

# *Метод классификации текстовых документов, основанный на полнотекстовом поиске*

В.И. Шабанов  
vs@rambler-co.ru

А.М. Андреев  
arca@inteltec.ru

## *Аннотация*

В статье рассматривается метод тематической классификации и метод обучения классификатора, основанные на применении алгоритмов полнотекстового поиска и ранжирования. На этапе обучения классификатора автоматически формируются списки поисковых запросов, характерных для документов каждой рубрики. На этапе классификации выполняется поиск этих запросов во всех обрабатываемых документах, ранжирование результатов поиска и вычисление мер подобия рубрика-документ.

## *Введение*

С каждым годом увеличивается объем доступных пользователю массивов текстовой информации, и поэтому становится все более актуальной задача поиска необходимых пользователю документов в таких массивах. Для решения этой задачи часто применяются различные тематические классификаторы, рубрикаторы и т. д., которые позволяют искать (автоматически или вручную) документы в небольшом подмножестве документной базы, соответствующем интересующей пользователя тематике.

В данной статье рассматривается способ построения системы автоматической классификации, способной обучаться на заданных пользователями образцах. Такая система может быть использована для повышения точности поиска в информационно-поисковой системе, формирования тематических коллекций страниц Интернет (focused crawling), а также для автоматической фильтрации сообщений.

## Постановка задачи

Процедуру автоматической классификации множества документов  $D=\{d_i\}$  на множестве рубрик  $R=\{r_j\}$ , организованных в иерархию  $H=\{<r_j, r_p> : r_j, r_p \in R\}$ , сформулируем следующим образом:

$$\text{Classify}(R, H, Sem, D) \rightarrow \{<d_i, r_j, f_{ij}> : d_i \in D, r_j \in R\},$$

где:

$Sem = \{Sem(r_j), r_j \in R\}$  – классификационные признаки рубрик,  $Sem(r_j)$  – признаки рубрики  $r_j$ .

$f_{ij}$  – мера подобия документа  $d_i$  рубрике  $r_j$  ( $0 \leq f_{ij} \leq 1$ )

Пусть мы имеем обучающую выборку  $T_{teach}=\{<d_i, r_j> : d_i \in D, r_j \in R\}$ , состоящую из документов  $d_i$ , которые пользователи-эксперты просмотрели и вручную приписали к соответствующим рубрикам  $r_j$ . Тогда обучение классификатора состоит в формировании классификационных признаков  $Sem$  на основе анализа обучающей выборки:

$$\text{Teach}(R, H, T_{teach}) \rightarrow Sem$$

В большинстве известных методов классификации признаки рубрик (их также можно назвать семантическими образами, поскольку они отражают семантику данной рубрики) явно или неявно содержат список характерных для этой рубрики терминов. Термины должны присутствовать в тексте документа для того, чтобы можно было принять решение о соответствии или несоответствии документа рубрике. В большинстве случаев в качестве терминов используются одиночные слова [6, 7, 10, 11], реже – словосочетания [11] или граммы [8].

Все такие методы основаны на том факте, что большинство тематик имеют множества характерных для них терминов. Например, для текстов юридической тематики это *вендикационный иск, конфискация имущества, правообладатель* и т. д.

Как правило, традиционные системы классификации построены на использовании однословных терминов и симметричной процедуры сопоставления. Схематично их алгоритмы обучения и классификации состоят из следующих шагов:

### 1. Обучение

- 1.1. Выделить из документов обучающей выборки списки слов (иногда для слова дополнительно запоминается частота встречаемости и/или вес);
- 1.2. Профильтровать списки слов в соответствии с их морфологическими и статистическими характеристиками [10];

- 1.3. Построить семантические образы рубрик (например, вычислить условные вероятности соответствия документа рубрике при наличии в нем заданного слова [10,11], или обучить нейронную сеть [7]);
2. Классификация
  - 2.1. Выделить из документов, поданных на классификацию, списки слов при помощи той же самой процедуры, которая использовалась на шаге 1.1;
  - 2.2. Сопоставить эти списки с семантическими образами рубрик (например, вычислить вероятность соответствия документа рубрике).

Использование однословных терминов накладывает ограничения на эффективность классификации, так как слова часто имеют несколько смысловых значений.

