А.Н. Аверин

РАЗРАБОТКА СЕРВИСА ПОИСКА БИГРАММ

1. Вводные замечания

Биграммой будем называть два слова, которые в том или ином тексте (корпусе текстов) являются соседними. Таким образом, биграммами являются как двухсловные словосочетания, так и любые два слова, расположенные рядом в некотором тексте.

Простой набор биграмм из корпуса текстов никакой самостоятельной лингвистической ценности не имеет. Однако частотные биграммы находят свое применение:

- при снятии морфологической и лексической неоднозначности[1] (в том числе в машинном переводе для выбора перевода слова);
- при построении словарей сочетаний;
- при выявлении фразеологизмов;
- при оптимизации[2] веб-сайтов.

---

[1] *Сокирко А.В., Толдова С.Ю.* Сравнение двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель Маркова и синтаксический анализатор именных групп) // URL: **http://company.yandex.ru/grant/2005/01_Sokirko_92802.pdf**

[2] Имеется в виду оптимизация сайта с целью привлечения максимально возможного числа посетителей. Для этого большое значение имеет позиция страниц сайта в рейтинге поисковых систем по ключевым запросам. В конечном счете, результат во многом зависит от того, насколько близка относительная частота встречаемости ключевых фраз на страницах сайта к «идеальной» с точки зрения поисковиков. Опытным путем получено, что для многих поисковиков такая частота – около 0,5%, хотя она может варьироваться в определенных пределах. То есть при прочих равных условиях сайт с относительной частотой словосочетания *красная машина* равной 0,5% будет по данному запросу

2. Способы ранжирования биграмм

Для лингвистических исследований могут быть полезны только упорядоченные наборы биграмм. При этом существует довольно много численных параметров, по которым можно ранжировать массив биграмм:

1) частота первого слова;
2) частота второго слова;
3) различные меры устойчивости биграммы, которые учитывают частоты обоих слов:
   a) *Mutual Information (MI)*;
   b) *MI3*;
   c) *Log-log*;
   d) *Log-likelihood*;
   e) *T-score*;
   f) *Z-score*.

Описание и сравнение приведенных мер устойчивости не входит в задачу автора[1]. Тем не менее, необходимо сказать несколько слов о частоте взаимной информации (*Mutual Information*) как о наиболее «популярном» средстве измерения устойчивости биграммы.

Введем следующие обозначения:

MI – объем взаимной информации,

n_1 и n_2 – первое и второе слово биграммы,

находиться в рейтинге выше сайтов, на которых это словосочетание встречается реже или, наоборот, чаще (в последнем случае поисковая система считает, что фразы *красная машина* добавлены в текст страниц сайта специально для привлечения посетителей).

[1] Подробно о мерах устойчивости говорится в статье: *Хохлова М.В.* Автоматизированные методы вычисления устойчивости двухсловных сочетаний в тексте // Третья международная научная конференция «Прикладная лингвистика в науке и образовании», 16–17 марта 2006. СПб., 2006. С. 153–157.

$F(n_1, n_2)$ – частота встречаемости слов n_1 и n_2, идущих подряд в корпусе текстов,

$F(n_1)$, $F(n_2)$ – частоты отдельных слов n_1 и n_2 в корпусе,

N – число словоформ в корпусе.

Тогда частота взаимной информации для словоформ n_1 и n_2 в данном корпусе текстов будет вычисляться по следующей формуле:

$$MI = \log_2\left(\frac{F(n_1,n_2)\bullet N}{F(n_1)\bullet F(n_2)}\right)$$

Данная формула становится неинформативной, когда составляющие биграммы сравнительно редко встречаются в корпусе отдельно друг от друга. В этом случае $F(n_1, n_2) \approx F(n_1) \approx F(n_2)$ и MI будет тем больше, чем меньше частота биграммы. Максимальное значение меры MI достигается при частоте биграммы, равной 1.

Чтобы избежать такого поведения при малых частотах, можно использовать меру устойчивости $MI3$, которая отличается от MI тем, что частота биграммы возведена в куб:

$$MI3 = \log_2\left(\frac{F^3(n_1,n_2)\ \bullet N}{F(n_1)\bullet F(n_2)}\right)$$

Для решения приведенных в самом начале статьи лингвистических задач требуется строить и исследовать (автоматически или вручную) не все биграммы из текста/корпуса текстов, а только те из них, которые содержат интересуемое слово. Поэтому часто говорят не просто о биграммах, а о биграммах для конкретных слов.

3. Программные средства для поиска биграмм

Лингвистическая значимость набора биграмм тем выше, чем больше объем текстов, на основе которых они построены. В этой связи интерес представляют такие программные средства, которые позволяют получить результаты о степени устойчивости биграмм удаленно, без необходимости иметь на рабочем компьютере сам корпус текстов. В этом смысле самый удобный вариант – хранение на интернет-сервере как самого корпуса текстов, так и всех средств, необходимых для получения биграмм для слова.

У интернет-сервисов поиска биграмм есть следующие преимущества:

1) нет необходимости хранить локально сам корпус текстов;
2) не требуется установки каких бы то ни было специальных программ.

