Оглавление

Задание 1

Разбить текст на фрагменты по N символов (шинглы). Составить словарь шинглов в CSV-формате.

Перед началом анализа текста выполняется его предобработка файла, в ходе которой из него исключаются часто встречающиеся неинформативные последовательности символов, перечисленные в словаре исключений (к таким могут относиться пробелы, союзы, знаки препинания, предлоги и т. д. Примеры « », «, », «. », « ,а », « ,но » « и » и т.д.). Результатом предобработки является рабочий файл.

На входе:

1. size — размер шингла;
2. inFile — имя исследуемого файла;
3. workFile — имя файла после предобработки (рабочий файл);
4. outFile — имя файла-отчета;
5. exclusion — имя файла словаря исключений. Каждая последовательность представлена отдельной строкой. Формат файла (один столбец):
   <"строка">

На выходе:

1. рабочий файл.
2. файл отчета. Содержит список шинглов в CSV-формате. Формат файла:

<порядковый_номер>;<"шингл">;<Ni>

Здесь:

шингл — последовательность символов;

Ni — количество шинглов номер i, найденных в файле.

Задание 2

Выполнить статистический анализ содержимого файла. Вычислить статистические величины для каждого байта и файла в целом. Результат записать в файлы отчета.

Статистические величины байта:

Ni — количество найденных байтов номер i в файле;

Pi = Ni/N — вероятность появления в тексте байта номер i;

Статистические величины файла:

N — общее количество байтов в файле (длина файла в байтах)

Nb — количество отличающихся значений байтов. Значение изменяется от 1 (когда файл заполнен одинаковыми символами) до 255 (когда встречаются все значения байтов)

H — энтропия. Вычисляется по формуле:

$H = -\sum_{i=1}^{K} pi * log(pi)$

K — количество различающихся байтов.

На входе:

1. inFile — имя файла;
2. byteStat — имя файла-отчета статистики байтов.
3. fileStat — имя файла-отчета статистики всего файла.

На выходе:

1. Файл-отчета статистики байтов. Отчет представляет собой таблицу из трех столбцов в формате CSV. Строки отсортированы по возрастанию значения байта):

<байт (0-255)> : <Ni> : <pi>

2. Файл-отчета статистики всего файла. Отчет состоит из трех строк:

N = <количество символов в файле>

K = <количество различных символов>

H = <вычисленное значение энтропии>

Задание 3

Выполнить рекурсивный поиск файлов в каталоге, значение энтропии которых больше заданного порогового значения.

H — энтропия. Вычисляется по формуле:

$H = -\sum_{i=1}^{K} pi * log(pi)$

Pi = Ni/N — вероятность появления в тексте байта номер i;

Ni — количество найденных байтов номер i в файле;

K — количество различающихся байтов.

На входе:

1. srcDir — каталог относительно которого выполнить рекурсивный поиск.
2. outFile — имя файла отчета.

На выходе:

1. Файл отчета в CSV-формате. Формат файла:

<порядковый_номер>;<"Полный путь к файлу">:<FileSize>:<MData>:<K>:<H>

Здесь:

FileSize — размер файла в байтах.

MData — время последнего изменения файла

Задание 4

Вычислить коэффициент Жаккара для двух файлов. Коэффициентом Жаккара двух множеств S и T называется величина $|S \cap T|/|S \cup T|$. Каждый файл делится на фрагменты по N символов (шинглы). Для полученных множеств шинглов вычисляется коэффициент Жаккара.

На входе:

1. size — размер шингла;
2. inFile1 — имя первого исследуемого файла;
3. inFile2 — имя второго исследуемого файла;
4. dictFile1 — имя файла словаря шинглов для первого файла;
5. dictFile2 — имя файла словаря шинглов для второго файла;
6. outFile — имя файла-отчета;

На выходе:

1. Словарь шинглов для первого файла в CSV-формате. Формат:

<порядковый_номер>;<"шингл">;<Ni>

Здесь:

шингл — последовательность символов;

Ni — количество шинглов номер i, найденных в файле.

