Горохов Глеб, 7 класс
МБОУ «Гимназия №5», г. Рязань

Частотный словарь

Руководитель: Проказникова Е.Н.

Область исследования

Частотный словарь. Частота появления N-граммов в текстах.

Цели и задачи работы

Главная цель работы – создать практичную программу с удобным интерфейсом для работы с текстом, понятное для пользователя представление частотного словаря.

Методы исследования

Основной метод исследования - анализ. Вследствие анализа SEO, Big Data, большого количества информации в Интернете программа пришла к законченному варианту.

Главный результат

Программа проста в использовании и имеет понятный интерфейс. В будущем проект будет дорабатываться и доводится до идеала.

Выводы

Вовремя разработки проекта научился правильно и рационально использовать информацию. Научился эффективно искать информацию в Интернете.

Частотный словарь – это полезный проект, который нужно развивать. Многое в наше время не могло бы существовать без частотных словарей.

Введение

Развитие современного общества привело к появлению большого количества цифровой информации. Поиск нужного материала, соответствующего некоторым критериям, становится все более затруднительным, также, как и выделение сути написанного. Все это привело к появлению различных математических методов контент-анализа, так как, только математический подход позволяет создать объективные алгоритмы компьютерного анализа текста и дает возможность делать независимую оценку смысловой сущности.

Одним из определений контент-анализа является следующее: «Контент-анализ – это методика выявления частоты появления в тексте определенных интересующих исследователя характеристик, которая позволяет ему делать некоторые выводы относительно намерений создателя этого текста или возможных реакций адресата» [1, 2].

Исторически сложилось так, что на первых этапах развития этого направления в качестве наиболее объективной оценки текстов использовали частоту появления в нем различных характеристик. Выводы относительно интересующих вопросов производились с помощью экспертных оценок. Сейчас для получения объективных оценок содержания текста используются алгоритмы на основе искусственного интеллекта и машинного обучения.

Наиболее сложной задачей контент-анализа является обработка данных из социальных сетей. Это связано большим количеством данных (база данных социальной сети Facebook на сегодняшний день содержит более 1 миллиарда пользовательских аккаунтов и каждый день пользователи добавляют более 200 миллионов фотографий и оставляют более 2 миллиардов комментариев к различным объектам сети). В настоящее время даже алгоритмы для работы с BigData и системы ИИ не способны обрабатывать данные подобной размерности за приемлемое время. При этом специалисты из исследовательских центров широко используют данные социальных сетей для моделирования социальных и других процессов. Интернет компании применяют полученные в ходе таких исследований знания для разработки инновационных аналитических бизнес-приложений и сервисов. Одной из основных компонент такого программного обеспечения является процесс лексической декомпозиции текста, в результате которой во входном тексте распознаются токены и на выходе генерируется их список.

Токен можно определить, как последовательность буквенных и/или цифровых символов, отделенную слева и справа знаками форматирования текста и/или препинания. Разбивка текста на токены называется токенизацией, а программы, выполняющие токенизацию, – токенайзерами. В большинстве случаев токены совпадают со словами, поэтому термину токен соответствует термин слово в теоретической лингвистике. В инструкциях для пользователей

и в интерфейсах лингвистического ПО достаточно часто используется термин слово (word), а не токен (token) поскольку он более понятен и привычен. Однако, с точки зрения теоретической интерпретации, между двумя терминами имеются существенные различия [4].

Кроме задачи токенизации при работе с текстами, опубликованными в социальных сетях, существует еще такая большая прикладная задача, как автоматизация извлечения коллокаций из больших корпусов текстов при информационном поиске и решении задач криминалистики и компьютерной безопасности.

Термин «коллокация» широко распространен в корпусной лингвистике. В рамках этого раздела смысловое значение этого термина упрощается по сравнению с традиционной лингвистикой и является статистическим. Основными понятиями такого подхода становится частота совместной встречаемости, поэтому коллокации в корпусной лингвистике могут быть определены как статистически устойчивые словосочетания. При этом статистически устойчивое сочетание может быть, как фразеологизированным, так и свободным. За последние годы появилось большое число исследований и разработок, посвященных коллокациям, затрагивающих как теоретические аспекты статистического подхода к данному понятию, так и практические методы выявления коллокаций [5].

