positional indexes

```
doc4='''hello its information retrieval A paragraph is a series of sentences that are organized and coherent, and are all related to a single topic. Almost every piece of writing you do that is longer than a few sentences should be organized into paragraphs.
       system contains boolean and phrasal queries'''
print(doc4)
doc5='''Information retrieval (IR) in computing and information science is the process of obtaining
      information system resources that are relevant to an information need from a collection of those resources. Searches can be based on full-text or other content-based indexing. Information retrieval is the science[1] of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds'''
print(doc5)
doc6='''An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines.
 In information retrieval a query does not uniquely identify a single object in the collection.
 Instead, several objects may match the query, perhaps with different degrees of relevance.'''
print(doc6)
l2=[doc4.lower(),doc5.lower(),doc6.lower()]
k=[]
p=l2[0].split(' ')
#print(p)
for i in p:
  k.append(i.strip())
#print(k)
#trying with the stop words
stopwords1=['an','','is','be','of',',','its','and','A','ourselves','are','the','is','and','or','hers','between','yourself','but','again','there','about','once','during','out','very','having','is','a','in','the','by','on','so','!','@','#','$','%','.','^','&','*','(',')','-','_','+','=','/']##\??

#‘with’, ‘they’, ‘own’, ‘an’, ‘be’, ‘some’, ‘for’, ‘do’, ‘its’, ‘yours’, ‘such’, ‘into’, ‘of’, ‘most’, ‘itself’, ‘other’, ‘off’, ‘is’, ‘s’,
# ‘am’, ‘or’, ‘who’, ‘as’, ‘from’, ‘him’, ‘each’, ‘the’, ‘themselves’, ‘until’, ‘below’, ‘are’, ‘we’, ‘these’, ‘your’, ‘his’, ‘through’,
# ‘don’, ‘nor’, ‘me’, ‘were’, ‘her’, ‘more’, ‘himself’, ‘this’, ‘down’, ‘should’, ‘our’, ‘their’, ‘while’, ‘above’, ‘both’, ‘up’, ‘to’, ‘ours’, ‘had’, ‘she’, ‘all’, ‘no’, ‘when’, ‘at’, ‘any’, ‘before’, ‘them’, ‘same’, ‘and’,
#‘been’, ‘have’, ‘in’, ‘will’, ‘on’, ‘does’, ‘yourselves’, ‘then’, ‘that’, ‘because’, ‘what’, ‘over’, ‘why’, ‘so’, ‘can’, ‘did’, ‘not’, ‘now’, ‘under’, ‘he’, ‘you’, ‘herself’, ‘has’, ‘just’, ‘where’, ‘too’, ‘only’, ‘myself’, ‘which’, ‘those’, ‘i’, ‘after’, ‘few’, ‘whom’,
#‘t’, ‘being’, ‘if’, ‘theirs’, ‘my’, ‘against’, ‘a’, ‘by’, ‘doing’, ‘it’, ‘how’, ‘further’, ‘was’, ‘here’, ‘than’]
stopwords=sorted(stopwords1)
print(stopwords)
r=[]
# Remove multiple empty spaces from string List
# Using loop + strip()
for i in range(0,len(l2)):
  z=l2[i].split(" ")
  res = []
  for ele in z:
    if ele=='':
      continue
    else:
        k=ele.strip()
        res.append(k.lower())
  r.extend(res)
print(r)
u=[]
for i in r:
  if  i.lower()  not in stopwords and i.upper() not in stopwords and i not in stopwords:
       u.append(i)
print(u)
u=sorted(u)
print(u)
l3=[doc4.lower(),doc5.lower(),doc6.lower()]
d1={}
for i in u:
  d1.update({i:{}})
  for j in range(0,len(l2)):
      phi=l2[j].split(" ")
      #print(phi)
      if(i in phi):
            d1[i].update({j:[]})
            c_ount=phi.count(i)
            print(c_ount)
            for k in range(0,c_ount):
                ind_ex=phi.index(i)
                d1[i][j].append(ind_ex)
                phi.remove(i)

print(d1)
```

```
{'all': {0: [18]}, 'almost': {0: [24]}, 'also': {1: [62]}, 'based': {1: [38]}, 'begins': {2: [4]}, 'boolean': {0:
[51]}, 'can': {1: [36]}, 'coherent,': {0: [15]}, 'collection': {1: [31]}, 'collection.': {}, 'computing': {1:
[4]}, 'contains': {0: [50]}, 'content-based': {1: [43]}, 'data,': {1: [69]}, 'databases': {1: [72]},
'degrees': {2: [54]}, 'describes': {1: [68]}, 'different': {2: [53]}, 'do': {0: [30]}, 'document,': {1: [56]},
'documents': {1: [59]}, 'does': {2: [34]}, 'engines.': {}, 'enters': {2: [8]}, 'every': {0: [25]}, 'example':
{2: [22]}, 'few': {0: [36]}, 'for': {1: [52, 57, 62, 68], 2: [21]}, 'formal': {2: [16]}, 'from': {1: [29]}, 'full-
text': {1: [40]}, 'hello': {0: [0]}, 'identify': {2: [37]}, 'images': {1: [75]}, 'indexing.': {1: [44]},
'information': {0: [2], 1: [0, 5, 17, 24, 41, 48], 2: [1, 18, 28]}, 'instead,': {2: [44]}, 'into': {0: [41], 2:
[11]}, 'longer': {0: [33]}, 'match': {2: [48]}, 'may': {2: [47]}, 'metadata': {1: [66]}, 'need': {1: [28]},
'needs,': {2: [20]}, 'not': {2: [35]}, 'object': {2: [40]}, 'objects': {2: [46]}, 'obtaining': {1: [12]},
'organized': {0: [13, 39]}, 'other': {1: [42]}, 'paragraph': {0: [5]}, 'paragraphs.': {}, 'perhaps': {2:
[51]}, 'phrasal': {0: [53]}, 'piece': {0: [26]}, 'process': {1: [10], 2: [3]}, 'queries': {0: [54], 2: [14]},
'query': {2: [10, 32]}, 'query,': {2: [50]}, 'related': {0: [19]}, 'relevance.': {2: [56]}, 'relevant': {1:
[24]}, 'resources': {1: [21]}, 'resources.': {1: [34]}, 'retrieval': {0: [3], 1: [1, 45], 2: [2, 30]}, 'science':
{1: [7]}, 'science[1]': {1: [49]}, 'search': {2: [23, 26]}, 'searches': {1: [35]}, 'searching': {1: [51, 56,
```

61]}, 'sentences': {0: [10, 36]}, 'series': {0: [8]}, 'several': {2: [45]}, 'should': {0: [38]}, 'single': {0:
[22], 2: [39]}, 'sounds': {1: [77]}, 'statements': {2: [17]}, 'strings': {2: [24]}, 'system': {0: [49], 1:
[20]}, 'system.': {2: [13]}, 'texts,': {1: [74]}, 'than': {0: [34]}, 'that': {0: [11, 30], 1: [22, 66]},
'themselves,': {1: [60]}, 'those': {1: [33]}, 'to': {0: [20], 1: [25]}, 'topic.': {0: [23]}, 'uniquely': {2: [36]},
'user': {2: [7]}, 'web': {2: [26]}, 'when': {2: [5]}, 'with': {2: [52]}, 'writing': {0: [28]}, 'you': {0: [29]}}