Тематическое упорядочение текстов при формировании сводных документов

© Васильев Виталий Геннадьевич

ООО «ЛАН-ПРОЕКТ»
vvg_2000@mail.ru

Аннотация

В работе рассматривается новый подход к автоматизации процессов подготовки сводных документов, основанный на тематическом упорядочении текстов. Проводится описание и сравнительный анализ различных методов решения данной задачи на различных тестовых массивах.

Введение

В настоящее время для автоматизации обработки потоков текстовых данных часто используются средства автоматической классификации текстов. Например, с их помощью осуществляется подготовка различных сводных и обобщенных справок путем отбора сообщений по интересующим тематикам. При формировании такого типа документов приходится сталкиваться с наличием повторяющейся информации и неупорядоченностью расположения тематически близких текстов в списках результатов. Возможная технология автоматизированного формирования сводных документов учетом решения указанных задач приведена на рис. 1.

Необходимо отметить, что так как в рамках приведенной технологии упорядочивание документов производится в рамках отдельной рубрики, то общее количество обрабатываемых документов оказывается относительно небольшим (порядка нескольких сотен документов). Данное свойство позволяет рассматривать более широкий спектр методов для решения задачи упорядочения документов

Реализация приведенной технологии также требует решения ряда дополнительных задач (сбор, классификация, выявление дубликатов документов), которые не рассматриваются в настоящей работе. С описанием применяемых подходов для решения ряда из них можно ознакомиться в дополнительной литературе. В частности, описание используемых методов классификации и выделения значимых фрагментов в текстах приводится в [17, 16].

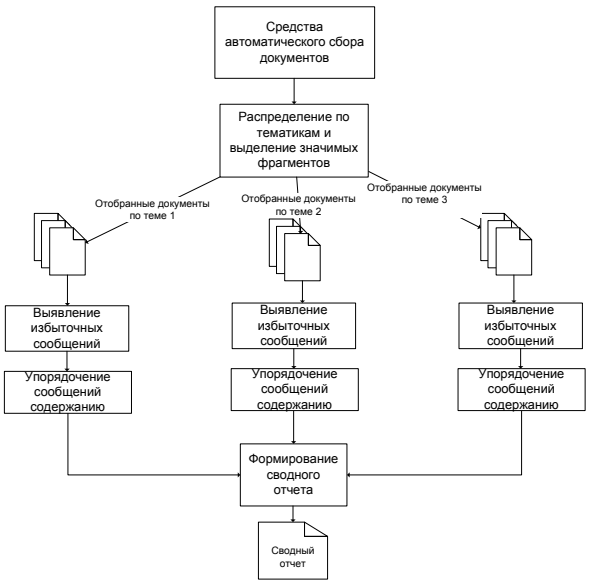

Рис. 1. Технология формирования сводных документов

Работа имеет следующую структуру. В первом разделе дается краткое описание моделей представления и мер близости текстов. Во втором разделе рассматриваются различные формальные постановки задачи тематического упорядочения документов и методы их решения. В третьем разделе приводятся результаты экспериментального исследования методов на различных тестовых массивах. В заключении приводятся общие выводы по результатам работы и описываются перспективные направления дальнейших исследований.

1 Модели представления и меры близости текстов

Существенным моментом при построении алгоритмов тематического упорядочения и выявления повторяющейся информации в текстах является выбор модели представления и меры близости текстов. При решении указанных задач различными авторами применяются как стандартные теоретико-множественные модели, используемые при решении задач поиска и классификации, так и специфические модели,

Труды 11[й] Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2009, Петрозаводск, Россия, 2009.

основанные на построении цифровых отпечатков, которые активно используются при выявлении дубликатов и копий документов.

В стандартных теоретико-множественных моделях в качестве информационных признаков обычно используются отдельные слова и словосочетания, а в качестве весов слов используются различные производные функции от следующих характеристик:

- частоты встречаемости слова в тексте,
- числа документов, содержащих слово,
- положения слова в тексте,
- присутствие слова в специальных словарях.

