Отчёт по программам

Программа Надежды Васильевой

Получилось ли у вас загрузить программу, понятны ли были инструкции, пояснения автора? Получилось ли запустить программу, понадобилось ли как-то настраивать работу?

Программу запустила с первого раза. Ничего дополнительно настраивать не пришлось. Всё было хорошо пояснено в инструкциях.

Оценка: 10

Понятен ли текст программы, достаточно ли комментариев? Есть ли у вас замечания по тексту, что можно улучшить, на что стоит обратить внимание?

Всё в программе понятно. Был момент, в котором, возникало какое-то усложнение, то есть вместо простого программного решения, автор использовал сложную конструкцию.

Пример:

```
with open(string, 'r', encoding="utf
    file_content = f.readlines()
    text = ''.join(file_content)
    text = text.lower()  # чтобы заг
```

Можно было бы просто использовать функцию read(), как мне кажется, - это бы упростило код. С точки зрения оптимизации я столкнулась с одним неясным моментом. Например, одна величина была посчитана в цикле, когда подсчёт можно было бы вынести за его пределы.

```
    for letter in text:  # подсчет букв и символов в тексте
        if letter.isalpha():
            if letter not in letters_cnt:
                letters_cnt[letter] = 0
            letters_cnt[letter] += 1
            if letter not in symbols_cnt:
                symbols_cnt[letter] = 0
            symbols_cnt[letter] += 1
        else:
            if letter not in symbols_cnt:
                symbols_cnt[letter] = 0
            symbols_cnt[letter] += 1
        letters_all_len = len(text)
        symbols_all_len = len(text)
```

Оценка: 9

Работа программы

-понятен, удобен интерфейс. *Оценка: 10*

-насколько верно и полно программа решает задачу, заявленную автором. Как мне кажется, программа достаточно полно решает поставленную задачу. *Оценка: 10*

-результаты тестирования программы на собственных примерах, файлах, тестах. Попытайтесь выявить неточности, недоработки, баги и т.п. Попытайтесь понять их причины. Оцените насколько легко их исправить(требуется дописать пару строк или надо менять алгоритм) - здесь требуется достаточно длинный обстоятельный ответ.

В основной части программы я не нашла никаких ошибок. Я тестировала её на русском, французском, английском языках, вот что программа выдавала на тестах.

Вводимый текст: Екатерина

Результат программы: Ошибок нет

Вводимый текст: Ekaterina's telephone number is 89169660525

Результат программы: Язык определялся неправильно. Программа показала французский.

Вводимый текст: Mon chien est magnifique +=@#$%^&*()»"{}±±§~`

Результат программы: Всё очень хорошо обрабатывается. Правильно подсчитывается, даже символы перевода на новую строку и символы табуляция считаются корректно.

Вводимый текст: Picturesque landscape

Результат программы: Неправильно определяется язык текста. Программа показывает, что написано на французском.

Основная программа работает без ошибок, однако, в прикладной задаче, можно было бы, например, выбрать более интересный алгоритм, чтобы расширить диапазон текстов, для которых язык будет правильно определяться. Например, определение статистики триграмм, также можно было бы увеличить количество индикаторов-биграмм, для которых мы определяем статистику. Ещё одно банальное решение - добавить поиск букв, которые есть только в конкретном языке - таких много. Если мы найдём, например, é, можно точно сказать, что это французский, а если мы нашли я, то тоже можно сказать, что это русский язык. Такое введение следует сделать просто как надстройку, к уже существующему алгоритму.

Используемые средства, библиотеки - насколько оправдано применение именно таких средств, хорошо ли вам знакомы эти средства, узнали ли вы что-то новое про них.

Надежда использовала очень популярные библиотеки, их применение вполне оправдано. Из программы почерпнула для себя применение enumerate, если бы я писала программу, то скорее всего вводила бы переменную-счётчик, а здесь всё грамотно сделано без лишних переменных.

Какие задачи можно решать с помощью данной программы, какие языковые проблемы исследовать, как можно использовать данную программу (в каких более сложных продуктах). Включите фантазию!

Прикладные задачи**:**

1. Составление надёжного пароля. Мы могли бы собирать статистику самых популярных символов в паролях, а также статистику самых популярных триграмм и биграмм. Далее, если пользователь не хочет придумывать пароль самостоятельно, предлагать ему надёжный пароль, который не включает эти популярные биграммы и триграммы, а также символы, которые согласно статистике являются распространёнными.
2. Составление тэгов для фотографий. При публикации фотографии или рекламы, добавлять какие-то избыточные тэги, структура которых содержит популярные триграммы, биграммы и символы. Таким образом, наша фотография могла бы стать более популярной.
3. Кодирование информации. Подсчёт статистики символов N-грамм очень пригождается при декодировании текста. Так как алгоритмы для декодирования используют подобную статистику.
4. Система антиплагиата. Программу можно применить, при выявлении является ли работа человека калькой с чужого текста. Можно сравнивать статистики и выявлять закономерности.