Появление больших корпусов текстов, размещенных в открытом доступе в социальных сетях привело к бурному развитию алгоритмов компьютерного анализа текста. Одним из наиболее простых методов является метод N-грамм.

Таким образом изучение методов компьютерного анализа текстовой информации является перспективным направлением для научных исследований, а разработка программного обеспечения и бизнес-приложений в данной области коммерчески востребовано.

Цели и задачи проекта

Цель работы: создание приложения подсчета частоты появления N-грамма в корпусе текста (частотный словарь).

Задачи проекта:

1. Создать удобный, интуитивно понятный интерфейс программы
2. Обеспечить возможность анализа произвольного корпуса текста (загрузка файла с текстом)
3. Обеспечить возможность введения произвольной коллокации (словосочетания) с последующим разбиением на N-граммы
4. Обеспечить возможность графического анализа результатов подсчета с помощью диаграмм

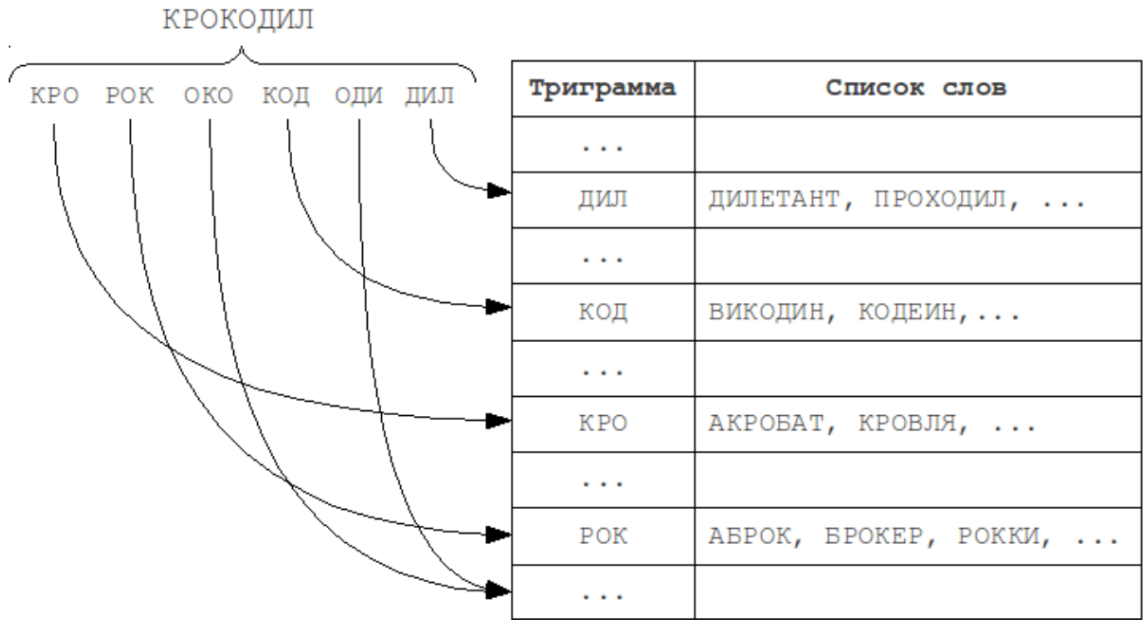
5. Обеспечить возможность сохранения результатов подсчета частот появления N-граммов в файле

Теоретическая часть

Метод N-грамм

Этот метод был придуман довольно давно, и является наиболее широко используемым, так как его реализация крайне проста, и он обеспечивает достаточно хорошую производительность. Алгоритм основывается на принципе:«Если слово А совпадает со словом Б с учетом нескольких ошибок, то с большой долей вероятности у них будет хотя бы одна общая подстрока длины N».Эти подстроки длины N и называются N-граммами.Во время индексации слово разбивается на такие N-граммы, а затем это слово попадает в списки для каждой из этих N-грамм. Во время поиска запрос также разбивается на N-граммы, и для каждой из них производится последовательный перебор списка слов, содержащих такую подстроку[6].

