# КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

**удк 001.103:002**

# АНАЛИЗ СОВСТРЕЧАЕМОСТИ СЛОВ

**М. А. Артемов, В. А. Сорокина, К. Е. Селезнев**

*Воронежский государственный университет*

**Поступила в редакцию 18.10.2012 г.**

**Аннотация.** В статье исследуется задача анализа совстречаемости рядом расположенных слов, описываются алгоритмы сбора, хранения и поиска данных, позволяющие выполнить данный анализ, а также производится оценка необходимого размера памяти для хранения собранной статистики.

**Ключевые слова:** алгоритмы анализа электронных текстовых документов, совстречаемость слов.

**Anotation.** In the paper the task of analysis occurrence neighbor words is researched. Beside algorithms of data gathering, storing and searching are described. The estimation of necessary volume of memory to store gathered data are made

**Keywords:** algorithms of analysis of electronic text documents, word occurrence.

## ВВЕДЕНИЕ

Во многих задачах анализа электронных текстовых документов необходимо находить слова, расположенные либо рядом, либо в пределах одного предложения или фразы. Примерами таких задач могут послужить построение байесовских классификаторов, поиск релевантных документов, поиск ключевых слов в тексте – учёт совстречаемости слов позволит значительно улучшить работу используемого алгоритма в целом.

Для нахождения часто совстречающихся слов необходимо определить, во-первых, какие именно слова встречаются рядом, и во-вторых, получить приемлемую оценку частоты их близкого расположения в тексте. Получение указанной статистической информации требует анализа большого количества документов. Это приводит к необходимости хранения и обработки большого количества данных, что может быть затруднительно без использования специализированных методов и алгоритмов.

В данной работе рассматривается задача сбора данных для анализа совстречаемости слов, приведена оценка необходимых объемов памяти, а также рассмотрены сами методы хранения и обработки информации.



## ОБЩАЯ ИДЕЯ МЕТОДА

В наиболее общем смысле рассматриваемую задачу можно формализовать следующим образом. На вход метода подаётся набор текстовых документов и набор искомых слов. Требуется определить и численно оценить, как часто заданные слова встречаются вместе.

Каждый текстовый документ рассматривается как последовательность слов. Анализ совстречаемости слов на уровне всего документа может не дать полезной статистической информации, поэтому необходимо собирать статистику на уровне абзаца, предложения или отдельной фразы. Здесь под фразой понимается последовательность стоящих рядом слов, длина которой может равняться или быть меньше длины всего документа.

При заданной длине фразы N каждый документ порождает определённое количество фраз в зависимости от соотношения N и общей длины этого документа L. При N = 1 каждое слово документа является отдельной фразой, а количество фраз равно длине всего документа. При N = L, напротив, фраза получается всего лишь одна. Задание L позволяет указать, насколько близко должны быть расположены искомые слова.

Каждая найденная фраза может рассматриваться как множество слов. Тогда требования исходной задачи можно формализовать следу-

ющим образом: необходимо вычислить количество фраз, которые содержат все поданные на вход слова. Однако, при такой формулировке есть небольшая проблема.

Пусть на вход подан текст DJQGSNFKTOY PDNVHFGWOEUTNVMSL, где каждая буква представляет слово, отделённое пробелами, и пусть задана длина фразы N=6. Тогда будут обнаружены следующие фразы.

Текст: DJQGSNFKTOYP
(0) DJQGSN
(1) JQGS**NF**
(2) QGS**NF**K
(3) GS**NF**KT
(4) S**NF**KTO
(5) **NF**KTOY
(6) FKTOYP

Получается, что слова N и F встречаются шесть раз, хотя на самом деле они встречаются рядом только однажды. Чтобы избежать избыточного вхождения, необходимо добавить дополнительное условие: либо одно из искомых слов должно быть завершающим, либо рассматриваемая фраза должна быть первой в исходном тексте. В первую фразу (0) текста, рассмотренного выше примера, оба искомых слова не входят, а вот во фразе (1) слово F является завершающим. Таким образом, данная пара будет учтена только один раз для данного текста.

## ОПИСАНИЕ МЕТОДА ХРАНЕНИЯ

Для решения задачи анализа совстречаемости слов, необходимо собрать обширную статистику различных фраз, которую удобно представлять в виде бинарной матрицы A размером W x F, где W – это количество уникальных слов из обработанного массива фраз, F – общее число обработанных фраз. Элемент матрицы равен 1, если слово встречается в данной фразе.

Таким образом, для рассмотренного выше примера получаем следующую матрицу табл.1.