Программа Георгия Кривова

Получилось ли у вас загрузить программу, понятны ли были инструкции, пояснения автора? Получилось ли запустить программу, понадобилось ли как-то настраивать работу? Запустить получилось с первого раза. Никаких дополнительных настроек я не проводила. *Оценка: 10*

Понятен ли текст программы, достаточно ли комментариев? Есть ли у вас замечания по тексту, что можно улучшить, на что стоит обратить внимание? У меня нету замечаний по тексту программы. Всё очень точно отражено. Комментарии наиболее полно описывают работу каждой функции. *Оценка: 10*

Работа программы

-понятен, удобен интерфейс. Всё очень удобно. Как вариант улучшения программы, можно, например, задать желательный диапазон значений для констант, потому что незнакомому с темой пользователю непонятно, насколько разумно вводить такие константы, как например, 1000 или 100000. Интуитивно понятно, что они должны быть маленькими, но я просто говорю это как предложение. Можно поставить максимальную оценку.

Оценка: 10

-насколько верно и полно программа решает задачу, заявленную автором. Как мне кажется, программа достаточно полно решает поставленную задачу. *Оценка: 10*

-результаты тестирования программы на собственных примерах, файлах, тестах. Попытайтесь выявить неточности, недоработки, баги и т.п. Попытайтесь понять их причины. Оцените насколько легко их исправить(требуется дописать пару строк или надо менять алгоритм) - здесь требуется достаточно длинный обстоятельный ответ.

Вот, что программа выдаёт на моих тестах:

Вводимый текст: Содержимое файла Email.txt. Я решила посмотреть, как работает программа с файлом, в котором встречаются адреса электронной почты.

В этом файле есть такие фрагменты:

john.smith@(comment)example.comи john.smith@example.com(comment)эквивалентны john.smith@example.com.

simple@example.com

very.common@example.com

disposable.style.email.with+symbol@example.com

other.email-with-hyphen@example.com

fully-qualified-domain@example.com

user.name+tag+sorting@example.com (может перейти в user.name@example.comпапку Входящие в зависимости от почтового сервера)

x@example.com (однобуквенная локальная часть)

example-indeed@strange-example.

Результат программы:

Программа Георгия отделила x в отдельное словоупотребление, в целом некорректно разделила текст на токены при наличии адресов электронной почты в нём. Я заметила, что почта делилась на две части - до @ и после. Хотелось бы, наверное, просто не учитывать адреса при подсчёте значений, это ведь не слова.

Вводимый текст:

katya#puchkova

katya1$puchkova

puchkova89169660534katya

екатерина….екатерина

Результат программы:

```
Ранг     Словоформа                Абс.частота    Отн.частота

1        puchkova                  2              0.40000
2        katya                     1              0.20000
3        katya1                    1              0.20000
4        puchkova89169660534katya 1              0.20000

== Сколько самых частотных лемм вывести? (введите число от 0 до 4): 4

Ранг     Лемма                     Абс.частота    Отн.частота

1        puchkova                  2              0.40000
2        katya                     1              0.20000
3        katya1                    1              0.20000
4        puchkova89169660534katya 1              0.20000
```

Как мне кажется, проще было бы удалять странные токены, которые не являются словами, ведь наша программа должна проверять закон для слов.

Также я подумала о том, чтобы сначала спросить для какого языка мы хотим определить правильность работы закона Ципфа, и выделять слова именно на этом языке, все остальные отвергать. *Оценка: 9*

Используемые средства, библиотеки - насколько оправдано применение именно таких средств, хорошо ли вам знакомы эти средства, узнали ли вы что-то новое про них.

Использование библиотек вполне оправдано, на семинарах нам рассказывали про разные способы лемматизации слов, Георгий делал это при помощи pymorphy. Я узнала про функцию isfile(). Также узнала про Counter из библиотеки collections.

Какие задачи можно решать с помощью данной программы, какие языковые проблемы исследовать, как можно использовать данную программу (в каких более сложных продуктах). Включите фантазию!

1. Родительский контроль. Если, например, администратор-родитель не хочет, чтобы его ребёнок пользовался компьютером, он может на этапе аутентификации подключить программу Георгия, которая попросит ввести какой-то текст, на более чем 100 символов. Потом определится коэффициент естественности текста, и если он не пересекает пороговое значение, можно блокировать доступ в систему.
2. Контроль правильности работы нашей итоговой модели, при использовании машинного обучения для обработки естественного языка. На семинарах нам показывали разные генераторы, которые строят тексты при помощи машинного обучения. Если мы хотим сделать такой генератор, в конце работы можно протестировать наш результат в программе Георгия и определить насколько естественным получился текст.