2. Словарь шинглов для второго файла в CSV-формате (см. п.1).
3. Файл отчета. Формат файла:

File: <имя_первого_файла>
  Size: <размер_файла_в_байтах>
  MDate: <дата_время_последнего_изменения_файла>
  SCount: <общее_количество_шинглов>
  SUniqCount: <количество_уникальных_шинглов>
File: <имя_второго_файла>
  Size: <размер_файла_в_байтах>
  MDate: <дата_время_последнего_изменения_файла>
  SCount: <общее_количество_шинглов>
  SUniqCount: <количество_уникальных_шинглов>

Con=<значение: $|S \cup T|$ >

Dis=<значение: $S \cap T$ >

KJ = <Коэффициент Жаккара>

Ni — количество шинглов номер i, найденных в файле;

Задание 5

Сценарий контролирует список исходящих tcp-соединений. Относительно каждого выявленного соединения фиксируется время начала соединения (время выявления нового соединения), время окончания соединения (ранее выявленное соединение отсутствует в списке активных соединений), полный путь до исполняемого файла, инициирующего сетевое соединение и атрибуты соединения. Допускается использование временных файлов для хранения промежуточных результатов. Соединения, инициированные программой из файла исключений (exclusion) не учитываются.

На входе:

1. oufile — имя файла с результатом работы программы;
2. exclusion — имя файла, содержащего словарь исключений исполняемых файлов. Имя каждого файла начинается с новой строки. Формат файла:
   < "полный путь до исполняемого файла">
3. interval — интервал опроса сетевых интерфейсов, в секундах.

На выходе:

1. Заполненный файл outfile. Формат файла.

<start_time>:<stop_time>:<l_ip>:<l_port>:<f_ip>:<f_port>:<pid>:<progamm>:<arg>

Здесь:
start_time — время начала соединения
stop_time — время окончания соединения
l_ip — локальный IP-адрес
l_port — номер удаленного порта
f_ip — удаленный IP-адрес
f_port — номер удаленного порта
pid — идентификатор процесса
progamm — полный путь к исполняемому файлу
arg — аргументы, с которыми был запущен файл

Задание 6

Разработать интерактивный интерфейс для работы с менеджером пакетов apt. Предусмотреть следующие возможности:
- установка заданного пакета
- отображение списка установленных пакетов (по фрагменту имени)
- отображение списка пакетов доступных для установки (по фрагменту имени)
- построение полного списка зависимостей для заданного пакета
- удаление пакета

Задание 7

Разработать интерактивный интерфейс для работы с сетевыми соединениями. Предусмотреть следующие возможности:
- отображение списка портов, ожидающих подключения
- отображение списка исходящих соединений;
- отображение списка входящих соединений;
- отображение статистики по выбранному сетевому протоколу (ip, tcp, udp, icmp)
- отображение таблицы маршрутизации

Задание 8

Разработать интерактивный интерфейс для работы с таблицей маршрутизации. Предусмотреть следующие возможности:
- отобразить существующую таблицу маршрутов
- добавить новый маршрут
- удалить существующий маршрут
- изменить атрибуты существующего маршрута (target, netmask, metric, device и т.д.)

Задание 9

Разработать сценарий для извлечения из файла всех телефонных номеров в различных форматах. Предусмотреть случаи, когда в телефонном номере присутствуют круглые скобки, дефисы и пробелы.

На входе:

1. infile — имя исходного файла
2. outfile — имя файла-отчета

На выходе:

1. Заполненный файл-отчет в CSV-формате. Формат файла:

<порядковый_номер>;<"+7(NNN)NNN-NN-NN">;<Ni>

Здесь:

N — цифра от 0 до 9;

Ni — сколько раз встретился номер.

Задание 10

Разработать сценарий для извлечения из файла всех ip, mac адресов и доменных имен.

На входе:

1. inFile — имя исходного файла;
2. ipFile — имя файла-отчета для ip-адресов;
3. macFile — имя файла-отчета для mac-адресов;
4. domFile — имя файла-отчета для доменных имен.

На выходе:

1. файл-отчета для ip-адресов в CSV-формате. Формат файла:
<порядковый_номер>;<"NNN.NNN.NNN.NNN">;<Ni>
Здесь:
N — цифра от 0 до 9;
2. файл-отчета для mac-адресов в CSV-формате. Формат файла:
<порядковый_номер>;<"XX:XX:XX:XX:XX:XX">;<Ni>
Здесь:
X — цифра в шетнадцатеричном формате от 0 до F;
3. файл-отчета для доменных имен в CSV-формате. Формат файла:
<порядковый_номер>;<"A.B.C">;<Ni>
Здесь:
A.B.C — доменное имя;

Задание 11

Разработать сценарий для статистического анализа текстового файла.
Сценарий выполняет подсчет следующих величин:

- общее количество символов;
- количество строчных букв;
- количество прописных букв;
- количество знаков пунктуации из числа перечисленных в словаре;
- количество служебных частей речи (предлоги, союзы, частицы, междометия) из числа перечисленных в словаре;
- количество слов;
- количество предложений. Отличительный признак: предложение заканчивается заканчивается одним из следующих знаков:(".", "…","!","?"), следующее предложение начинается с прописной буквы;
- количество абзацев (перевод строки);

На входе:

1. infile — имя исходного файла.
2. dict1 — имя файла словаря знаков пунктуации. Формат файла (один столбец):

<"Знак_пунктуации">

Пример:

"."
","
"…"

3. dict2 — имя файла словаря служебных частей речи. Формат файла (один столбец):

<"Служебный_символ">

Пример:

"а"
"но"
"из"

4. outfile — имя файла-отчета.