Существует большое количество слов, которые сами по себе не могут быть надежными классификационными признаками. Пример: наличие в тексте документа слов *холодная*, *быстрая*, *сварка*, *сталь* не обязательно означает, что текст относится к теме «автохимия». Однако, наличие в тексте словосочетаний *холодная сварка* или *быстрая сталь* позволяет с большей степенью уверенности сделать вывод о тематике этого текста.

Как видно из описанного выше алгоритма, модифицировать его для использования словосочетаний совсем непросто, поскольку на шаге 1.1 и 2.1 система должна выделять одни и те же словосочетания, даже если в тексте классифицируемого документа они сформулированы не так, как в тексте документа обучающей выборки. Пример: пусть на шаге 1.1 мы выделили из обучающего документа словосочетание *сопоставление рубрики с документом*. Тогда для успешной классификации следующего абзаца мы на шаге 2.1 должны выделить из него это же словосочетание.

Для того чтобы иметь возможность нестрогого сопоставления терминов семантического образа рубрики с текстом документа, в данной работе предлагается отказаться от симметричной схемы и заменить ее нечетким поиском словосочетаний в тексте документа. Под нечеткостью в данном случае подразумевается то, что поиск словосочетания считается успешным, если слова, из которых оно состоит, содержатся в некотором небольшом фрагменте текста, причем сопоставление нечувствительно к изменению грамматической формы слов.

Пример: при поиске словосочетания *двухрычажная подвеска* будут успешно сопоставлены следующие фрагменты текстов:

- ...диагностика подвесок (Mc Pherson, двухрычажная и т. д.) ...

- ...при разработке подвески новой модели Honda Civic разработчики отказались от двухрычажной схемы в пользу более компактного Mc Pherson...

## ***Семантические образы рубрик***

В предлагаемом методе семантический образ рубрики состоит из взвешенного списка поисковых запросов, по которым может быть найден соответствующий рубрике документ:

$$\mathbf{Sem}(r) = \langle \theta(r), l(r) \rangle, \quad (1)$$

где:

$\theta(r) = \{ \langle t_i, f_i, w_i \rangle \}, i = 1 \dots n$  - список терминов (поисковых запросов) рубрики  $r$  ( $r \in \mathbf{R}$ );

$l(r)$  - пороговое значение классифицирующей функции (см. ниже), комбинирующей поисковые ранги в общую меру подобия документа рубрике.

$t_i$  - поисковый запрос;

$f_i$  - вес поискового запроса (значимость запроса  $t_{ji}$  для рубрики  $r$ );

$w_j$  - пороговое значение поискового веса (ранга), начиная с которого поиск по данному запросу начинает учитываться при классификации;

Процедура обучения классификатора заключается в обнаружении специфических для каждой из рубрик рубрикатора  $\mathbf{R}$  поисковых запросов, формировании численных мер значимости каждого из этих запросов относительно соответствующей рубрики, а также вычислении пороговых значений поискового веса.

Процедура классификации состоит в индексировании анализируемого документа  $d$  (построении для него инверсного индекса  $\Gamma^I(d)$ , см. ниже), и рекурсивном обходе дерева рубрик с выполнением на каждом уровне иерархии полнотекстового поиска по терминам, содержащимся во множествах  $\mathbf{Sem}(r)$  всех рубрик данного уровня. Все успешно найденные в  $\Gamma^I(d)$  термины ранжируются, а на основании полученных рангов вычисляется мера подобия документа каждой из рубрик.

Решение о соответствии или несоответствии документа рубрике принимается после сравнения этой меры с вычисленными во время обучения пороговыми значениями  $l(r_i)$ . Аналогично [9] рекурсив-

ный обход дерева рубрик выполняется лишь для нескольких рубрик текущего уровня, для которых мера подобия максимальна.

Преимущества метода:

- 1) **высокая производительность** – вычисление поисковых запросов на инверсном индексе можно делать очень быстро.
- 2) **возможна быстрая пакетная классификация** – при построении общего инверсного индекса для группы документов, становится возможной одновременная классификация всех проиндексированных документов.
- 3) **высокая точность** – пользуясь данным методом можно поместить в семантический образ рубрики слова, словосочетания и даже целые фразы, при наличии которых в тексте можно «почти наверняка» утверждать, что документ соответствует рубрике.
- 4) **полнота классификации** – поиск может найти словосочетание в тексте, даже если его формулировка изменена.
- 5) **возможность ручной настройки классификатора** – структура семантических образов рубрик проста для понимания, добавление или удаление терминов (поисковых запросов) приводит ко вполне предсказуемым последствиям.
- 6) **возможность учета гиперссылок** – полнотекстовый поиск легко можно модифицировать так, чтобы он учитывал гипертекстовую структуру документов.