Наиболее часто используемыми на практике являются триграммы — подстроки длины 3. Выбор большего значения N ведет к ограничению на минимальную длину слова, при которой уже возможно обнаружение ошибок [6].

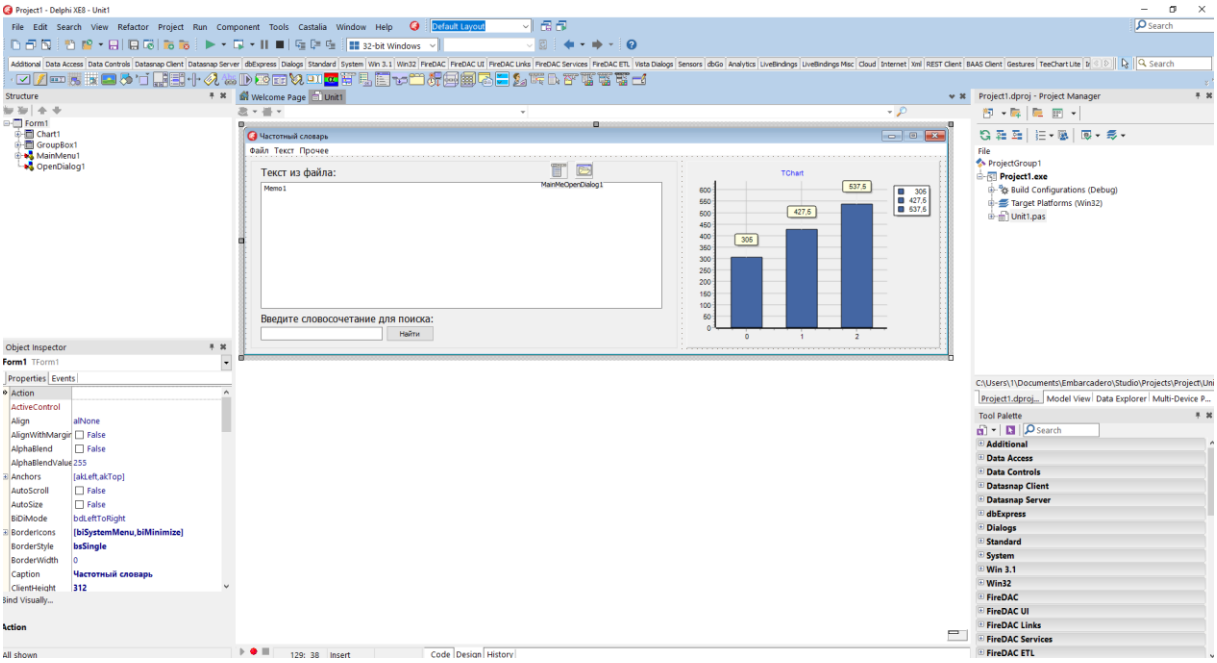


Следует отметить что, метод N-грамм оставляет полный простор для использования собственных метрик с произвольными свойствами и сложностью, но за это приходится пла-

тить — при его использовании остается необходимость в последовательном переборе около 15% словаря, что достаточно много для словарей большого объема[6].