Оценка других программ, написанных по моей теме

Я оценивала программу Андрея Николаева.

Получилось ли у вас загрузить программу, понятны ли были инструкции, пояснения автора? Получилось ли запустить программу, понадобилось ли как-то настраивать работу? Запустить получилось с первого раза, никаких дополнительных настроек я не проводила. *Оценка: 10*

Понятен ли текст программы, достаточно ли комментариев? Есть ли у вас замечания по тексту, что можно улучшить, на что стоит обратить внимание? Текст программы был понятен. Единственно, возникали моменты, где код можно было бы упростить. Например, также как у Надежды, автор данной программы сначала делает список из строк, а потом соединяет их в одну строку, хотя можно было бы просто использовать метод read(). Но здесь я нашла только это, поэтому можно поставить максимальную оценку. *Оценка: 10*

Работа программы

-понятен, удобен интерфейс. Всё было понятно, единственно в терминале можно увидеть, что «полные результаты записаны в соответствующие файлы» - хотелось бы увидеть в каких конкретно файлах, так было бы проще. Ещё можно увидеть в списках имя, а рядом с именем цифру. Непонятно, что это за цифра. Я понимаю, что это абсолютная частота, но, возможно, надо было бы написать, что это именно она. Хотелось бы отметить, что Андрей обрабатывает текст из файла, название которого невозможно поменять, если не менять текст программы. Как мне кажется, это неудобно - если хочется протестировать программу на своих файлах, то придётся менять текст программы, а не вводить название файла из терминала.

Оценка: 8

-насколько верно и полно программа решает задачу, заявленную автором. Как мне кажется, программа достаточно полно решает поставленную задачу. Автор делит текст на токены, то есть программа выполняет графематический анализ, извлекает токены, выводит необходимую статистику. Ищет даты, имена, фамилии. *Оценка: 10*

-результаты тестирования программы на собственных примерах, файлах, тестах. Попытайтесь выявить неточности, недоработки, баги и т.п. Попытайтесь понять их причины. Оцените насколько легко их исправить(требуется дописать пару строк или надо менять алгоритм) - здесь требуется достаточно длинный обстоятельный ответ.

Тесты, которые я проводила, показали, что программа плохо справляется с двойными фамилиями. Я подумала, что можно было бы при встрече слова с дефисом делить его на две части - до дефиса и после - и обрабатывать эти части способом Андрея. Таким

образом, можно было бы добиться расширения множества фамилий, которые программа смогла бы обработать.

Вводимый текст:

Пучкова Екатерина

Андрей Карпов

Маргарита Левицкая-Филиппова

Татьяна Крылов-Пучкова

Рязанов-Андропов

Курчатов-Садовничий

Виктор Рене-Ферма

Рафаэль Липов-Берёзов

Семёнов-Французский

Петров-Васечкин

Петров-Иванов

Степанов-Сидоров

Результат программы:

Найденные фамилии:

Counter({'Карпов': 1})

~ Найденные организации:

Counter({'Рязанов-Андропов': 1, 'Курчатов-Садовничий': 1, 'Семёнов-Французский': 1, 'Петров-Васечкин': 1, 'Петров-Иванов': 1, 'Степанов-Сидоров': 1})

Как можно видеть двойные фамилии считаются организациями, но это вопросы не к Андрею, а к тому, кто делал библиотеку, с помощью которой Андрей обрабатывал текст.

Вводимый текст:

Риа Новости

RIA News

Kia Automobiles

Porsche

Ferrari

Apple

Iphone

МГУ им. Ломоносова

Результат программы:

~ Найденные организации:

Counter({'Риа Новости': 1, 'Apple': 1, "МГУ им. Ломоносова'": 1})

~ Найденные фамилии:

Counter({'Ломоносов': 1})

Как можно видеть, во многих случаях программа плохо обрабатывает текст. Но, поскольку я сама писала этот вид задания, я осознаю, что идеально извлечь токены конкретного вида очень трудно. Однозначного решения здесь нету. Поэтому, не смотря на ошибки, можно поставить девятку. *Оценка: 9*

Используемые средства, библиотеки - насколько оправдано применение именно таких средств, хорошо ли вам знакомы эти средства, узнали ли вы что-то новое про них.

Автор использовал популярные библиотеки, использовал их по назначению. Я узнала про библиотеку Navec, ближе познакомилась с pymorphy.

Какие задачи можно решать с помощью данной программы, какие языковые проблемы исследовать, как можно использовать данную программу (в каких более сложных продуктах). Включите фантазию!