Методы выделения информационных признаков и вычисления их весов достаточно подробно описаны в литературе [17]. При этом наиболее распространенной схемой вычисления весов является TF-IDF, при использовании которой вес слова t_j, $j=1,...,m$, в тексте d_i, $i=1,...,n$ определяется как $w_{ij}=f_{ij}\log\left(\frac{n}{n_j}\right)$, где n - общее число текстов в массиве, f_{ij} - частота встречаемости слова t_j в тексте d_i.

В моделях на основе цифровых отпечатков в качестве информационных признаков используются хэш-коды различных элементов текстов. В последнем случае удается получать более компактное представление текстов, а вычисление близости между текстами производить путем простого сравнения хэш-кодов. Рассмотрим более подробно методы, применяемые для построения цифровых отпечатков. Данные методы можно условно разбить на следующие виды:

- синтаксические - текст описывается с помощью набора хэш-кодов для цепочек последовательно идущих слов [1, 2, 6],
- лексические - текст описывается с помощью хэш-кодов для наборов слов, которые входят в некоторое множество [8, 10].

Одной из первых работ по синтаксическому подходу является [2]. В ней предлагается представлять текст в виде множества хэш-кодов всех последовательностей соседних слов длины k, которые названы шинглами. Поскольку число шинглов обычно достаточно большое (примерно равно числу слов в документе), то для сокращения их числа авторами работы предлагается несколько эвристических подходов:

- отбираются шинглы равные 0 по некоторому модулю;
- отбирается шинглов с минимальными значениями хэш-кодов;
- отобранные шинглы объединяются в группы - мегашинглы.

В ряде работ предлагается использовать также и другие подходы к выбору последовательностей в текстах. В работах [1, 18, 12] в качестве элементов для вычисления хэш-кодов рассматриваются последовательности слов входящие в отдельные предложения, в k соседних предложений с перекрытием и без, в блоки предложений заканчивающиеся предложениями с хэш-кодом равным нулю по некоторому модулю, а также в текст целиком. В частности, в работе [13] текст представляется в виде последовательности хэш-кодов k-грамм символов. Для ее построения предлагается лексический алгоритм Winnowing, в котором по последовательности хэш-кодов $h_1,...,h_n$ всех k-грамм перемещается окно размера w, где n - число k-грамм выделенных в тексте. В каждом окне $h_i,...,h_{i+w-1}$, $i=1,...,n-w+1$, отбирается один элемент с минимальным значением.

В работах [5, 8, 10] используется лексический подход к построению цифровых отпечатков. Он основан на построении функции, которая осуществляет отображение вектора весов признаков x размерности m в битовый вектор $z=(z_1,...,z_f)$ (f обычно 32 или 64) следующим образом:

$$z_i=\begin{cases}0, & y_i<0,\\ 1, & y_i\geq 0,\end{cases} \qquad (1)$$

где $y_i=\sum_{j=1}^{m}x_j(-1)^{s_{ji}}$, $s_j=(s_{j1},...,s_{jf})$ - битовый вектор длины f для информационного признака с номером $j=1,...,m$, полученный с помощью стандартной хэш-функции (MD5 или SHA-1). Особенностью получаемого вектора z является то, что близкие вектора признаков получают близкие значения данного вектора.

В работах [4, 18] рассматривается алгоритм Imatch, который также основан на лексическом подходе. В нем из текста документа отбираются заданный процент таких терминов, которые не встречаются в слишком большом или в слишком маленьком числе текстов обрабатываемого массива. Цифрой отпечаток получается путем вычисления хэш-функции SHA-1 от строки, составленной из отсортированного списка отобранных терминов.