### **Представление классифицируемого документа**

Документ, подаваемый на вход процедуре классификации, представляется инвертированным индексом со следующей структурой:

$$\Gamma^I(d) = \langle \tau, \nu(\tau_i), c(\tau_i), z(\tau_i) \rangle,$$

где

$\tau = \{\tau_i\}, i=1 \dots n$  – множество слов документа  $d$ ;

$n$  – количество слов в документе (размер словаря документа);

$\nu(\tau_i) \in \mathbf{R}$  – вес (значимость) слова  $\tau_i$  в документе;

$c(\tau_i) = \{c_{ij}\}, j=1 \dots n(\tau_i), c_{ij} \in \mathbf{N}$  – вектор координат (номеров позиций), в которых встречается слово  $\tau_i$ .

$z(\tau_i) = \{z_{ij}\}, j=1 \dots n(\tau_i), z_{ij} \in \mathbf{R}$  – вектор весов зон документа (название, заголовков и т. д.), в которых встречается слово  $\tau_i$ .

$g(\tau_i) = \{g_{ij}\}, j=1 \dots n(\tau_i), g_{ij} \in \mathbf{R}$  – вектор информации о грамматических формах слова  $\tau_i$  в тексте документа.

Данное представление позволяет вычислять следующие виды поисковых запросов:

- поиск одного слова:

$$Q_l = \tau \rightarrow \text{истина, если } \tau \in \tau \quad (2)$$

- поиск группы слов, находящихся в тексте на расстоянии не более  $l$  слов.

$$\begin{aligned} Q_{\&} = (l, \tau_1 \& \tau_2 \& \dots \& \tau_k) \rightarrow \text{истина, если} \\ \tau_1 \dots \tau_k \in \tau \\ \wedge \exists a \in \mathbf{N} : \\ [(\exists c_{1j} \in \mathbf{c}(\tau_1) : a \leq c_{1j} \leq a+l) \\ \wedge (\exists c_{2j} \in \mathbf{c}(\tau_2) : a \leq c_{2j} \leq a+l) \\ \wedge \dots \\ \wedge (\exists c_{kj} \in \mathbf{c}(\tau_k) : a \leq c_{kj} \leq a+l)] \end{aligned} \quad (3)$$

- то же самое, но с фиксацией порядка слов в тексте:

$$\begin{aligned} Q_{\#} = (l, \tau_1 \# \tau_2 \# \dots \# \tau_k) \rightarrow \text{истина, если} \\ \tau_1 \dots \tau_k \in \tau \\ \wedge \exists a \in \mathbf{N} : \\ [(\exists c_{1j} \in \mathbf{c}(\tau_1) : a \leq c_{1j} \leq a+l) \\ \wedge (\exists c_{2j} \in \mathbf{c}(\tau_2) : a \leq c_{2j} \leq a+l) \\ \wedge \dots \\ \wedge (\exists c_{kj} \in \mathbf{c}(\tau_k) : a \leq c_{kj} \leq a+l) \\ \wedge a \leq c_{1j} < c_{2j} < \dots < c_{kj} \leq a+l] \end{aligned} \quad (4)$$

- буквальный поиск (поиск слов, записанных в точности так, как они перечислены в запросе):

$$Q_{seq} = "\tau_1 \tau_2 \dots \tau_k" \equiv (k, \tau_1 \# \tau_2 \# \dots \# \tau_k) \quad (5)$$

- поиск альтернатив (поиск одного из заданных слов):

$$\begin{aligned} Q_l = \tau_1 | \tau_2 | \dots | \tau_k \rightarrow \text{истина,} \\ \text{если } \exists \tau_m \in \tau_1 \dots \tau_k : \tau_m \in \tau \end{aligned} \quad (6)$$

- любая комбинация из перечисленных выше видов запросов (составной запрос), например

$$Q = "\tau_1 \tau_2" | (5, \tau_3 \& \tau_4) | (6, \tau_5 \# \tau_6 \# \tau_7) | \dots$$

При этом векторы координат  $\mathbf{c}(Q)$  любого из подзапросов состоят из величин  $a$  из выражений 3 и 4;

Помимо, собственно, проведения поиска, данное представление позволяет выполнять ранжирование результатов, при котором учитываются следующие факторы:

- расстояние между словами в тексте – чем ближе слова находятся друг от друга, тем выше вес. Для вычисления этой составляющей веса используются вектора  $c(\tau_i)$ ;
- вхождение слов в важные зоны документа (заголовки, названия разделов и т. д.). Для учета этого фактора используются вектора  $z(\tau_i)$
- совпадение грамматической формы слова с той, которая указана в запросе – вес повышается, если в тексте слово находится в такой же или близкой форме (другой падеж того же существительного) и понижается, если слово в тексте отличается.