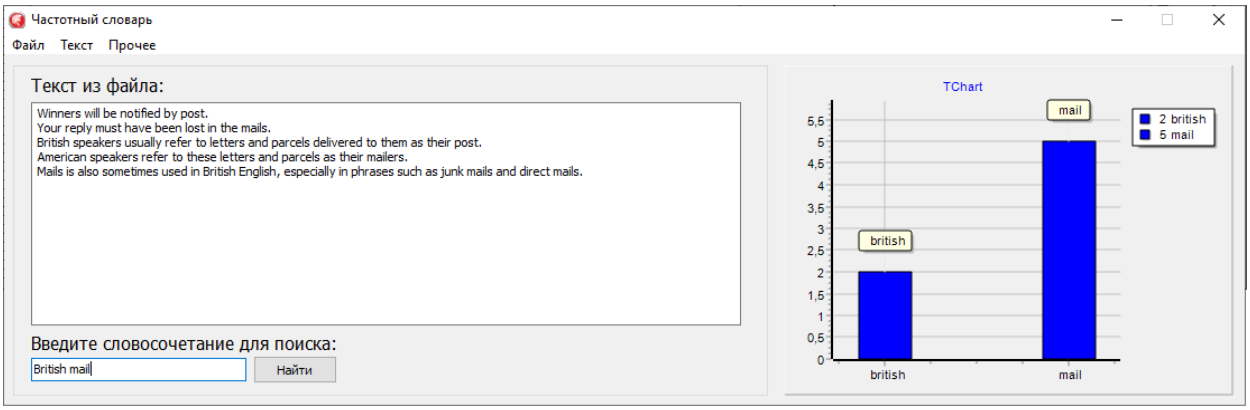
Описание программы

Запуск программы может быть осуществлен с помощью файла .exe или из рабочей области DelphiXEB.



Чтобы выбрать текст, с которым будет производится работа, нужно нажать кнопку [Файл] и выбрать в выпадающем меню [Открыть]. Нужно выбрать файл с расширением .txt, в который помещён текст.

Текст из выбранного пользователем файла будет отображён в окне «Текст из файла». Далее пользователю нужно ввести любое словосочетание (одно или более слов). Программа автоматически разобьёт введённое словосочетание на слова и будет работать с каждым по-отдельности. По завершении обработки статистика появления слов в тексте будет выведена в виде диаграммы. Для того чтобы сохранить результат работы программы, пользователю предоставляется возможность выбрать файл. Для этого нужно нажать кнопку [Текст], а затем из выпавшего меню нажать кнопку [Вывод]. Как и с выбором файла для ввода, мы также можем выбрать любой файл с расширением .txt для вывода.



После данной операции результат будет сохранён в файл в виде «Слово кол-во». Например:

mail 5

british 2

post 1

По завершении работы можно либо просто закрыть файл, нажав [Файл] >>>[Закрыть], либо закрыть программу, нажав [Прочее] >>> [Выход] .

Код программы приведен в ПРИЛОЖЕНИИ 1

Особенности разработки

Проект представляет собой клиентское приложение, разработанное для операционных систем Windows версии 7 и выше. Разработка приложения осуществлялась в DelphiXEB. В проекте алгоритмически возможно выделение N-грамма только в виде целого слова. В данной версии программного обеспечения отсутствует возможность учета словоформ и предполагаемых ошибок.

Пути дальнейшего развития проекта

Данный проект является не завершенным. Во-первых, планируется перевод проекта в на мобильную платформу Android с использованием языка Java в рамках выпускного проекта IT школы Samsung

Во-вторых, будет произведена доработка алгоритмов, позволяющая при подсчете частоты появления слова учитывать его словоформы и возможные ошибки.

В-третьих, добавить в приложение функционал, позволяющий на основе полученных данных построить вероятностную модель, которая затем может быть использована для оценки вероятности N-грамм в некоторомтестовом корпусе.

Выводы

Вовремя разработки проекта научился правильно и рационально использовать информацию. Научился эффективно искать информацию в Интернете.

Частотный словарь – это полезный проект, который нужно развивать. Многое в наше время не могло бы существовать без частотных словарей.

Список литературы и источников

1) Шалак Владимир «Элементы математических методов компьютерного контент-анализа текстов» https://ecm-journal.ru/docs/Ehlementy-matematicheskikh-metodov-kompjuternogo-kontent-analiza-tekstov.aspx

2) Федотова Л.Н. "Анализ содержания - социологический метод изучения средств массовой коммуникации". - М.: Институт социологии РАН, 2001. - 202 с.

3) Антон Коршунов, Иван Белобородов, Назар Бузун, Валерий Аванесов, Роман Пастухов,Кирилл Чихрадзе, Илья Козлов, Андрей Гомзин, Иван Андрианов, Андрей Сысоев,Степан Ипатов, Илья Филоненко, Кристина Чуприна,Денис Турдаков, Сергей Кузнецов «Анализ социальных сетей: методы и приложения» https://www.ispras.ru/proceedings/docs/2014/26/1/isp_26_2014_1_439.pdf

4) В.А. Яцко «ПРЕДМЕТНАЯ ОБЛАСТЬ КОМПЬЮТЕРНОЙ ЛИНГВИСТИКИ» http://lab314.brsu.by/kmp-lite/CL/predmetnaya-oblast-kompyuternoy-lingvistiki.pdf