Для вычисления степени близости текстов x и y наибольшее распространение получили следующие меры:

$sim_{jaccard}(x,y)=\frac{|S(x)\cap S(y)|}{|S(x)\cup S(y)|}$ - мера Жаккарда, используется при описании текста в виде множества хэш-кодов [2, 1, 6, 18], где $S(x)$ и $S(y)$ - множества информационных признаков в текстах x и y;

$sim_{contain}(x,y)=\frac{|S(x)\cap S(y)|}{|S(x)|}$ - мера включения x в y, также обычно используется при описании текста в виде множества хэш-кодов [2], где $S(x)$ и $S(y)$ - множества информационных признаков в текстах x и y;

$sim_{\cos ine}(x, y) = \frac{x^T y}{\|x\|_2 \|y\|_2}$ - косинусная мера близости, используется при описании текста в виде вектора весов информационных признаков [9], где $x = (x_1, ..., x_m)$ - вектор весов признаков в тексте x и $y = (y_1, ..., y_m)$ - вектор весов признаков в тексте y, m - общее число различных признаков во всех документах;

$sim_{ham\min g}(x, y) = \sum_{i=1}^{m} |x_i - y_i|$ - мера Хэмминга, используется при представлении текстов в виде битовых векторов в лексических моделях цифровых отпечатков [8, 10, 11], где $x_i \in \{0,1\}$ - битовый признак с номером $i = 1,...,m$ у текста x, y_i - битовый признак с номером $i = 1,...,m$ у текста y.

2. Методы тематического упорядочения текстов

Задачу тематического упорядочения текстов заключается в нахождении такой перестановки заданного множества текстов, что тематически близкие тексты будут находиться рядом друг с другом, а тематически отличные - далеко. Исходя из приведенного определения, она является близкой по своему содержанию к задаче коммивояжера, задача одномерного размещения элементов, задаче иерархического кластерного анализа, а также задачам ранжирования документов при поиске текстов.

Рассмотрим сначала использования методов решения задач коммивояжера и одномерного размещения элементов [7, 15] для тематического упорядочения текстов.

В рамках задачи коммивояжера задача тематического упорядочения текстов формально определяется следующим образом. Требуется найти такую перестановку $(l_1, ..., l_n)$ номеров текстов $\{1, ..., n\}$, на которой достигается максимум функции

$$C_{TSP}(l_1, ..., l_n) = \sum_{i=2}^{n} sim(x_{l_{i-1}}, x_{l_i}),$$

где $sim(x, y)$ - мера близости между текстами x и y.

В рамках задачи одномерного размещения элементов формальная постановка задачи тематического упорядочения будет следующая. Требуется найти такую перестановку $(l_1, ..., l_n)$ номеров текстов $\{1, ..., n\}$, на которой достигается минимум функции

$$C_{PPP}(l_1, ..., l_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} |j - i| sim(x_{l_i}, x_{l_j})$$

Обе приведенные задачи являются NP-полными, что приводит к необходимости использования приближенных алгоритмов для возможности обработке наборов из нескольких сотен текстов. Приближенные алгоритмы можно разбить на следующие типы (в скобках указаны названия наиболее распространенных алгоритмов):

"жадные" алгоритмы – основаны на последовательном построении решения путем выбора локально-оптимального решения на каждом шаге (алгоритм ближайшего соседа);

методы локальной оптимизации – основаны на итерационном изменении решения с помощью преобразований из заданного множества до тех пока нельзя будет улучшить имеющееся решение (метод двойного выбора, метод тройного выбора, алгоритм Лина-Кернигана и д.р.);

методы случайного поиска – основаны на использовании методов статистического моделирования (генетические алгоритмы, прямое моделирование "Монте-Карло", эволюционные алгоритмы, моделирование отжига).

В настоящей работе остановимся на использовании наиболее быстрых приближенных методов, вычислительная сложность которых не более $O(n^3)$:

- метод ближайшего соседа;
- метод двойного выбора;
- метод перестановки смежных элементов;
- генетический алгоритм.

В методе ближайшего соседа получение решения $(l_1, ..., l_n)$ осуществляется следующим образом. Номер первого текста l_1 произвольным образом (обычно $l_1 = 1$). Номер l_{i+1} текста определяется путем нахождения ближайшего текста к l_i среди тех текстов, номера которых не присутствуют во множестве $(l_1, ..., l_i)$.