## Классификация документов

Оценка, соответствует документ рубрике, или нет, ведется по следующим параметрам:

- документ соответствует рубрике, если запросы, по которым он был найден, имеют высокий поисковый ранг;
- документ скорее соответствует рубрике, если он найден по большому количеству запросов из ее семантического образа.
- доля поисковых запросов рубрики, по которым был найден документ, среди всех запросов, по которым он был найден, должна быть велика.

Ниже показан алгоритм классификации одного документа, учитывающий эти характеристики.

*Classify* ( $R, H, Sem, d$ )

1. [документ не приписан ни к одной из рубрик]  $X = \emptyset$
2. [индексирование]  $\Gamma^I(d) = \langle \tau, v(\tau_i), c(\tau_i), z(\tau_i) \rangle$
3. [начинаем обход с корневой рубрики]  $r = r_0$
4. [рекурсивный спуск]
  - 4.1.  $X(r_i) = Search(r_i, \Gamma^I(d))$
  - 4.2.  $X = X \cup X(r_i)$
  - 4.3. Выполнить шаг 4 для  $n$  рубрик из  $X(r_i)$ , для которых мера соответствия документа рубрике максимальна.
5. результат =  $X$

Данный алгоритм использует функцию  $Search(r^*, \Gamma^I(d))$ , которая состоит из следующих шагов:

1.  $\forall r_i \in \{r : \langle r^*, r \rangle \in H\}$  [для всех рубрик, дочерних по отношению к рубрике  $r^*$ ] выполнить
  - 1.1.  $\forall \langle t_{ij}, f_{ij}, w_{ij} \rangle \in \theta(r_i)$  [для всех терминов рубрики]
    - 1.1.1. преобразовать  $t_{ij}$  в поисковый запрос  $Q$ : для однословного термина запрос имеет вид  $Q_I$ , а для многословного –  $Q_{\&}$ ;

- 1.1.2. выполнить полнотекстовый поиск по запросу  $Q$ .
- 1.1.3. если результат – истина, вычислить поисковый ранг  $f(t_{ij})$ , иначе – принять  $f(t_{ij})$  равным 0. Ранг вычисляется следующим образом:

$$f(t_{ij}) = \text{rank}(Q, \mathbf{I}^I(d), f_{ij}). \quad (7)$$

- 1.1.3.a.  $\text{rank}(Q, \mathbf{I}^I(d), f_{ij})$  является функцией от поисковых рангов отдельных вхождений слов запроса в текст:

$$\text{rank}(Q, \mathbf{I}^I(d), f_{ij}) = 1 - (1 - f_{ij} \cdot w_{\max}) \prod_{k=1, k \neq k_{\max}}^{N_{\text{subq}}} (1 - f_{ij} \cdot w_k)^a \quad (8)$$

где:

$w_k$  – вес  $k$ -ого вхождения,  $k = 1 \dots N_{\text{subq}}$

$w_{\max} = \max(w_k)$

$k_{\max} = \arg \max(w_k)$

$a$  – настроечный параметр

- 1.1.3.b. вес  $k$ -ого вхождения однословного термина  $\tau$  с координатой  $c_k \in \mathbf{c}(\tau)$  определяется следующим образом:

$$w_k = \nu(\tau) \cdot z_k \cdot \text{gram}(\tau, g_k) \quad (9)$$

где:

$\nu(\tau) \in \mathbf{R}$  – вес (значимость) слова  $\tau$  в документе;

$z_k \in \mathbf{z}(\tau)$  – вес зоны документа (название, заголовков, обычный текст и т. д.), в которой находится вхождение термина  $\tau$  с координатой  $c_k$

$g_k \in \mathbf{g}(\tau)$  – номер грамматической формы термина с координатой  $c_k$

$\text{gram}(\tau, g_k)$  – мера расстояния между грамматической формой в запросе  $Q$  и формой в тексте документа  $g_k$ .