Для обработки условия, что одно из слов должно быть завершающим, необходимо добавить еще одну матрицу (B), также размером W x F. Она будет хранить информацию о завершающем слове для данной фразы. Для рассмотренного выше примера матрица B будет иметь следующий вид табл. 2

Кроме того, для каждой фразы (т.е. для каждой строки матрицы A) необходимо хранить число C, показывающее сколько раз данное сочетание слов встретилось в исходном массиве документов.

## СТАТИСТИЧЕСКАЯ ОЦЕНКА ВХОДНОЙ ИНФОРМАЦИИ

В качестве источников фраз данных рассматриваются электронные письма из общедо-

Таблица 1

*Матрица (A), описывающая вхождения слов во фразы*

| | D | F | G | J | K | N | O | P | Q | S | T | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| фраза1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| фраза2 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| фраза3 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| фраза4 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| фраза5 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| фраза6 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

Таблица 2

*Матрица (B), представляющая завершающее слово фразы*

| | D | F | G | J | K | N | O | P | Q | S | T | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| фраза1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| фраза2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| фраза3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| фраза4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| фраза5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| фраза6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

ступных массивов писем TREC (http://plg.uwaterloo.ca/~gvcormac/treccorpus07/). Эти массивы сформированы из личной переписки многих добровольцев, давших разрешение на использование своей почты в исследовательских проектах. Использовалась выборка из 1000 писем.

Получена следующая статистика табл.3.

Если хранить данные в матрице размером 1000 (количество документов) на 37216 (количество уникальных слов), а на одну ячейку отводить один бит, то получается 1000 x 37216 x x 1 x 2 ~ 8 мегабайт для хранения матриц A и B, 1000 * 2 ~ 2 кб для хранения вектора C, 37216 * * 127 ~ 4мб – для хранения текстов слов, что сумме дает примерно 12 Мб.

Однако, несмотря на небольшой объем памяти требуемой для хранения данной информации, данная статистика не позволит учитывать взаимное расположение слов относительно другу друга в тексте. Как уже упоминалось, интересно получить не только информацию о вхождении конкретных слов в документ, но и о зависимости появления одного слова рядом с другим. Чтобы получить такие данные, необходимо собрать статистику для фраз с небольшой длиной например, 5,10,15. табл.4–6.

Для оценки требуемого объема памяти необходимо хранения собранных данных при длине фразы $N = 5$ (табл. 4), нужно выполнить следующие вычисления 434621 (количество уникальных фраз длиной 5) * 37216 (количество уникальных слов, используемых в исследуемых фразах)* 1(бит) * 2 ~ 3,7 гб плюс 434621 (количество уникальных фраз длиной 5) * 2(байт) ~ 0.8 mb (объем необходимых для хранения вектора C ), а также 37216 * 127 ~ 4мб – для хранения слов, то есть необходимо около 4 гигабайтов.

Аналогично для длины фразы $N = 10$ (табл. 5) получается, что для хранения данной информации также нужно около 4,1 гигобайт памяти. В таблице 6 представлена статистика для фраз длиной 15, для хранения которой необходимо около 4,2 гигабайтов памяти. Таким образом, можно заметить что с ростом длины фразы, увеличивается требуемый объем памяти.

## ОПТИМИЗАЦИЯ ХРАНЕНИЯ

Кроме того, необходимо заметить, что бит векторы, хранящие информацию, в каких фразах или письмах данное слово или фраза встретилось, являются сильно разряженными. Другими словами, возможно дополнительно сжать хранимую информацию для экономии памяти и ускорения ввода-вывода. Для сжатия будем

Таблица 3

*Статистика отдельных слов, собранная на массиве документов TREC*

| | |
|---|---|
| общее число обработанных документов | 1000 |
| максимальное число слов в письме | 13447 |
| среднее слов в письме | 742 |
| минимальное число слов | 0 |
| общее число уникальных слов во всем массиве документов | 37216 |
| максимальное число уникальных слов в письме | 2379 |
| среднее уникальных слов в письме | 236 |
| минимальное уникальных число слов в письме | 0 |
| самое встречаемое слово | 29096 раз в 932 письмах (такое слово одно) |
| средняя встречаемость слова | 742 раз в 236 письмах |
| минимальная встречаемость | 1 раз в 1 письме (таких слов 10115) |

Таблица 4

*Статистика для фраз длинной 5, собранной на массиве документов, состоящем из 1000 электронных документов*

| | |
|---|---|
| Общее число фраз длиной 5, собранных на всех документах | 688436 |
| уникальное число фраз длиной 5 | 434621 |
| Максимальное число фраз длиной 5 в одном документе | 800 |
| Минимальное число фраз, длиной не больше 5, в одном документе | 1 |