В методе двойного выбора для нахождения решения задачи коммивояжера сначала строят случайную перестановку текстов $(l_1^0, ..., l_n^0)$, которая рассматривается как замкнутый путь (гамильтонов цикл) в графе, у которого вершинами являются тексты. Далее рассматривают все возможные пары не смежных ребер (l_a, l_{a+1}), (l_b, l_{b+1}) и производится их замена на ребра (l_a, l_b) и (l_{a+1}, l_{b+1}), если это приводит к улучшению решения. Данная процедура повторяется до тех пор, пока нельзя будет улучшить имеющее решение.

В методе перестановке смежных элементов начальное решение строится случайным образом. Текущее решение $(l_1, ..., l_n)$ модифицируется путем перестановки соседних элементов l_i и l_{i+1}, если новый порядок имеет меньшую стоимость. Пусть $L_j = \sum_{s=1}^{j-1} sim(x_{l_s}, x_{l_j})$ - сумма весов элементов расположенных слева от элемента l_j,

$R_j = \sum_{s=j+1}^{n} sim\left(x_{l_s}, x_{l_j}\right)$ - сумма весов элементов расположенных справа от l_j. Перестановку l_i и l_{i+1} можно выполнить, если $L_i - R_i + R_{i+1} - L_{i+1} + 2sim\left(x_{l_i}, x_{l_{i+1}}\right) < 0$.

В методе на основе генетического алгоритма [19] на первом шаге формируется набор (популяция) T_1 из некоторого числа случайных перестановок текстов. На последующих шагах из имеющегося набора T_k производится формирование нового набора перестановок T_{k+1} путем объединения и преобразования имеющихся перестановок с помощью операторов скрещивания и мутации, и отбора перестановок с максимальным значением целевой функции.

Рассмотрим теперь использование агломеративных методов иерархического кластерного анализа [17] для решения задачи тематического упорядочения текстов. В данных методах производится последовательное объединение имеющего набора текстов $x_1,...,x_n$ во все более крупные классы. Схема типичного агломеративного алгоритма имеет следующий вид.

Схема агломеративного иерархического алгоритма

1. Положить $t = 0$, $\Omega^{(0)} = \left\{\omega_1^{(0)}, ..., \omega_n^{(0)}\right\}$, где $\omega_i^{(0)} = \{x_i\}$, $\delta_{ij}^{(0)} = \rho\left(x_i, x_j\right)$, $i, j = 1, ..., n$.
2. Построить разбиение $\Omega^{(t+1)}$ путем объединения классов $\omega_l^{(t)}$ и $\omega_r^{(t)}$ в разбиении $\Omega^{(t)}$, где $(l,r) = \arg \min_{i,j=1,...,n-t,\, i \neq j} \delta\left(\omega_i^{(t)}, \omega_j^{(t)}\right)$.
3. Если $t = n-1$, то завершить работу алгоритма, в противном случае положить $t = t+1$ и перейти к шагу 2. ■

Для нахождения приближенного решения воспользуемся тем фактом, что на каждом шаге $t = 1, ..., n-1$ работы агломеративного алгоритма разбиение $\Omega^{(t+1)}$ получается путем объединения двух ближайших классов $\omega_s^{(t)}$ и $\omega_r^{(t)}$ в разбиении $\Omega^{(t)}$, т. е. $(s,r) = \arg \min_{i,j=1,...,n-t,\, i \neq j} \delta\left(\omega_i^{(t)}, \omega_j^{(t)}\right)$, где δ – функция расстояния между классами.

В качестве приближенного решения будем использовать такую перестановку $\left(l_1^*, ..., l_n^*\right)$, в которой для любого $t = 1, ..., n-1$ элементы классов $\omega_s^{(t)}$ и $\omega_r^{(t)}$, которые входят в один больший класс, располагаются непосредственно друг за другом. Причем порядок следования элементов классов $\omega_s^{(t)}$ и $\omega_r^{(t)}$ должен быть таким, чтобы обеспечить максимум функции $C\left(l_1, ..., l_n\right)$.