1. Оценка труда. Есть особый вид работ, в которых люди, распределяясь по группам, должны проверять какие-то заявления или документы. Проверив документ, они могут вписать своё имя в список, а вместе с именем соответсвующий ID документа, который проверили. Далее можно использовать программу Андрея для определения частоты фамилии в этом списке. А потом в зависимости от частоты встречаемости фамилии в тексте выдать зарплату.
2. Рекрутинг талантливых студентов. Можно скачать списки олимпиад, которые проводятся для студентов. Используя программу Андрея, просмотреть: какие имена чаще всего встречаются. Если указаны даты рождения участников, также извлечь их из текста. Используя эту информацию, составить списки самых популярных студентов и начать процесс поиска контактов этих людей, чтобы пригласить на стажировку.
3. Формирование чёрного списка. В социальный сетях бывает такое, что некоторые люди пишут в комментариях неприличные вещи. Можно собрать фамилии этих людей, сформировать чёрный список, и с такими не работать.

Сравнение:

Мы с Андреем делали хоть и один вид задания, но как мне кажется, акценты были расставлены по разному. Андрей более подробно обрабатывал фамилии, мне стало интересно, как моя программа справляется с этой задачей.

Сравнение на данных:

Вводимый текст:

Пучкова Екатерина

Андрей Карпов

Маргарита Левицкая-Филиппова

Татьяна Крылов-Пучкова

Рязанов-Андропов

Курчатов-Садовничий

Виктор Рене-Ферма

Рафаэль Липов-Берёзов

Семёнов-Французский

Петров-Васечкин

Петров-Иванов

Степанов-Сидоров

Результат моей программы:

======Имена======

Екатерина Пучков:1

Татьяна Крылов:1

Рязанов:1

Виктор Рене:1

Организации программа не нашла совсем, что является корректным.

Вводимый текст:

Риа Новости

RIA News

Kia Automobiles

Porsche

Ferrari

Apple

Iphone

МГУ им. Ломоносова

Результат моей программы:

======Организации======

МГУ имя. Ломоносов:1

В данном примере моя программа намного хуже справилась с обработкой, программа Андрея смогла извлечь намного больше организаций.

Используемые инструменты:

- Для извлечения имён:
- Я использовала extract_fact(), Андрей использовал поле tag, которое он получил с помощью метода parse из библиотеки pymorpy.
- Для извлечения организаций

Я применяла метод tag_ner() для извлечения организаций и географических объектов. Андрей применял средства navec.

- Для извлечение дат

У нас были разные регулярные выражения для извлечения дат, к тому же он использовал средства библиотеки re, я использовала nltk.

Рассказ о показе своей программы «пользователям»

Алексей Шамшиев сказал, что у меня плохо написаны комментарии, я постараюсь впредь писать комментарии более подробно. Также мы, обсуждая мою программу, поняли, что я при подсчёте относительной частоты токена делю количество найденных токенов этого вида на общее количество токенов, которые я нашла своей функцией - это некорректно. Необходимо делить количество конкретных токенов на количество токенов всех видов, так как среди тех, на количество которых я делила нету полноценных имён. Этой ошибки у меня не было на первом графике, который показывал долю всех токенов вместе, а на других графиках эту ошибку я допустила.

Георгий Кривов указал, что двойные фамилии не обрабатываются моей программой «TaskApply.py», надо, конечно, подправить этот момент. Всё остальное его устроило.

Илья Поздняков не нашёл файла «Announcement.txt», потому что я не прописала, где конкретно его нужно искать. Из-за этого его программа свалилась, потому что он загрузил какой-то свой файл, а не файл заявления.

Андрей Арутюнов сказал, что у меня много дублирования кода, это замечание сделал также Алексей Шамшиев. Андрей посоветовал писать в формате PEP8, даже предложил использовать некоторые программы, которые, насколько поняла, сами мой код приведут к нормальной форме.

Мне было приятно слушать критику, потому что я поняла, что нужно исправить. Всё было сказано по делу. Оценку почти всем поставила максимальную, потому что, как мне кажется, стоит снижать, если человек, не понял каких-то очевидных вещей и стал ругать код из-за этого. Но мы друг друга понимали, поэтому таких проблем не возникло.

Алексей Шамшиев. *Оценка: 10*

Андрей Николаев. *Оценка: 10*

Георгий Кривов. *Оценка: 10*

Андрей Арутюнян. *Оценка: 10*

Андрей Спиваков. *Оценка: 9.* Замечание Андрея очень правильное, но я подумала отредактировать код по его примеру, который он прислал в отчёте, но фотография почему-то получилась урезанной, и я не смогла это сделать.