Таблица 5

*Статистика для фраз длинной 10, собранной на массиве документов, состоящем из 1000 электронных документов*

| | |
|---|---|
| Общее число фраз длиной 10, собранных на всех документах | 714478 |
| уникальное число фраз длиной 10 | 465866 |
| Максимальное число фраз длиной 10 в одном документе | 800 |
| Минимальное число фраз, длиной не больше 10, в одном документе | 1 |

Таблица 6

*Статистика для фраз длинной 15, собранной на массиве документов, состоящем из 1000 электронных документов*

| | |
|---|---|
| ТОбщее число фраз длиной 15, собранных на всех документах | 716662 |
| уникальное число фраз длиной 15 | 478107 |
| Максимальное число фраз длиной 15 в одном документе | 800 |
| Минимальное число фраз, длиной не больше 15, в одном документе | 1 |

использовать хорошо известный алгоритм "Кодирование длин серий", т.е. будем записывать бит вектор в следующем виде:

< длина последовательности 1 или 0,
значение >

Например, вектор 000000010000011100000 1 примит вид 801150135011.

Тогда поиск по хранящейся в таком виде матрице встречаемости слов, немного усложниться и будет выглядеть как

s = k Ø, i = Ø,

пока (s < m ui < h)

s + = ki

i++,

где n – индекс слова, ki – длина подпоследовательности 0 или 1, li – значение (0 или 1), m – длина фразы, в корой данное слово ищется.

Тогда l(i–1) есть искомое значение, если оно равно 1 , то данное слово входит в состав интересующей нас фразы, если 0 – то нет.

## АЛГОРИТМ ПОИСКА

Для оценки частоты совстречаемости заданных слов по собранному массиву данных необходимо выполнить следующие действия.

1) Определить номера столбцов, соотвествующих заданным словам;

2) Выполнить операцию ИЛИ для найденных столбцов матрицы B;

3) Выполнить операцию И для найденных столбцов матрицы A;

4) Выполнить операцию И для векторов, полученных на шаге 2 и 3;

5) Полученный на шаге 4 вектор даёт номера фраз, в которых встречаются поданные на вход слова. Необходимо просуммировать соответствующие элементы вектора C и, таким образом, будет получить оценку того, как часто поданные на вход слова встречаются в одной фразе.

## ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена задача анализа совстречаемости слов. Для решения данной задачи были собраны и обработаны статистические данные, предложен способ их хранения. Также, предложен алгоритм выполнения запросов для оценки частоты совстречаемости указанного набора слов. В продолжение данной работы необходимо оценить скорость работы алгоритма на больших объёмах текстовой информации.

СПИСОК ЛИТЕРАТУРЫ

1. *Шемакин Ю. И.* Начала компьютерной лингвистики. – М.: Издательство МГОУ, А/О "Росвузнаука", 1992.

2. *Гладкий А. В.* Формальные грамматики и языки. — М., 1973. — 368 с.

**3.** *Нариньяни А. С.* Лингвистические процессоры и представление знаний: Сб. науч. тр. — Новосибирск: ВС СО АН СССР, 1981. — 138 с.

4. *Бюлер К.* Теория языка. Репрезентативная функция языка. – М.: Изд. группа «Прогресс» , 2000. – 582 с.

5. *Лотман Ю. М.* К современному понятию текста // Статьи по семиотике искусства. – СПб.: Академический проект, 2002. – С. 79–83.

6. *Филиппов К. А.* Лингвистика текста: Курс лекций. – СПб.: Изд-во С.-Петерб. ун-та, 2003. – 336 с.

7. Daniel D. K. Sleator, David Temperly Parsing English with a Link Grammar – School of Computer Studies, Carnegie-Melon University, Pittsburg, PA, 1991.

**Артемов Михаил Анатольевич** – доктор физико-математических наук, профессор, заведующий кафедрой программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: artemov_m_a@mail.ru

**Сорокина Виктория Александровна** – аспирант кафедры программного обеспечения и администрирования информационных систем факультета прикладной математики и механики Воронежского государственного университета. Email: vica.sorokina@gmail.com

**Селезнев Константин Егорович** – кандидат технических наук, доцент программного обеспечения и администрирования информационных систем Воронежского государственного университета

**Artemov Mikhail A.** – Head of Department of Software and Information System Administering, Voronezh State University. Sciences, Professor. E-mail: artemov_v_a@mail.ru

**Sorokina Viktoriya A.** – Postgraduate student, the department of "The Software and information system administering", Voronezh State University. Email: vica.sorokina@gmail.com

**Seleznev Konstantin E.** – Associate Professor, the department of "The Software and information system administering", Voronezh State University