Рассмотрим теперь рекурсивную процедуру нахождения перестановки $L^* = \left(l_1^*, ..., l_n^*\right)$. Пусть для элементов классов $\omega_s^{(t)}$ и $\omega_r^{(t)}$, объединяемых на шаге t, уже построены перестановки $L_s^* = \left(l_{s1}^*, ..., l_{sn_s}^*\right)$ и $L_r^* = \left(l_{r1}^*, ..., l_{rn_r}^*\right)$ соответственно, где n_s – число элементов в классе $\omega_s^{(t)}$, n_r – число элементов в классе $\omega_r^{(t)}$. Для произвольной перестановки $\left(u_1, ..., u_n\right)$ оператор $R(u,p)$, $p \in \{0,1\}$, определяется следующим образом:

$R\left(\left(u_1, ..., u_n\right), 0\right) \equiv \left(u_1, ..., u_n\right)$,

$R\left(\left(u_1, ..., u_n\right), 1\right) \equiv \left(u_n, u_{n-1}, ..., u_1\right)$.

Перестановка $L_{s\cup r}^*$ для объединенного класса $\omega_{l\cup r} = \omega_l \cup \omega_r$ получается следующим образом:

$\left(p_s^*, p_r^*\right) = \arg \max_{p_s, p_r \in \{0,1\}} sim\left(R_{n_s}\left(L_s^*, p_s\right), R_1\left(L_r^*, p_r\right)\right)$,

$L_{s\cup r}^* = \left(R\left(L_s^*, p_s^*\right), R\left(L_r^*, p_r^*\right)\right)$.

Несложно заметить, что вычислительная сложность данной процедуры с учетом проведения иерархического кластерного анализа составляет $O\left(n^2\right)$, где n – число текстов.

Рассмотрим теперь использование методов ранжирования текстов, применяемых в поисковых системах Интернет. Одним из наиболее распространенных методов является PageRank [3]. На его основе рядом авторов предложены алгоритмы для упорядочения текстов из произвольного множества по степени близости к некоторому эталонному тексту или набору текстов [14, 20]. В частности, в работе [14] предлагается следующий алгоритм.

Пусть W - матрица размера $n \times n$, где w_{ij} - мера близости текстов с номерами i и j, $D = diag\left(d_1, ..., d_n\right)$ - диагональная матрица, $d_i = \sum_{j=1}^{n} w_{ij}$, $S = D^{-1/2} W D^{-1/2}$, $f^{(0)} = \left(f_1^{(0)}, ..., f_n^{(0)}\right)$ - вектор весов текстов, задающий начальное ранжирование текстов, $\alpha \in (0,1)$ - параметр, $y = \left(y_1, ..., y_n\right)$ - вектор эталонных текстов (в настоящей работе полагаем, что $y = (1,0,...,0)$, т.е. упорядочение производится относительно первого текста), $t_{\max}$ - число итераций.

Вектор итогового ранжирования текстов $f^* = \left(f_1, ..., f_n\right)$ находится с помощью следующей итерационной процедуры:

$f(t+1) = \alpha S f(t) + (1-\alpha) y$, $t = 1, ..., t_{\max}$.

Несложно показать, что последовательность весов текстов $f(t)$ сходится к $f^* = (1-\alpha)(I - \alpha S)^{-1} y$. Для получения итоговой перестановки текстов производится упорядочение элементов вектора f^* по убыванию.

3. Эксперименты

Для экспериментального исследования эффективности различных подходов к решению задачи тематического упорядочения текстов были использованы массивы, приведенные в следующей таблице.