Тем не менее, такой вариант накладывает весьма жесткие временные требования на построение списка биграмм по запросу пользователя. Особенно это актуально при использовании корпусов большого объема (а только такие корпусы и подходят для обеспечения статистической достоверности результатов ранжирования биграмм).

Нами было произведено исследование сервисов поиска биграмм, доступных в сети Интернет и открытых для общего использования. Были протестированы, в частности, следующие сервисы поиска биграмм:

- DWDS (Das Digitale Wörterbuch der deutschen Sprache) – немецкий язык;
- Cobuild Concordance and Collocations Sampler – английский язык;
- Bigram Plus – английский язык;
- Yandex Direct – русский язык.

После такого исследования были сделаны следующие выводы:

1) сервисов поиска биграмм, имеющих свободный доступ через Интернет, достаточно мало – по 1–2 для рассмотренных языков;
2) для русского языка был найден лишь один такой сервис, причем круг задач, в которых его можно использовать, весьма ограничен.

На основе этих выводов было принято решение разработать сервис поиска биграмм со следующими характеристиками:

1) свободный доступ к сервису через Интернет;
2) корпус текстов большого объема;
3) тексты корпуса – русскоязычные;
4) быстрая выдача результатов на запрос пользователя;
5) наличие в сервисе морфологического компонента (поиск биграмм для лексем, а не для словоформ).

Сервис, удовлетворяющий описанным требованиям, был разработан в соавторстве с А.В. Сокирко (г. Москва) и доступен на сайте **www.aot.ru** (страница **www.aot.ru/demo/bigrams.html**).

4. Сервис поиска биграмм АОТ: описание

Адрес в сети Интернет: **http://www.aot.ru/demo/bigrams.html**
Объем корпуса текстов: 448 млн словоформ.
Язык корпуса: русский.
Время построения страницы результатов запроса: ≈8 сек.[1]
Задаваемые параметры поиска:

- одно из двух слов биграммы (ключевое слово);
- расположение ключевого слова на первой/второй позиции в биграмме;
- пороговое значение частоты биграммы (по умолчанию 2).

[1] Незначительную погрешность может давать скорость соединения с сетью Интернет.

Доступные варианты ранжирования результатов:

- частота биграммы;
- частота коллоката (т.е. второго слова биграммы);
- мера *Mutual Information*;

В качестве корпуса текстов использована текстовая база библиотеки Максима Мошкова.

Пример первых пяти строк таблицы для ключевого слова *стол* приведен в табл. 1:

Таблица 1. Результаты поиска по запросу *стол*

Word1	Word2	WordFreq1	WordFreq2	BigramsFreq	MI	Контекст
СТОЛ	УСТАВИТЬ	203089	3183	330	7.839585	контекст
СТОЛ	ЗАВАЛИТЬ	203089	7937	561	7.286914	контекст
СТОЛ	ПРЕЗИДИУМ	203089	2840	199	7.274384	контекст
СТОЛ	ЛОМИТЬСЯ	203089	2123	131	7.090969	контекст
СТОЛ	ВУЛФА	203089	2992	175	7.013751	контекст

Поиск ключевого слова в корпусе производится во всех его грамматических формах. Для каждой биграммы есть возможность просмотра контекстов из корпуса. При этом выводится

название произведения, автор и фрагмент текста, включающий биграмму:

Анна Арнольдовна Антоновская. 3. Время освежающего дождя
Длинный ***стол уставлен*** красивыми кувшинами, полными благоухающих роз.

Степан Павлович Злобин. 2. Степан Разин
Стол уставили - аж протопопица ахнула.

Илья Ильф и Евгений Петров. Рассказы
За столом сидят чистенькие старички, у всех под бородами салфеточки, ***стол уставлен*** разной едой, никаких нет порций, бери что хочешь, понимаете, хватай что хочешь.

Гюстав Флобер. Саламбо
Вслед за тем ***столы уставили*** мясными блюдами.

…

5. Сервис поиска биграмм АОТ: описание алгоритма

В этой части статьи приводится описание алгоритма, лежащего в основе созданного сервиса поиска биграмм.

При проектировании сервиса мы исходили из того, что корпус текстов является статическим. Было два основных требования:

1) поиск должен вестись минимум времени;
2) все компоненты сервиса должны занимать в оперативной памяти сервера не больше 100 Мб.

Разработку сервиса можно разделить на три части:

1) предобработка исходных текстов корпуса (производится один раз): на сервер выкладываются результаты предобработки, которые в дальнейшем не подвергаются изменениям;

2) обеспечение работы сервиса в реальном времени: поиск биграмм для заданного пользователем слова в корпусе, используя файлы предобработки;
3) пользовательский интерфейс.

5.1. Предобработка текстов корпуса

Корпус текстов сервиса представляет собой все художественные произведения библиотеки Максима Мошкова на 2003 год.

