```
import requests


def shingles(doc, n):
    # create a dictionary of shingles for each doc
    doc_dictionary = {}
    for i in range(len(doc)):
        # set the doc id for each doc
        docID = 'doc' + str(i)

        # get all words
        words = doc[i]

        # define the set of shingles for each doc
        shinglewords = set()

        # loop through all char in words
        shingles = []
        for j in range(len(words) - n + 1):
            shingles = words[j: j + n]
            shingles = ''.join(shingles)

            if shingles not in shinglewords:
                shinglewords.add(shingles)

        doc_dictionary[docID] = shinglewords

    return doc_dictionary

def remove_stop_words(doc):
    stopwords_list = requests.get("https://gist.githubusercontent.com/rg089/35e00abf8941d72d4
    stopwords = set(stopwords_list.decode().splitlines())
    stopwords = list(stopwords)

    res = []
    for word in doc.split(" "):
        if word not in stopwords:
            res.append(word)

    return ' '.join(res)

def jaccard_dictance(s1, s2):
    intersect = len(list(s1.intersection(s2)))
    union = (len(s1) + len(s2)) - intersect
    res = intersect /  union

    return res
```

```
doc1 = 'Life is suffering'
doc2 = 'Suffering builds character'
doc3 = 'Character is the essence of life'


docs = {}

new_doc1 = remove_stop_words(doc1.lower())
new_doc2 = remove_stop_words(doc2.lower())
new_doc3 = remove_stop_words(doc3.lower())

docs[0] = new_doc1
docs[1] = new_doc2
docs[2] = new_doc3


new_docs = shingles(docs, 2)

print('distance bewteen doc1 and doc2: ', jaccard_dictance(new_docs['doc1'], new_docs['doc0']
print('distance bewteen doc1 and doc3: ', jaccard_dictance(new_docs['doc0'], new_docs['doc2']
print('distance bewteen doc3 and doc2: ', jaccard_dictance(new_docs['doc2'], new_docs['doc1']
```

```
distance bewteen doc1 and doc2:  0.2857142857142857
distance bewteen doc1 and doc3:  0.17857142857142858
distance bewteen doc3 and doc2:  0.25
```

✓ 0s completed at 9:19 PM