- 1.1.3.c. вес  $k$ -ого вхождения многословного термина  $\tau_1$  &  $\tau_2$  & ... &  $\tau_m$  с координатами, соответственно,  $c_{k1} \in \mathbf{c}(\tau_1)$ ,  $c_{k2} \in \mathbf{c}(\tau_2)$ , ...  $c_{km} \in \mathbf{c}(\tau_m)$ , определяется следующим образом:

$$w_k = \left( \prod_{x=1}^m \nu(\tau_x) \cdot z_{kx} \cdot \text{gram}(\tau_x, g_{kx}) \right)^a \left( \prod_{x=2}^m \text{dist}(c_{kx-1}, c_{kx}) \right)^b \quad (10)$$

где:

$\nu(\tau_x)$ ,  $z_{kx}$ ,  $g_{kx}$   $\text{gram}(\tau, g_k)$  – аналогичны выражению 9;

$dist(c_1, c_2)$  – функция, оценивающая вероятность информативности двух слов с координатами  $c_1$  и  $c_2$ , учитывающая расстояние между словами  $|c_1 - c_2|$ , а также их порядок в тексте  $sign(c_1 - c_2)$ . Максимальный вес получают вхождения, где слова запроса идут подряд в точности в том же порядке, в котором они перечислены в запросе;

$a, b$  – настроечные параметры.

1.1.4. [первая пороговая фильтрация] если  $f(t_{ij}) < w_{ij}$ , принять  $f(t_{ij})$  равным 0;

1.1.5. если  $f(t_{ij}) > 0$ , запомнить кортеж:

$$Z(d) = Z(d) \cup \langle r_i, t_{ij}, f(t_{ij}) \rangle$$

2.  $\forall r_i \in \{r : \langle r, r \rangle \in H\}$  вычислить меру сходства документа с этой рубрикой по следующей формуле:

$$similarity(d, r_i) = f_1^a \cdot f_2^b \cdot f_3^c \quad (11)$$

где:

$f_1$  – функция от поисковых рангов всех запросов рубрики  $r_i$ , по которым был найден документ:

$$f_1 = 1 - \prod_{\langle t_{ij}, f(t_{ij}) \rangle \in Z(r_i, d)} (1 - f(t_{ij})) \quad (12)$$

$$Z(r_i, d) = \{ \langle t_{ij}, f(t_{ij}) \rangle : \langle r_i, t_{ij}, f(t_{ij}) \rangle \in Z(d) \} \quad (13)$$

$f_2$  – доля поисковых запросов рубрики  $r_i$  от всех поисковых запросов, по которым найден  $d$ :

$$f_2 = \frac{|Z(r_i, d)|}{|Z(d)|} \quad (14)$$

$f_3$  – доля поисковых запросов рубрики  $r_i$ , по которым найден  $d$ , от всех поисковых запросов рубрики  $r_i$ :

$$f_3 = \frac{|Z(r_i, d)|}{|\theta(r_i)|} \quad (15)$$

$a, b, c$  – настроечные параметры.

3. Фильтрация результатов:

3.1.  $similarity(r_i, d) = 0$ , если  $|Z(r_i, d)| < Z_{min}$  – исключить из рассмотрения связи рубрика-документ, найденные по слишком маленькому количеству терминов;

3.2.  $similarity(r_i, d) = 0$ , если  $similarity(r_i, d) < l(r_i)$  – исключить из рассмотрения связи рубрика-документ, вес которых ниже

порогового значения, полученного во время обучения классификатора (см. ниже);

3.3. если  $|\{r_k : \text{similarity}(r_k, d) > 0\}| > S_{max}$ , исключить документ из рассмотрения на данном уровне иерархии, т.к. он относится к слишком большому количеству рубрик.

4. Приписать документ ко всем рубрикам, для которых  $\text{similarity}(r_k, d) > 0$ . В случае, когда число таких рубрик велико, приписать документ не ко всем рубрикам, а только к  $n$  рубрик, для которых величина  $\text{similarity}(r, d)$  максимальна. Запомнить результаты в  $X = \{<d, r_k, \text{similarity}(r_k, d)>\}$ .

5. Конец алгоритма.