Таблица 1. Тестовые массивы для оценки тематического упорядочения

Название массива	Число текстов	Число рубрик	Комментарий
Yandex News	256	21	Массив построен путем ручного отбора и коррекции документов из тематически различных сюжетов в системе Яндекс Новости. Результаты автоматической группировки новостей в данном случае не учитывались.
Google News	511	24	Массив построен путем ручного отбора групп новостей по определенным тематикам с помощью поисковой системы Google. Группы организованы в виде двухуровневого дерева. Например, в группу сюжетов "НАТО Афганистан" попали сюжеты про события в населенных пунктах Гельменд, Лагман, Саидхель и др. Результаты автоматической группировки новостей в данном случае не учитывались.
Reuters 21578-6	935	6	Подмножество из 6 рубрик ("gold", "gnp", "gas", "nat-gas", "ship", "sugar") массива Reuters-21578.
ROMIP 2004	2000	173	Обучающая выборка для дорожки классификации нормативно-правовых документов РОМИП 2004 (первые 2000 текстов).
Reuters 21578	5000	142	Массив Reuters-21578, http://www.daviddlewis.com (первые 5000 текстов).

Для оценки качества работы алгоритмов с точки зрения конечного пользователя произведем оценку расположение эталонных номеров классов текстов в итоговой перестановке с помощью следующего показателя

$$C_{\text{var}}(l_1,...,l_n)=\sum_{j=1}^{k}\left(\frac{1}{n_j}\sum_{i\in\omega_j}\left(l_i-\bar{l}_j\right)^2\right)$$ - суммарная дисперсия, которая представляет собой сумму разбросов элементов эталонных классов $\omega_1,...,\omega_k$ в полученной перестановке $(l_1,...,l_n)$, где n_j - число элементов в классе ω_j, k - число эталонных классов. Заметим, что минимальное значение $C_{\text{var}}(l_1,...,l_n)$ достигается в том случае, когда тексты из одного класса расположены рядом. Так как $\sum_{i=1}^{n_j} i^2=\frac{1}{6}n_j(n_j+1)(2n_j+1)$, получаем, что для произвольной перестановки $(l_1,...,l_n)$ справедливо следующее неравенство

$$C_{\text{var}}(l_1,...,l_n)\geq\frac{1}{12}\sum_{j=1}^{k}\left(n_j^2-1\right).$$

В результате можно определить следующий относительный показатель «нормированная дисперсия»

$$C(l_1,...,l_n)=\frac{\sum_{j=1}^{r}\left(n_j^2-1\right)}{12\sum_{j=1}^{r}\left(\frac{1}{n_j}\sum_{i\in\omega_j}\left(l_i-\bar{l}_j\right)^2\right)},$$

который принимает значения из промежутка от 0 до 1.

Рассмотрим теперь результаты экспериментов. В табл. 2 приводятся оценки времени работы алгоритмов на тестовых массивах. При этом используются следующие обозначения: Hier – алгоритм на основе иерархического кластерного анализа, 2-opt – алгоритм на основе двойного выбора, NN – алгоритм ближайшего соседа, GA – генетический алгоритм, PPP – метод перестановки смежных элементов, Rank – алгоритм упорядочения текстов PageRank. Для представления текстов использовалась теоретико-множественная модель, а для вычисления расстояний косинусная мера близости. Необходимо отметить, что время вычисления расстояний между всеми парами текстов в данном случае не учитывалось.

Таблица 2. Время работы (сек)

Время (сек)	HIER	2OPT	Rank	NN	GA	PP
Yandex News	0.06	0.08	0.02	0.03	19	2.7
Google News	0.1	0.65	0.1	0.09	34	3.5
Reuters 21578_6	0.22	4.72	0.36	0.24	64	8.67
Romip 2004	1.7	86	2.5	1.8	133	90
Reuters 21578	13	>1500	36	6	398	>1500

В табл. 3 приводятся оценки показателя "нормированная дисперсия" для различных алгоритмов, моделей представления текстов и методов вычисления расстояний между текстами. В таблице используются следующие обозначения:

2OPT_TAIL – модификация алгоритма 2OPT, в которой производится циклический сдвиг полученной перестановки элементов таким образом, чтобы концевые элементы l_1 и l_n были максимально не похожи друг на друга;

TFIDF – для представления текстов используется стандартная теоретико-множественная модель, а для вычисления близости используется косинусная мера;

KGRAMM – текст представляется хэш-кодами всех последовательностей слов длины k, в качестве меры близости используется мера включения;

KGRAMM TFIDF – используется комбинированная мера близости, получаемая в результате объединения матриц с косинусной мерой, вычисленной на основе стандартных теоретико-множественных признаков, и мерой включения, вычисленной на основе к-грамм слов.

Для удобства анализа результатов в таблице в каждом столбце жирным шрифтом выделено наилучшее значение показателя, а курсивом следующие два наилучших показателя.

Таблица 3. Нормированная дисперсия

Алгоритм	**Модель**	**Yandex News**	**Google News**	**Reuters 21578_6**	**Romip**
PPP	TFIDF	0.0176	0.0148	0.0375	0.0014
	KGRAMM TFIDF	*0.0246*	0.0186	0.0375	0.0014
2OPT	TFIDF	0.0184	0.0163	0.0377	0.0015
	KGRAMM	0.0034	0.0104	0.0359	0.0014
	KGRAMM TFIDF	*0.0254*	0.0189	0.0377	0.0015
2OPT TAIL	TFIDF	**0.0350**	**0.0979**	0.0387	0.0015
	KGRAMM	0.0036	0.0088	0.0372	0.0014
HIER	KGRAMM	0.0048	0.0148	0.0363	0.0015
	TFIDF	0.0160	*0.0519*	**0.0427**	0.0015
	KGRAMM TFIDF	0.0158	0.0492	0.0427	0.0015
NN	KGRAMM	0.0037	0.0104	0.0348	**0.0016**
	TFIDF	0.0072	0.0506	0.0392	0.0013
	KGRAMM TFIDF	0.0081	*0.0515*	*0.0406*	0.0015
RANK	TFIDF	0.0049	0.0372	*0.0417*	0.0016
GA	TFIDF	0.0087	0.0089	0.0347	0.0014

Таким образом, проведенные эксперименты показывают, что наиболее эффективным с точки зрения качества работы являются методы двойного выбора (2OPT-TAIL) и на основе иерархического кластерного анализа (HIER). Наилучшие результаты, как правило, достигаются при использовании косинусной меры близости, либо комбинированной косинусной и к-граммной меры близости (меры включения). Методы, применяемые для ранжирования результатов поиска в поисковых системах Интернет в данном случае оказываются малопригодными.

Выводы и направления дальнейших исследований

Таким образом, в настоящей работе проведен сравнительный анализ эффективности использования различных методов для решения задач тематического упорядочения и устранения избыточной информации в текстах. Разработаны два новых алгоритма тематического упорядочения текстов, один основанный на использовании иерархических методов кластерного анализа, а другой основанные на модификации алгоритма приближенного решения задачи коммивояжера.

В качестве перспективных направлении дальнейших исследований можно выделить следующие:

- анализ эффективности процедур тематического упорядочения для обработки других типов документов: электронной почты, законодательных актов, служебных документов, научных работ и т.п.
- анализ эффективности использования других методов для тематического упорядочения текстов (например, методов на основе нейронных сетей Кохонена, методов проецирования данных на плоскость, методов ранжирования страниц в поисковых системах Интернет);
- анализ эффективности использования в алгоритмах устранения избыточной информации результатов выделения значимых фрагментов при классификации текстов;
- разработка эффективных алгоритмов устранения повторяющихся фрагментов в различных текстах из заданного массива.

Работа выполнена при поддержке гранта Президента Российской Федерации для государственной поддержки молодых российских ученых МК-12.2008.10.