На выходе:

Файл-отчет. Формат файла:
общее количество символов: <значение>
количество строчных букв: <значение>
количество прописных букв: <значение>
количество знаков пунктуации: <значение>
количество служебных частей речи: <значение>
количество слов: <значение>
количество предложений: <значение>
количество абзацев: <значение>

Задание 12

Разработать интерактивный интерфейс для команды find. Предусмотреть возможность поиска по следующим атрибутам:

- каталог относительно которого начинать рекурсивный поиск;
- маска файла;
- период времени последнего доступа к файлу;
- период времени последнего изменения содержимого файла;
- период времени последнего изменения прав доступа к файлу;
- диапазон размера файла;
- значение индексного дескриптора;
- тип файла;

Для найденных файлов предусмотреть возможность отображения следующих атрибутов (список определяется пользователем):

- время доступа к файлу;
- время изменения файла;
- время изменения прав доступа к файлу;
- размер файла;
- тип файла;
- номер индексного дескриптора;
- права доступа;
- имя владельца/группы файла.

Задание 13

Разработать сценарий для извлечения даты и времени подключения USB-устройств из файла /var/log/kern.log (ubuntu).

На входе:

1. infile — имя исходного log-файла;
2. dict — имя файла словаря подключенных когда-либо устройств;
3. usblog — имя файла журнала подключенных когда-либо устройств.

На выходе:

1. Файл словаря подключенных когда-либо устройств в CSV-формате. Формат файла:

<Порядковый_номер>:<Product>:<Manufacturer>:<SerialNumber>:<Mfr>:<idVendor>:<idProduct>:<bcdDevice>

2. Файл журнала подключенных когда-либо устройств в CSV-формате. Формат файла:

<dtConnect>;<tdDisconnect>;<SerialNumber>

Здесь:

dtConnect и tdDisconnect — время подключения и отключения устройства в формае «yyyy-mm-dd hh24:mi:ss».

Задание 14

Разработать сценарий для извлечения из access.log информации о количестве подключений с удаленных хостов. Записи расположить в порядке убывания количества подключений.

На входе:

1. infile — имя исходного log-файла;
2. outfile — имя файла с результатом работы программы.

На выходе:

1. Файл с результат11ом работы программы в CSV-формате. Формат файла:

<Порядковый_номер>:<IP>:< количество обращений>

Задание 15

Разработать сценарий для извлечения из access.log информации о статистике GET-запросов. Записи расположить в порядке убывания их частоты

На входе:

1. infile — имя исходного log-файла;
2. outfile — имя файла с результатом работы программы.

На выходе:

1. Файл с результатом работы программы в CSV-формате. Формат файла:

<Порядковый_номер>:<GET - запрос>:< количество обращений>

Задание 16

Разработать сценарий для XOR-преобразования файла. Предусмотреть случай, когда файлы разных размеров.

На входе:

1. infile — имя исходного файла для преобразования;
2. maskfile — файл используемый в качестве маски для XOR-преобразования;
3. outfile — имя преобразованного файла.

На выходе:

1. Преобразованный файл.

Задание 17

Разработать сценарий, который выполняет изменение времени модификации файлов в заданном каталоге случайным образом в заданном диапазоне дат и времени.

Сценарий принимает на вход пять параметров:

1. mind — минимальное значение даты;
2. maxd — максимальное значение даты;
3. mint — минимальное значение времени;
4. maxt — максимальное значение времени;
5. path — путь к файлу.

Задание 18

В каталоге расположены пронумерованные файлы в формате:
1имя_файла
2имя_файла
...
10имя_файла
…
Фрагмент «имя_файла» может состоять из символов, цифр (кроме первого знакоместа), специальных символов. Необходимо изменить наименование файлов следующим образом:
01имя_файла
02имя_файла
...
10имя_файла
…

Количество цифр в номере определяется параметром n. Например при n=4 имена файлов примут следующий вид:
0001имя_файла
0002имя_файла
...
0010имя_файла
…

На входе:
inpath — путь где расположены файлы;
n — количество цифр в номере;
outpath — имя преобразованного файла;
На выходе:
Файлы из inpath должны быть скопированы в outpath и переименованы.