Как видно из описания алгоритма, он легко может быть адаптирован так, чтобы выполнять одновременную классификацию большого корпуса документов  $D$  с использованием общего для всего корпуса инвертированного индекса  $I^T(D)$ . В этом случае процедура автоматической классификации имеет вид:

$Classify(R, H, Sem, D) \rightarrow \{<d, r, f_{dr}>\}$

## ***Представление документов обучающей выборки***

Обучение классификатора, основанного на полнотекстовом поиске, состоит в обнаружении специфических для каждой из рубрик терминов и формировании для каждого термина численной меры значимости, а также порогового значения поискового веса. Дополнительно в процессе обучения выполняется вычисление порогового значения меры подобия документа рубрике.

Документ обучающей выборки представлен в виде взвешенного множества терминов:

$$d = \{<t_k, \tau_k, w_k>\}, k = 1 \dots n_{terms}, \quad (16)$$

где

$t_k$  – термин (нормализованное представление);

$\tau_k$  – текст (список слов) самой частой реализации термина;

$w_k$  – значимость термина для документа  $d$ .

Обозначим

$terms(d) = \{t_k : <t_k, \tau_k, w_k> \in d\}$  – множество терминов документа  $d$ .

$weight(d, t_k) = w_k : <t_k, \tau_k, w_k> \in d$  – значимость термина  $t_k$  в документе  $d$ .

Выделение, нормализация и оценка значимости терминов рассмотрены в [1]. В данной статье рассматривается процедура, которая получает на входе множество рубрик  $R$ , иерархию  $H$  и обучаю-

шую выборку  $T_{teach}$ , в которой каждый документ представлен в соответствии с выражением 16.

Процедура обучения обладает следующими характеристиками:

- Классификация документов, на которых выполнялось обучение, выполняется с точностью, близкой к 1.
- Баланс между точностью классификации (числом ложных срабатываний процедуры *Search*) и полнотой (количеством распознаваемых формулировок) обеспечивается за счет формирования множеств терминов с  $n$ -кратной избыточностью и соответствующего требования  $n$ -кратного покрытия классифицируемого документа терминами.
- Выбор терминов в семантические образы рубрик происходит на основе анализа их различительной силы [2,3,4]. Данный метод хорошо себя зарекомендовал в задачах анализа текстов, и, как правило, приводит к попаданию в семантический образ терминов, точно отражающих тематику рубрики.

Алгоритм процедуры обучения  $Teach(\mathbf{R}, \mathbf{H}, T_{teach})$  состоит из следующих шагов:

1.  $\forall d \in D_{teach}$  : преобразовать  $d$  ко внутреннему представлению (см. выражение 16);
2.  $r = r_0$ ;
3. найти рубрики дочерние по отношению к  $r$ :

$$\mathbf{R}(r) = \{r_i : \langle r, r_i \rangle \in \mathbf{H}\}$$

$$\mathbf{H}(r) = \{\langle r, r_i \rangle : \langle r, r_i \rangle \in \mathbf{H}\}$$

4.  $\forall r_i \in \mathbf{R}(r)$  построить подмножество обучающей выборки, соответствующее поддереву рубрикатора, растущему из  $r_i$ :

$$T_{teach}(r_i) = \{\langle d, r_j \rangle : r_j \in \mathbf{Sub}(r_i)\}$$

5. Выполнить обучение на «плоском» рубрикаторе, состоящем из непересекающихся классов (рубрик)  $r_i \in \mathbf{R}(r)$ , каждому из которых соответствует подмножество обучающей выборки  $T_{teach}(r_i)$ . Запомнить получившиеся в результате обучения семантические образы рубрик из  $\mathbf{R}(r)$ :

$$Sem(r) = TeachSingle(\mathbf{R}(r), \{T_{teach}(r_i)\})$$

6. Выполнить процедуру  $Classify\left(\mathbf{R}(r), \mathbf{H}(r), \bigcup_{r_i \in \mathbf{R}(r)} T_{teach}(r_i)\right)$

7. Определить пороговые значения поискового веса  $w_{ij}$  (см. выражение 1) для всех терминов всех рубрик на основе данных, полученных на шаге 6. Подробное описание данной операции см. ниже;

8. Определить  $l(r)$  – пороговое значение меры соответствия классифицируемого документа рубрике  $r$ . Подробное описание данной операции см. ниже;
9. Повторить шаги 3-8 рекурсивно для всех рубрик из  $\mathbf{R}(r)$ .
10. Конец алгоритма