Предобработка состояла из следующих этапов (все этапы автоматизированы):

1) Объединение всех текстов произведений библиотеки в один файл **corpusbase.txt**[1].
2) Построение из файла **corpusbase.txt** инвертированного списка лексем и сохранение его в файл **baseindex.txt**. Таким образом в файл **baseindex.txt** в алфавитном порядке записываются все лексемы из **corpusbase.txt** со списком позиций в файле **corpusbase.txt**, откуда начинаются вхождения данного слова. Файл **baseindex.txt** используется при оптимизации построения страницы с контекстами для биграммы.
3) Построение файла биграмм **bigrams.txt**. Для этого файл **corpusbase.txt** считывается по предложениям. Слова предложения записываются в файл **bigrams.txt** по два в строку (т.е. из N-словного предложения создается $N - 1$ биграмм). При этом также производится лемматизация словоформ. Приведем фрагмент получаемого на данном этапе файла **bigrams.txt**:

[1] Имена файлов и другие имена собственные, упоминаемые ниже, приводятся исключительно для удобства описания и не соотносятся с реальными объектами.

он неизменно неизменно подкрепляться подкрепляться бокал бокал отличный отличный хоккеймер

4) Файл **bigrams.txt** сортируется по алфавиту. Каждой биграмме приписывается ее частота (по умолчанию 1). Если при сортировке обнаруживаются множественные вхождения одной биграммы, то проделываются следующие операции:
   - каждое вхождение биграммы кроме первого удаляется;
   - у первого вхождения инкрементируется частота.

В этом же файле каждому слову биграммы приписывается его частота в исходном корпусе текстов.

5) Считается частота взаимной информации (*Mutual Information*) для каждой биграммы. Полученное значение также записывается в строку с биграммой.

В результате строки файла **bigrams.txt** принимают следующий вид (на примере биграммы *он неизменно*):

он неизменно 32 203089 27704 2,521345

Здесь

- 32 – число вхождений биграммы в корпус;
- 203089 27704 – частоты слов *он* и *неизменно* в исходном корпусе;
- 2,521345 – мера *Mutual Information*.

6) Теперь проще всего было бы считать на сервере информацию из этого файла в некоторую структуру данных и далее по запросу пользователя выдавать биграммы и дополнительную информацию о ней из этой структуры. Но размер файла **bigrams.txt** для используемого корпуса текстов превышает 500 Мб, а на используемом сервере свободной оперативной памяти – 100 Мб. Кроме того, даже если бы свободной памяти было бы больше 500 Мб – такая эксплуатация ОЗУ неоправданна. Было принято другое решение.

Для файла **bigrams.txt** был создан, как и для исходного файла **corpusbase.txt,** инвертированный индекс **bigrindex.txt**, в который были записаны все леммы из **bigrams.txt** (в алфавитном порядке) со списком позиций в файле **bigrams.txt**, откуда начинаются вхождения данной леммы. Размер такого файла был менее 100 Мб, поэтому информацию из него можно было хранить в оперативной памяти сервера.

5.2. Обеспечение работы сервиса в реальном времени

На жесткий диск сервера были положены файлы **corpusbase.txt, baseindex.txt, bigrams.txt** и **bigrindex.txt**. Содержимое файла **bigrindex.txt** каждый раз при запуске сервиса загружается в ОЗУ в массив структур **STerm**, каждая из которых содержит:

1) слово, входящее в одну из биграмм корпуса;
2) массив смещений вхождений биграмм с этим словом в файле **bigrams.txt**.

Предположим, пользователь запрашивает биграммы для некоторого ключевого слова **W**. В этом случае система работает следующим образом:

1) По массиву структур **STerm**, находящемуся в оперативной памяти сервера, осуществляется бинарный поиск слова **W**;

2) При отсутствии такого слова пользователю предлагается ввести другое слово; если же слово нашлось, считывается массив смещений для него;
3) В соответствии с полученными смещениями считываются нужные строки файла **bigrams.txt**; для каждой строки производятся следующие действия:
   a) Считывание биграммы, ее частоты, частот обоих слов в корпусе и значения *MI*;
   b) Производится проверка того, что частота биграммы больше или равна заданному пользователем пороговому значению (по умолчанию 2). Также проверяется, что позиция ключевого слова в найденной биграмме соответствует заданному пользователем значению (первое либо второе слово биграммы). Если биграмма не удовлетворяет хотя бы одному из этих условий, то она пропускается. Иначе – считанная для биграммы информация записывается в массив строк, предназначенный для выдачи пользователю.
4) Выдача пользователю результатов. При нажатии на кнопку «Контекст» для требуемой биграммы осуществляется бинарный поиск ее составляющих по файлу **baseindex.txt** (представляющему собой инвертированный индекс для файла **corpusbase.txt**). Оттуда считываются смещения в корпусе для предложений с этой биграммой. Затем производится непосредственно считывание предложений из файла **corpusbase.txt** и вывод их в итоговую таблицу в документе html.

В разделах 5.1 и 5.2. нашей статьи были рассмотрены два аспекта разработки сервиса поиска биграмм, расположенного на сайте **www.aot.ru**:

- предобработка текстов корпуса;
- обеспечение работы сервиса в реальном времени.

Третий аспект (пользовательский интерфейс) выходит за рамки лингвистики, поэтому в данной работе он не описывается.