10: [5], 267: [148, 170], 301: [41], 49: [84], 628: [19]})
guitar: defaultdict(<class 'list'>, {10: [6, 11, 28], 100: [73], 104: [7], 106: [92], 107: [122], 11: [111, 122], 110: [16], 114: [86], 116: [20], 118: [15, 63
weird: defaultdict(<class 'list'>, {10: [7], 499: [14], 690: [73], 721: [51], 794: [50], 886: [86], 923: [66]})
angle: defaultdict(<class 'list'>, {10: [8], 103: [34], 301: [105], 305: [22], 338: [79], 523: [104], 579: [57], 648: [9, 35], 663: [75], 678: [60], 720: [14, 5
arrived: defaultdict(<class 'list'>, {10: [9], 143: [0], 239: [34], 338: [46], 341: [31], 382: [42], 405: [76], 45: [10], 505: [13], 61: [25], 65: [84], 657: [1
making: defaultdict(<class 'list'>, {10: [10, 25], 111: [69], 143: [29], 158: [48], 172: [171], 212: [74], 249: [90], 336: [158], 338: [26], 425: [50], 444: [12
```

- I have used json file format to save this pickle format as its a lambda type default dictionary
- Loading it gives the correct output

- This is an example query that I ran.
-
```
Number of documents retrieved for query 1 using positional index: 2
Names of documents retrieved for query 1 using positional index: ['249.txt', '361.txt']
```