### Формирование семантических образов рубрик одного уровня иерархии

Наиболее сложный этап обучения классификатора содержится в процедуре *TeachSingle*, которая формирует семантические образы рубрик. Данная процедура вычисляет для каждой пары термин \* рубрика приближенное значение различительной силы по следующей формуле:

$$discr(t_k, r_i) = W_{stat} \cdot \left( \frac{q_{ki}}{Q_i} \right)^{a_2} \cdot \left( \frac{Q - Q_i}{q_k - q_{ki}} \right)^{a_3} \quad (17)$$

где:

$W_{stat}$  – различительная сила термина без учета распределения документов по рубрикам (см. [4]);

$$W_{stat i} = \frac{1}{\sum_{i=1}^N \overline{f_i^2}} \left[ \frac{\sum_{i=1}^N \overline{f_i^2} \cdot \overline{f_i^2}}{\sum_{i=1}^N \overline{f_i^2}} - \overline{f_i^2} \right] \quad (18)$$

$\overline{f_i}$  – среднее число появлений термина  $t_i$  в документах обучающей выборки;

$\overline{f_i^2}$  – средний квадрат числа появлений термина  $t_i$  в документах;

$q_{ki}$  – количество документов, приписанных рубрике  $r_i$ , содержащих термин  $t_k$ ;

$$q_{ki} = |\{d_n : d_n \in \mathbf{D}_{teach}, \langle d_n, r_i \rangle \in \mathbf{T}_{teach} \Rightarrow t_k \in terms(d_n)\}| \quad (19)$$

$q_k$  – количество документов обучающей выборки, содержащих термин  $t_k$ ;

$$q_k = |\{d_n : d_n \in \mathbf{D}_{teach} \Rightarrow t_k \in terms(d_n)\}| \quad (20)$$

$Q_i$  – количество документов, приписанных к рубрике  $r_i$ ;

$$Q_i = |\{d_n : d_n \in \mathbf{D}_{teach}, \langle d_n, r_i \rangle \in \mathbf{T}_{teach}\}| \quad (21)$$

$$Q - \text{ количество документов в обучающей выборке.} \\ Q = | \mathbf{D}_{teach} | \quad (22)$$

Итак, алгоритм обучения рубрик одного уровня иерархии выглядит следующим образом:

1. Вычислить величины  $discr(t_k, r_i)$  для каждого термина  $t_k$  и каждой рубрики  $r_i$  по формуле 17.
2. Для каждого термина  $t_k$  выбрать  $r(t_k)$  – рубрику, относительно которой его различительная сила максимальна:
 
$$r(t_k) = \arg \max_i discr(t_k, r_i)$$
3. Для каждой рубрики  $r_j$  выполнить шаги 3.1 – 3.4.
  - 3.1. Определить множество документов, приписанных рубрике  $r_j$  и ее подрубрикам:
 
$$\mathbf{D}_{sub} = \{ d \in \mathbf{D}_{teach} : (\exists r_k \in Sub(r_j) : \langle d, r_k \rangle \in T_{teach}) \}$$
  - 3.2. Выбрать минимальное подмножество терминов  $\{t_k\}$ , для которых  $r(t_k) = r_j$ , и которое покрывает все документы множества  $\mathbf{D}_{sub}$  как минимум  $n_{cover}$  раз, причем сумма величин  $discr(t_k, r_j)$  которого максимальна. ( $n_{cover}$  – некоторая константа);
  - 3.3. Выбранное на предыдущем шаге подмножество запомнить в семантическом образе рубрики  $\theta(r_i)$ , вес каждого из терминов принять равным  $discr(t_k, r_j)$ .
  - 3.4. Нормировать веса терминов семантического образа рубрики
4. Конец работы алгоритма. Множества признаков рубрик сформированы.

### ***Вычисление пороговых весов терминов и рубрик***

После того, как исходные семантические образы сформированы, необходимо вычислить пороговые значения поискового веса каждого из терминов, а также пороговые значения меры подобия документа рубрике  $l(r)$ . Данный этап позволяет удалить из признаков рубрик термины, которые приводят к ложным срабатываниям.

Вычисление пороговых значений делается путем использования обратной связи от процедуры автоматической классификации, которой подали на вход документы обучающей выборки и семантические образы рубрик с нулевыми порогами.