5) Захаров В.П., Хохлова М.В. АНАЛИЗ ЭФФФЕКТИВНОСТИ СТАТИСТИЧСКИХ МЕТОДОВ ВЫЯВЛЕНИЯ КОЛЛОКАЦИЙ В ТЕКСТАХ НА РУССКОМ ЯЗЫКЕ

http://www.dialog-21.ru/digests/dialog2010/materials/html/22.htm

6) Никита Сметанин «Нечёткий поиск в тексте и словаре» https://habr.com/ru/post/114997/

ПРИЛОЖЕНИЕ 1 “КОД ПРОГРАММЫ»

```
unit Unit1;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Menus, Vcl.ExtCtrls,
VclTee.TeeGDIPlus, VCLTee.TeEngine, VCLTee.Series, VCLTee.TeeProcs,
VCLTee.Chart;
type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N8: TMenuItem;
    N10: TMenuItem;
    OpenDialog1: TOpenDialog;
    Memo1: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Edit1: TEdit;
    N6: TMenuItem;
    GroupBox1: TGroupBox;
    Chart1: TChart;
    Series1: TBarSeries;
  procedure N10Click(Sender: TObject);
  procedure N4Click(Sender: TObject);
  procedureFormCreate(Sender: TObject);
  procedure N5Click(Sender: TObject);
  procedure N8Click(Sender: TObject);
```

```
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
public
   F1: Text;
{ Public declarations }
end;
const MAX = 100;
var
  Form1: TForm1;
u, s, res: string;
ws: array[1..MAX] of string;
slov: array[1..MAX] of string;
kol: array[1..MAX] of integer;
w: string;
len: integer;
i, j, q, n, x: integer;
   F: TextFile;

implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
Reset(F);
while not Eof(F) do begin
readln(F, u);
len := length(u);
for i:=1 to MAX do kol[i]:=0;
i:=1;
whilei<= len do
if (lowercase(u[i]) >= 'a') and (lowercase(u[i]) <= 'z') then begin
w := lowercase(u[i]);
i := i + 1;
while (i<= len) and
          ((lowercase(u[i]) >= 'a') and
```

```
        (lowercase(u[i]) <= 'z')) do begin
w := w + lowercase(u[i]);
i := i + 1;
end;
j := 1;
while (j <= q) do
j := j + 1;
if j > q then begin
q := q + 1;
ws[q] := w;
end;
end
else
i := i + 1;
end;
s:=Edit1.Text;
len := length(s);
i:=1;
whilei<= len do
if (lowercase(s[i]) >= 'a') and (lowercase(s[i]) <= 'z') then begin
w := lowercase(s[i]);
i := i + 1;
while (i<= len) and
        ((lowercase(s[i]) >= 'a') and
        (lowercase(s[i]) <= 'z')) do begin
w := w + lowercase(s[i]);
i := i + 1;
end;
j := 1;
while (j <= n) do
j := j + 1;
if j > n then begin
n := n + 1;
slov[n] := w;
end;
```

```
end
else
i := i + 1;
for i:=1 to n do begin
for j:=1 to q do begin
ifPos(slov[i], ws[j], 1)<>0 then kol[i]:=kol[i]+1;
end;
end;
for i:=1 to n do begin
with Series1 do begin
Add(kol[i], slov[i], clBlue);
end;
end;
CloseFile(F);
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  Memo1.Lines.Clear;
end;
procedure TForm1.N10Click(Sender: TObject);
begin
  Form1.Close;
end;
procedure TForm1.N4Click(Sender: TObject);
begin
if OpenDialog1.Execute then begin
Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
AssignFile(F, OpenDialog1.FileName);
end;
end;
procedure TForm1.N5Click(Sender: TObject);
begin
  Memo1.Lines.Clear;
end;
procedure TForm1.N8Click(Sender: TObject);
```

```
begin
if OpenDialog1.Execute then begin
AssignFile(F1, OpenDialog1.FileName);
Rewrite(F1);
for i:=1 to n do begin
writeln(F1, slov[i], ' ', kol[i]);
end;
CloseFile(F1);
end;
end;
end.
```