you should import PorterStemmer

```
In [1]: from nltk.stem import PorterStemmer
```

```
In [71]: class IR():
             def __init__(self):
                 self.inittoken_list = []
                 self.stemmed_list = []
                 self.stopwarded_list = []
                 self.punctuation_list = [".", "'", '"', "?"]
                 self.poter = PorterStemmer()
                 self.stopward_list = []
                 self.stopward_dic = {"ALL":[]}
                 self.stopward_flag = [0, 0]
                 return None

             def read_file(self, storage_place):
                 if len(self.inittoken_list) != 0:
                     return "you have already put some data in here"
                 ## vertify type of input
                 if not isinstance(storage_place, str):
                     print("you should input where you store your document in string type.")
                     return False
                 ## make document in to a list of list of strings, seperated in lines
                 storage_place = storage_place.strip("/")
                 document_list = open(storage_place, 'rt').readlines()

                 ## make document into a single list of string
                 for line in document_list:
                     start_flag = 0
                     for stop_flag in range( len(line)):
                         if line[ stop_flag] == ' ' :
                             self.inittoken_list.append( ''.join(line[ start_flag : stop_flag]))
                             start_flag = stop_flag + 1
                         if line[-1] == '.' and stop_flag == len(line)-1: # check the last word
                             self.inittoken_list.append( ''.join(line[ start_flag : -1]))
                 return self.inittoken_list

             def stemming(self):
                 if len(self.stemmed_list) != 0:
                     return "you have already stemmed your document"
                 for voca_index in  range( len( self.inittoken_list)):
                     for pun in self.punctuation_list:
                         if pun in self.inittoken_list[voca_index] :
                             self.inittoken_list[voca_index] = self.inittoken_list[voca_index].replace(pun, '')
                 ## lowercast and stem the document, which poter.stem() will auto-lowercast
                 for voca in self.inittoken_list:
                     self.stemmed_list.append( self.poter.stem( voca))
                 return self.stemmed_list

             def stopwarding(self):
                 ## check for if stopward list create or not
                 if len(self.stopward_list) == 0:
                     init_stopward_list = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'l
l", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself',
'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that',
"that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'doe
s', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'wit
h', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',
'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how',
'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so',
'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're',
've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "has
n't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't",
'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
                     self.stopward_adding( init_stopward_list)

                 ## check for the state of stopward, same for stopwarded, different for not yet
                 if self.stopward_flag[0] == self.stopward_flag[1]:
                     return "you have already stopwarded your document"
                 else:
                     if self.stopward_flag[1] > 0:
                         dest_document = self.stopwarded_list
                     else:
                         dest_document = self.stemmed_list

                 ## stopward
                 for voca_index in range( len( dest_document)):
                     if dest_document[voca_index] in self.stopward_dic:
                         self.stopward_dic['ALL'].append( voca_index)
                         self.stopward_dic[ dest_document[voca_index]].append(voca_index)
                     else:
                         self.stopwarded_list.append( dest_document[voca_index])
                 self.stopward_flag[1] = self.stopward_flag[1] +1
                 return self.stopwarded_list

             def stopward_adding(self, new_ward_list):
                 ## check for type of list
                 if not isinstance(new_ward_list, list):
                     print("want a list. in stopward_adding")
                     return False
                 for stopward in new_ward_list:
                     ## check for type of each ward in list
                     if not isinstance(stopward, str):
                         print("want a list of string. in stopward_adding")
                         return False
                     ## stem and add
                     stemmed_stopward = self.poter.stem( stopward)
                     if not stemmed_stopward in self.stopward_dic:
                         self.stopward_list.append( stemmed_stopward)
                         self.stopward_dic.update({stemmed_stopward: []})
                 self.stopward_flag[0] = self.stopward_flag[0] +1
                 return 0

             def punctuation_adding(self, new_pun):

                 return 0

             def save_result(self):
                 with open("R09725049_result.txt" , "w") as text_file:
                     text_file.write(str(self.stopwarded_list))
                 return "file saved"
```

Create an object of IR

```
In [72]: temp1 = IR()
```

use .read_file(file_route) to import document you want

(type of file_route should be string)

```
In [73]: temp1.read_file('D:/Desktop/IR/PA1/pa1.txt')
```

```
Out[73]: ['And',
          'Yugoslav',
          'authorities',
          'are',
          'planning',
          'the',
          'arrest',
          'of',
          'eleven',
          'coal',
          'miners',
          'and',
          'two',
          'opposition',
          'politicians',
          'on',
          'suspicion',
          'of',
          'sabotage,',
          "that's",
          'in',
          'connection',
          'with',
          'strike',
          'action',
          'against',
          'President',
          'Slobodan',
          'Milosevic.',
          'You',
          'are',
          'listening',
          'to',
          'BBC',
          'news',
          'for',
          'The',
          'World']
```

use .stemming() to lowercast and stem your document

```
In [74]: temp1.stemming()
         #len(temp1.stemmed_list)
```

```
Out[74]: ['and',
          'yugoslav',
          'author',
          'are',
          'plan',
          'the',
          'arrest',
          'of',
          'eleven',
          'coal',
          'miner',
          'and',
          'two',
          'opposit',
          'politician',
          'on',
          'suspicion',
          'of',
          'sabotage,',
          'that',
          'in',
          'connect',
          'with',
          'strike',
          'action',
          'against',
          'presid',
          'slobodan',
          'milosev',
          'you',
          'are',
          'listen',
          'to',
          'bbc',
          'news',
          'for',
          'the',
          'world']
```

use .stopwarding() to stopward your document

```
In [75]: temp1.stopwarding()
```

```
Out[75]: ['yugoslav',
          'author',
          'plan',
          'arrest',
          'eleven',
          'coal',
          'miner',
          'two',
          'opposit',
          'politician',
          'suspicion',
          'sabotage,',
          'connect',
          'strike',
          'action',
          'presid',
          'slobodan',
          'milosev',
          'listen',
          'bbc',
          'news',
          'world']
```

you can save your own document named "result.txt" by .save_result()

```
In [76]: temp1.save_result()
```

```
Out[76]: 'file saved'
```

```
In [77]: len(temp1.stopwarded_list)
```

```
Out[77]: 22
```

```
In [ ]:
```