In [1]:
```
import spacy
import PyPDF2
import enchant
import en_core_web_sm
from spacy.matcher import PhraseMatcher
from scipy import spatial
import time
```

In [2]:
```
def getSpacyDocument(pdf_text, nlp):
    main_doc = nlp(pdf_text)  # create spacy document object

    return main_doc
```

In [3]:
```
def setCustomBoundaries(doc):
    # traversing through tokens in document object
    for token in doc[:-1]:
        if token.text == ';':
            doc[token.i + 1].is_sent_start = True
        if token.text == ".":
            doc[token.i + 1].is_sent_start = False
    return doc
```

In [4]:
```
def txt_file_reader(filename):
        f = open(filename, "r",encoding='utf8')
        text = ' '.join([i for i in f])
        return text
```

In [5]:
```
def createKeywordsVectors(keyword, nlp):
    doc = nlp(keyword)  # convert to document object

    return doc.vector


# method to find cosine similarity
def cosineSimilarity(vect1, vect2):
    # return cosine distance
    return 1 - spatial.distance.cosine(vect1, vect2)


# method to find similar words
def getSimilarWords(keyword, nlp):
    similarity_list = []
    d = enchant.Dict("en")
    keyword_vector = createKeywordsVectors(keyword, nlp)

    for tokens in nlp.vocab:
        if (tokens.has_vector):
            if (tokens.is_lower):
                if (tokens.is_alpha):
                    similarity_list.append((tokens, cosineSimilarity(keyword_vector, tokens.vector)))

    similarity_list = sorted(similarity_list, key=lambda item: -item[1])
    similarity_list = similarity_list[:30]

    top_similar_words = [item[0].text for item in similarity_list]

    top_similar_words = top_similar_words[:8]
    top_similar_words.append(keyword)

    for token in nlp(keyword):
        top_similar_words.insert(0, token.lemma_)

    for words in top_similar_words:
        if words.endswith("s"):
            top_similar_words.append(words[0:len(words)-1])

    top_similar_words = list(set(top_similar_words))

    top_similar_words = [words for words in top_similar_words if d.check(words) == True]
```

```
    return ", ".join(top_similar_words)


#keywords = ['label', 'package']
#similar_keywords = getSimilarWords(keywords, nlp)
```

In [6]:
```
def database():
    nlp=en_core_web_sm.load()
    txt=txt_file_reader('mazafaka.txt')
    doc_obj_txt=getSpacyDocument(txt,nlp)
    return doc_obj_txt
```

In [7]:
```
def search_for_keyword(keyword):
    nlp=en_core_web_sm.load()
    doc_obj=my_database
    phrase_matcher = PhraseMatcher(nlp.vocab)
    phrase_list = [nlp(keyword)]
    phrase_matcher.add("Text Extractor", None, *phrase_list)

    matched_items = phrase_matcher(doc_obj)

    matched_text = []
    for match_id, start, end in matched_items:
        text = nlp.vocab.strings[match_id]
        span = doc_obj[start: end]
        matched_text.append(span.sent.text)
    return matched_text
```

In [8]:
```
text=txt_file_reader('1profitring.com.txt')
```

In [9]:
```
import re
words = str(re.split(r'#\W+', text))
print(words[:100])
```

```
['mybouncesonly@gmail.com:realms\n webmaster@hbz.bz:realms5\n hooklist1@gmail.com:realms\n leslieewi
```

In [22]:
```
nlp=en_core_web_sm.load()
docs=[getSpacyDocument(text,nlp)]
```

In [11]:
```
from spacy.tokens import DocBin
doc_bin = DocBin(attrs=["ENT_IOB", "ENT_TYPE"])
```

In [15]:
```
doc_bin = DocBin(attrs=["LEMMA"])
docss = nlp(text)
doc_bin.add(docss)
```

In [23]:
```
docs
```

```
giliganzer@gmail.com:2504inamanina2006
betheking91@gmail.com:pondy)*
adeuk72@gmail.com:#yhw2p9ehv
power2earntw@gmail.com:ruffduck
mhugh50@gmail.com:1q2w3e
jasond188@gmail.com:boddy123
mudman817@gmail.com:ophelia817
jw9085387@gmail.com:1q2w3e
etrafficsurge@gmail.com:car123
viknik19ko66kn@gmail.com:vikokkk1966
hgordon@sbcglobal.net:#1pr626
rrhomes@hotmail.com:primo3
mvanzijl79@gmail.com:mo080879
notrafficneeded@gmail.com:red59
5angels4u@gmail.com:101617adzglorybe2716
jerrycoffee50@gmail.com:baracuda76
teamelitemom@gmail.com:ps1085ps1085
darsbro@gmail.com:rogue2012
imrers@gmail.com:ciuc2000#
misterkool50@gmail.com:l110558
```

In [24]:
```
#docs = [nlp("Hello world!")]
doc_bin = DocBin(docs=docs)
doc_bin.to_disk("spacydoc5.spacy")
```

