Spam filter documentation

This spam filter is using python and reading from emails I copied and pasted into txt files.

- The main running logic is in the following photo. It uses smaller methods within the SpamFilter class, this shows the protocol it follows to filter these emails

```
def presentationRun(self):
    for i in self.fileNames:
        f = open(f"emails/{i}", "r")
        txt = self.createEmailBody(f)
        finalBody = txt[0]
        entireBody = txt[1]
        if self.checkAgainstLinks(entireBody):
            print(f"{i} contains a blacklisted link, and is spam")
        else:
            if self.checkAgainstHashes(finalBody):
                print(f"{i} This email was found in the spam hash list")
                sleep(.25)
            else:
                if self.checkForUnsubscribe(entireBody):
                    print(f"{i} is not spam")
                    sleep(.25)
                else:
                    print(f"{i} has no unsubscribe link, adding to hashlist")
                    sleep(.25)
                    result = hashlib.md5(finalBody.encode())
                    self.hashes.append(result.hexdigest())
```

- The first thing that happens is the email is put through the createEmailBody method which just converts the text file into a string, but also creates a finalBody string that cuts out the first 3 lines of the email as well as, if applicable, the ending of an email, which would be if theres more than 2 lines of blank space. This allows for a more reliable hash since it wont include the persons name or potential different footers. This is returned along with the entireBody variable, which is the entire email in string format, which is used to check for blacklisted links and an unsubscribe link.

```
def createEmailBody(self, f):
    finalBody = ""
    entireBody = ''
    counter = 0
    emptyCounter = 0
    for i in f:
        entireBody += i
        if emptyCounter > 2:
            return finalBody
        if counter < 2:
            pass
        else:
            if i.strip():
                finalBody += i
                emptyCounter = 0
            else:
                emptyCounter += 1
        counter += 1
    return [finalBody, entireBody]
```

- After creating these 2 separate strings for the given file, it then checks for the presence of blacklisted links, that are specified beforehand. They are a mix of very specific links and general links, both work well with filtering.

```
                            The default ones are set as:
talent.com
https://www.google.com/url?q=https://delighted.com/e/en-x-insta-
cart/c/juInXzZ6hPRL34oM9tRwgmTI/0/00vXlNt7&amp;source=gmail&amp;ust=166889
3715674000&amp;usg=AOvVaw2OHXQcgLnWtNinkbpk8ssl
paypal.com
https://links.joinhoney.com/u/click?_t=70657193eb7a404887947be80fb10777
affirm.com
```

```
def checkAgainstLinks(self, txt):
    for i in self.links:
        if i in txt:
            return True
    return False
```

- If it contains a link, the current email is passed, and the email is marked as spam, not adding it to the hashlist, since the first thing checked is links
- If the email does not contain a specified link, it will then check the emails hash against the hashlist.

- If the email is in the hashlist, it is marked as spam.

```
def checkAgainstHashes(self, txt):
    result = hashlib.md5(txt.encode()).hexdigest()
    for i in self.hashes:
        if i == result:
            return True
    return False
```

- If the email is not in the hashlist, it then checks for an unsubscribe link

```
def checkForUnsubscribe(self, txt):
    substring = "Unsubscribe"
    final = str(txt)
    if final.find(substring) != -1 or final.find(substring.lower()) != -1 or final.find(substring.upper()) != -1:
        return True
    else:
        return False
```

- If it lacks an unsubscribe link, the finalBody version of the email is then hashed and the hash is stored into the hashes array.

```
class SpamFilter():
    def __init__(self):
        self.fileNames = ["ups.txt", "staples.txt", "glassdoor.txt",
                          "spamexample.txt", "job.txt", 'talent.txt', "apt.txt", "test.txt", "indeed.txt", "a
        self.hashes = []
        self.links = ["talent.com",
                      "https://www.google.com/url?q=https://delighted.com/e/en-x-insta-cart/c/juInXzZ6hPRL34o
```

Demonstrating it in action

- This is how it will be ran to demonstrate it, firstly the main logic from the top is ran once, filtering out and identifying spam, adding it to a hashlist if necessary, the exact same files will be ran through it again, but this time it would be faster since theres a hashlist that its being checked against first.

```
from SpamFilter import SpamFilter

if __name__ == "__main__":
    spam = SpamFilter()
    spam.presentationRun()
    print('---------------------------\n')
    spam.presentationRun()
    print(len(spam.hashes), "hashes in list")
    print(len(spam.links), "links in list")
```

- This is the result

```
ups.txt has no unsubscribe link, adding to hashlist
staples.txt is not spam
glassdoor.txt is not spam
spamexample.txt has no unsubscribe link, adding to hashlist
job.txt has no unsubscribe link, adding to hashlist
talent.txt contains a blacklisted link, and is spam
apt.txt has no unsubscribe link, adding to hashlist
test.txt has no unsubscribe link, adding to hashlist
indeed.txt has no unsubscribe link, adding to hashlist
affirm.txt contains a blacklisted link, and is spam
google.txt has no unsubscribe link, adding to hashlist
bestbuy.txt has no unsubscribe link, adding to hashlist
lucid.txt has no unsubscribe link, adding to hashlist
instacart.txt contains a blacklisted link, and is spam
intern.txt has no unsubscribe link, adding to hashlist
intern2.txt This email was found in the spam hash list
zillow.txt is not spam
ups2.txt This email was found in the spam hash list
paypal.txt contains a blacklisted link, and is spam
honey.txt contains a blacklisted link, and is spam
tabajo.txt has no unsubscribe link, adding to hashlist
--------------------------

ups.txt This email was found in the spam hash list
staples.txt is not spam
glassdoor.txt is not spam
spamexample.txt This email was found in the spam hash list
job.txt This email was found in the spam hash list
talent.txt contains a blacklisted link, and is spam
apt.txt This email was found in the spam hash list
test.txt This email was found in the spam hash list
indeed.txt This email was found in the spam hash list
affirm.txt contains a blacklisted link, and is spam
google.txt This email was found in the spam hash list
bestbuy.txt This email was found in the spam hash list
lucid.txt This email was found in the spam hash list
instacart.txt contains a blacklisted link, and is spam
intern.txt This email was found in the spam hash list
intern2.txt This email was found in the spam hash list
zillow.txt is not spam
ups2.txt This email was found in the spam hash list
paypal.txt contains a blacklisted link, and is spam
honey.txt contains a blacklisted link, and is spam
tabajo.txt This email was found in the spam hash list
```

- As you can see, the first run of this added 11 hashes to the list as well as blocking 5 files due to blacklisted links, and allowing the rest.
- Noteably, Intern1.txt and Intern2.txt had a different beginning and the second one was detected in the hashlist inside the first run of the method, even though they were addressed to different people

```
                              INTERN1.TXT
Hello,

You have been shortlisted for a student internship which will give you the
opportunity to earn
350.00 per week. Kindly Submit your full name in a text message to the
number below for more information
Tel: (360) 203-7255


Dr Michele Parker

                              INTERN2 .TXT
Hello Parker Gagliano,

You have been shortlisted for a student internship which will give you the
opportunity to earn
350.00 per week. Kindly Submit your full name in a text message to the
number below for more information
Tel: (360) 203-7255


Dr Michele Parker
```

Conclude + Summary

To conclude, this spam filter will first, check for blacklisted links, then check if the email is already in a hashlist, if these arent true, then it will check for an unsubscribe link, if that is true, its allowed through, if not it is added to the hashlist and marked as spam.