

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования «ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

09.03.04 - Программная инженерия

Профиль направления подготовки бакалавриата

«Системное и прикладное программное обеспечение»

ОТЧЁТ О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

Выполнил:

студент 2 курса группы 22207

Сергин Даниил Артемович

Место прохождения практики:

Кафедра информатики и математического обеспечения

Сроки прохождения практики:

29.05-09.06

Руководитель практики:

Богоявленская Ольга Юрьевна, к.т.н., доцент

Оценка _____

Дата _____

Петрозаводск — 2023

Содержание

Введение	3
1 Теория	4
1.1 Генераторы текста	4
1.2 Цепи Маркова	5
2 Практическая часть	6
2.1 Задача	6
2.2 Подзадача	6
3 Заключение	7

Введение

В современном мире все больше внимания уделяется нейросетям. Их способности становятся все более развитыми, и на данный момент ИИ уже умеет вести диалоги с пользователем, генерировать картины или помогать в написании программ. Одной из главных задач в этой области является генерация текстов. Самые большие нейросети (например, ChatGPT) обучаются на огромных объемах данных, что позволяет им генерировать ответы на запросы, которые выглядят очень правдоподобно.

Однако не всегда пользователю требуется генерировать что-то сложное, а нужно лишь нечто похожее на осмысленный текст. Для этих целей существуют более простые алгоритмы, такие как цепи Маркова. Задачей данной практики заключается в том, чтобы ознакомиться с этим алгоритмом и реализовать одну из его частей.

1 Теория

1.1 Генераторы текста

Генератор текста — компьютерная программа, генерирующая тексты (сленг генерёнку), корректные с точки зрения большинства языковых норм, но, как правило, лишённые смысла.

Генерация текста путём составления из полностью случайных слов даёт мусорный результат: бессмысленный для человека и легко распознаётся анализаторами текстов, и поэтому не применяется. Обычно применяется генерация по вручную написанным фразам-шаблонам.

В случаях, когда не важен смысл генерируемого текста, он составляется из «мешанки» предложений из разных исходных текстов, или из частей предложений. Такой метод совместно с шаблонами и синонимизацией используется в чат-ботах и ботах-комментаторах в соцсетях и блогах. Такие боты копируют собеседнику фразы, записанные с других чатов или сайтов. Более продвинутые чат-боты сортируют фразы по ключевым словам, поэтому их ответ более приближен к теме диалога.

1.2 Цепи Маркова

Цепь Маркова — последовательность случайных событий с конечным или счётным числом исходов, где вероятность наступления каждого события зависит только от состояния, достигнутого в предыдущем событии. Характеризуется тем свойством, что, говоря нестрого, при фиксированном настоящем будущее независимо от прошлого. Названа в честь А. А. Маркова (старшего), который впервые ввёл это понятие в работе 1906 года.

Одной из первых научных дисциплин, в которой цепи Маркова нашли практическое применение, стала лингвистика (в частности текстология). Сам Марков для иллюстрации своих результатов исследовал зависимость в чередовании гласных и согласных в первых главах «Евгения Онегина» и «Детских годов Багрова-внука». В настоящий момент цепи Маркова получили широкое распространение в сфере генерации текстов, так как позволяют создавать правдоподобные тексты, хотя их осмысленность сильно уступает тем текстам, которые генерируют нейросети.

Для генерации текста с помощью цепей Маркова берется исходный текст, который затем разбивается на префиксы и суффиксы. Префикс — сочетание нескольких слов, которому сопоставляются один или несколько суффиксов (то есть слов, идущих в оригинале после него). Каждой паре префикс-суффикс также приписывается частота, с которой эта пара встречается в источнике. Затем на основе полученных данных алгоритм уже генерирует текст, выбирая каждое следующее слово на основе предыдущих и словаря префиксов.

2 Практическая часть

2.1 Задача

Задачей нашей команды было написание кода для поиска префиксов и суффиксов в уже разбитом на части тексте. Программа должна принимать на вход массив со словами из исходного текста и на основе него создавать словарь префиксов и суффиксов. Также для каждой пары должна быть определена частота использования в оригинальном тексте.

Для реализации был выбран язык программирования Python. Результат работы алгоритма записывается в файл на языке JSON.

2.2 Подзадача

Основная задача по реализации модуля поиска префиксов была разделена на три подзадачи, в соответствии с количеством людей в группе: написать функции ввода/вывода, написать основную функцию поиска и провести тестирование программы.

Я занимался первой подзадачей - написанием функций ввода/вывода. Для успешной реализации понадобилось изучить встроенный в Python модуль JSON. Он позволяет приводить любые структуры данных к JSON-объекту — вплоть до пользовательских классов. А из него получать совместимую для работы в Python сущность — объект языка. Также для проверки существования файла использовалась библиотека OS.

Результатом работы стали две функции `GetPrefixesAndSuffixes` и `WritePrefixesAndSuffixes`. (см. Листинги 1 и 2 в Приложении)

Ввод выполняется путем передачи массива, как параметра функции поиска. Массив является результатом работы модуля разбора текста. Для вывода используется функция `WritePrefixesAndSuffixes`, записывающая в необходимый файл формата JSON результат работы. Если файл с таким именем уже существует, то функция допишет в него новые данные. За счет этого возможна обработка нескольких текстов.

`GetPrefixesAndSuffixes` является вспомогательной функцией, которая позволяет реализовать описанное выше. Она получает данные из указанного файла формата JSON и возвращает словарь, созданный на основе этих данных. Это необходимо, так как запись в файл новых префиксов и суффиксов происходит путем добавления их в старый словарь и перезаписи файл.

3 Заключение

В ходе данной практики была выполнена задача по ознакомлению с алгоритмами генерации текста, а также реализована часть алгоритма цепей Маркова, отвечающая за поиск префиксов и суффиксов в исходном тексте.

Список литературы

1. Керниган Брайан У., Пайк Роб Практика программирования. - М.: Вильямс, 2015. - 288 с.
2. Модуль json [Электронный ресурс]. URL: <https://pythonworld.ru/moduli/modul-json.html>

Приложения

Листинг 1 – Вспомогательная функция GetPrefixesAndSuffixes

```
def GetPrefixesAndSuffixes(filename):  
    """  
    :param filename: файл с префиксами формата JSON  
    :return: словарь префиксов и суффиксов вида: { prefix1: [suffix1, suffix2, ...]  
    :rtype: dict  
    """  
  
    # Проверка, что файл имеет расширение JSON  
    if filename.split('.')[-1] != "json":  
        raise Exception("Файл должен иметь расширение .JSON")  
  
    # Чтение и преобразование данных в словарь  
    with open(filename, "r") as file:  
        prefixesAndSuffixes = file.read()  
    return json.loads(prefixesAndSuffixes)
```

Листинг 2 – Функция вывода WritePrefixesAndSuffixes

```
def WritePrefixesAndSuffixes(filename, values):  
    """  
    :param filename: файл для записи или добавления префиксов  
    :param values: словарь префиксов и суффиксов вида: { prefix1: [suffix1, suffix2  
    """  
  
    prefixesAndSuffixes = {} # словарь префиксов и суффиксов  
    # Если файл уже существует - записываем данные из него в словарь  
    if os.path.exists(filename):  
        prefixesAndSuffixes = GetPrefixesAndSuffixes(filename)  
    prefixesAndSuffixes.update(values) # Добавление новых префиксов в словарь  
    # Запись в файл  
    with open(filename, "w") as file:  
        file.write(json.dumps(prefixesAndSuffixes, ensure_ascii=False))
```