In [25]:
```
doc_bin2 = DocBin().from_disk("spacydoc5.spacy")
```

In [26]:
```
doc_bin2
```

Out[26]:
```
<spacy.tokens._serialize.DocBin at 0x1d280fd23a0>
```

In [46]:
```
doc_list=list(doc_bin2.get_docs(nlp.vocab))
Doc.set_extension("my_custom_attr", default=None,force=True)
#print([doca._.my_custom_attr for doca in doc_list])
for doca in doc_list:
    doca._.my_custom_attr
print(doca)
```

```
vikhik19ko66kh@gmail.com:vikokkk1966
hgordon@sbcglobal.net:#1pr626
rrhomes@hotmail.com:primo3
mvanzijl79@gmail.com:mo080879
notrafficneeded@gmail.com:red59
5angels4u@gmail.com:101617adzglorybe2716
jerrycoffee50@gmail.com:baracuda76
teamelitemom@gmail.com:ps1085ps1085
darsbro@gmail.com:rogue2012
imrers@gmail.com:ciuc2000#
misterkool50@gmail.com:l110558
foxybird123@gmail.com:eastwest
mdwhttl@gmail.com:mdunky09
cruiserbrown@gmail.com:traffic22
reducethehype@gmail.com:dxh2ahgytml3z8rt
richard.moyer.1953@gmail.com:tinkerbell
httslcontact@gmail.com:zipperzoo1
malsoufi01@gmail.com:malsoufixyz
itsup2u@usa.com:highway1
w56496@aol.com:baby21$$
```

In [28]:
```
from spacy.tokens import Doc
from spacy.vocab import Vocab
docs_ = Doc(Vocab(doc_bin2))
```

In [47]:
```
doca
```

Out[47]:
```
mybouncesonly@gmail.com:realms
 webmaster@hbz.bz:realms5
 hooklist1@gmail.com:realms
 leslieewillats@gmail.com:tigger29
 giliganzer@gmail.com:2504inamanina2006
 betheking91@gmail.com:pondy)*
 adeuk72@gmail.com:#yhw2p9ehv
 power2earntw@gmail.com:ruffduck
 mhugh50@gmail.com:1q2w3e
 jasond188@gmail.com:boddy123
 mudman817@gmail.com:ophelia817
 jw9085387@gmail.com:1q2w3e
 etrafficsurge@gmail.com:car123
 viknik19ko66kn@gmail.com:vikokkk1966
 hgordon@sbcglobal.net:#1pr626
 rrhomes@hotmail.com:primo3
 mvanzijl79@gmail.com:mo080879
 notrafficneeded@gmail.com:red59
 5angels4u@gmail.com:101617adzglorybe2716
```

In [48]:
```
def search_for_keyword(keyword):
    nlp=en_core_web_sm.load()
    doc_obj=doca
    phrase_matcher = PhraseMatcher(nlp.vocab)
    phrase_list = [nlp(keyword)]
    phrase_matcher.add("Text Extractor", None, *phrase_list)

    matched_items = phrase_matcher(doc_obj)

    matched_text = []
    for match_id, start, end in matched_items:
        text = nlp.vocab.strings[match_id]
        span = doc_obj[start: end]
        matched_text.append(span.sent.text)
    return matched_text
```

In [49]:
```
search_for_keyword('mdwhttl@gmail.com:mdunky09')
```

Out[49]:
```
['mybouncesonly@gmail.com:realms\n webmaster@hbz.bz:realms5\n hooklist1@gmail.com:realms\n leslieewillats@gmai
l.com:tigger29\n giliganzer@gmail.com:2504inamanina2006\n betheking91@gmail.com:pondy)*\n adeuk72@gmail.com:#y
hw2p9ehv\n power2earntw@gmail.com:ruffduck\n mhugh50@gmail.com:1q2w3e\n jasond188@gmail.com:boddy123\n mudman8
17@gmail.com:ophelia817\n jw9085387@gmail.com:1q2w3e\n etrafficsurge@gmail.com:car123\n viknik19ko66kn@gmail.c
om:vikokkk1966\n hgordon@sbcglobal.net:#1pr626\n rrhomes@hotmail.com:primo3\n mvanzijl79@gmail.com:mo080879\n
notrafficneeded@gmail.com:red59\n 5angels4u@gmail.com:101617adzglorybe2716\n jerrycoffee50@gmail.com:baracuda7
6\n teamelitemom@gmail.com:ps1085ps1085\n darsbro@gmail.com:rogue2012\n imrers@gmail.com:ciuc2000#\n misterkoo
l50@gmail.com:l110558\n foxybird123@gmail.com:eastwest\n mdwhttl@gmail.com:mdunky09\n cruiserbrown@gmail.com:t
raffic22\n reducethehype@gmail.com:dxh2ahgytml3z8rt\n richard.moyer.1953@gmail.com:tinkerbell\n httslcontact@g
mail.com:zipperzoo1\n malsoufi01@gmail.com:malsoufixyz\n itsup2u@usa.com:highway1\n w56496@aol.com:baby21$$\n
ifalola@teprofits-chicago.com:alafia\n pinkie005@gmail.com:pinkie25\n myproductz4u@gmail.com:fairlea2920\n frn
cswhite@gmail.com:narf0044\n kanakjyoti4@gmail.com:']
```

In [ ]:
```

```