Литература

[1] Brin S., Davis J., Garcна-Molina H. Copy detection mechanisms for digital documents // SIGMOD Rec. 24, 2, 1995. - 398-409. DOI= http://doi.acm.org/10.1145/568271.223855

[2] Broder A., Glassman S., Manasse M., Zweig G. Syntactic Clustering of the Web // DEC SRC Technical Note 1997-015, 1997. – 13 p.

[3] Brin S., Page L. The anatomy of a large scale hypertextual web search engine. In *Proc. 7th International World Wide Web Conf.*, 1998.

[4] Chowdhury A., Frieder O., Grossman D.A., McCabe M.C., Collection statistics for fast duplicate document detection // ACM Transactions on Information Systems, 20(2), 2002. – pp. 171-191

[5] Charikar M. Similarity estimation techniques from rounding algorithms // Proc. 34th Annual Symposium on Theory of Computing (STOC 2002). pp. 380-388.

[6] Forman G., Eshghi K., Chiocchetti S. Finding Similar Files in Large Document Repositories // KDD'05, 2005. – 8 p.

[7] Gutin G., Punnen A.P. The Traveling Salesman Problem and Its Variations (Combinatorial Optimization). Kluwer Academic Publishers, 2004. – 850 p.

[8] Henzinger M. Finding near-duplicate web pages: a large-scale evaluation of algorithms // Proceedings of the 29th Annual international ACM SIGIR Conference on Research and Development in information Retrieval, 2006. – pp. 284-291. DOI= http://doi.acm.org/10.1145/1148170.1148222

[9] Hoad T. C., Zobel J. Methods for identifying versioned and plagiarized documents // J. Am. Soc. Inf. Sci. Technol. 54, 3, 2003. - 203-215. DOI= http://dx.doi.org/10.1002/asi.10170
[10] Manku G. S., Jain A., Das Sarma A. Detecting near-duplicates for web crawling // Proceedings of the 16th international Conference on World Wide Web, 2007. – pp. 141-150. DOI= http://doi.acm.org/10.1145/1242572.1242592
[11] Seo J., Croft W. Local Text Reuse Detection // SIGIR'08, July 20–24, 2008, Singapore. – 8 p.
[12] Shivakumar N., Garcia-Molina H. Finding near-replicas of documents on the web // Proceedings of Workshop on Web Databases (WebDB'98), 1998. pp. 204-212.
[13] Schleimer, S., D.S. Wilkerson and A. Aiken "Winnowing: Local Algorithms for Document Fingerprinting", SIGMOD 2003, Jun. 9-12, 2003.
[14] D. Zhou, J. Weston, A. Gretton, O. Bousquet and B. Schlkopf. Ranking on data manifolds. In Proceedings of NIPS'2003.
[15] Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. – М. Вильямс, 2007. – 384 с.
[16] Васильев В.Г. Комплексная технология автоматической класси-фикации текстов. Компьютерная лингвистика и интеллек-туальные технологии: по материалам ежегодной международ-ной конференции "Диалог" (Бекасово, 4-8 июня, 2008). Вып. 7(14). – М. РГГУ. ISBN 978-5-7281-1022-4. с. 83-90.
[17] Васильев В.Г., Кривенко М.П. Методы автоматизированной обработки текстов. – М. ИПИ РАН, 2008. – 304 с.
[18] Зеленков Ю.Г, Сегалович И.В Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL'2007, Переславль-Залесский, Россия, 2007. – 9р.
[19] Корнеенко В.П. Методы оптимизации. Учебник. – М.: Высшая школа, 2007. – 664 с.
[20] Тарасов С.Д. Автоматическое составление обзорных рефератов новостных сюжетов // Труды 10-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2008, Дубна, Россия, 2008.

Thematical arrangement of texts for creating digests

© Vitaly Vasilyev
vvg_2000@mail.ru

In this work new approach for digest creating is suggested. It is based on thematical arranging texts collection in such way that thematically similar texts are placed close to each other in the resulting list of documents. Different methods for solving this task are proposed and experimentally evaluated on wide range of collections.