Алгоритм вычисления пороговых значений выглядит следующим образом:

1. [индексирование документов обучающей выборки]  $\Gamma^I(\mathbf{D}_{teach}) = \{ \langle \tau_j, v(\tau_j), c(\tau_j), z(\tau_j) \rangle \}$

2. [начинаем обход с корневой рубрики]  $r = r_0$
3. [рекурсивный спуск]

3.1. *CalculateBounds* ( $r, \Gamma^I(\mathbf{D}_{teach})$ )

3.2. Выполнить шаг 3 для всех рубрик из  $\mathbf{R}(r)$

Данный алгоритм использует процедуру *CalculateBounds*, которая выполняет полнотекстовый поиск и ранжирование, а затем вычисляет пороговые значения весов. Ее алгоритм состоит из следующих шагов:

1.  $\forall r_i \in \mathbf{R}(r)$  выполнить
  - 1.1.  $\forall \langle t_{ij}, f_{ij} \rangle \in \theta(r_i)$ 
    - 1.1.1. преобразовать  $t_{ij}$  в поисковый запрос  $Q$ : для однословного термина запрос имеет вид  $Q_I$ , а для многословного –  $Q_\&$ ;
    - 1.1.2. выполнить полнотекстовый поиск по запросу  $Q$  на обратном индексе  $\Gamma^I(\mathbf{D}_{teach})$ . Результат – множество документов  $\mathbf{D}(Q) \subset \mathbf{D}_{teach}$
    - 1.1.3.  $\forall d \in \mathbf{D}(Q)$  вычислить поисковый ранг  $f(d, t_{ij})$  согласно выражению 7, иначе – принять  $f(d, t_{ij})$  равным 0;
    - 1.1.4. запомнить кортеж  $\mathbf{Z} = \mathbf{Z} \cup \langle d, t_{ij}, f(d, t_{ij}) \rangle$
2. [анализ терминов]  $\forall t_k \in \bigcup_{r_i \in \mathbf{R}(r)} \theta(r_i)$ 
  - 2.1.  $\mathbf{Z}(t_k) = \{ \langle d, t_k, f(d, t_k) \rangle : \langle d, t_k, f(d, t_k) \rangle \in \mathbf{Z} \}$  – результаты поиска по запросу, построенному из термина  $t_k$
  - 2.2. если  $\mathbf{Z}(t_k) = \emptyset$ , удалить термин  $t_k$  из  $\theta(r_i)$
  - 2.3. отсортировать  $\mathbf{Z}(t_k)$  по убыванию  $f(d, t_k)$
  - 2.4. двигаясь последовательно по  $\mathbf{Z}(t_k)$  найти первый элемент  $\langle d, t_k, f(d, t_k) \rangle$ , для которого выполняется:
 
$$\langle t_k, f_{ik} \rangle \in \theta(r_i) \wedge \langle d, r_x \rangle \in \mathbf{T}_{teach} \wedge r_x \neq r_i \quad (23)$$
  - 2.5. если самый первый элемент  $\mathbf{Z}(t_k)$  удовлетворяет условию 23, удалить термин  $t_k$  из  $\theta(r_i)$ , исключить из  $\mathbf{Z}$  все элементы,  $\langle d, t_k, f(d, t_k) \rangle$
  - 2.6. если в  $\mathbf{Z}(t_k)$  нет элементов, удовлетворяющих условию 23, принять запомнить граничное значение поискового веса  $w_k = 0$
  - 2.7. иначе, запомнить  $w_k = f(d, t_k) + \xi_1$ ,  $\xi_1$  - некоторая константа, исключить из  $\mathbf{Z}$  все элементы,  $\langle d, t_k, f(d, t_k) \rangle$  у которых  $f(d, t_k) < w_k$
3.  $\forall r_i \in \mathbf{R}(r)$  выполнить
  - 3.1.  $\forall d_i \in \mathbf{D}_{teach}$ 
    - 3.1.1.  $\mathbf{Z}(d_j) = \{ \langle d_j, t_k, f(d_j, t_k) \rangle : \langle d_j, t_k, f(d_j, t_k) \rangle \in \mathbf{Z} \}$

- 3.1.2. вычислить меру сходства документа  $d_j$  с рубрикой  $r_i$  по формуле 11, используя данные из  $Z(d_j)$
  - 3.1.3. запомнить вычисленное на предыдущем шаге значение  $\text{similarity}(d_j, r_i)$  в  $Y(r_i) = Y(r_i) \cup \langle d_j, \text{similarity}(d_j, r_i) \rangle$ :
  - 3.2. отсортировать  $Y(r_i)$  по убыванию  $\text{similarity}(d_j, r_i)$
  - 3.3. двигаясь последовательно по  $Y(r_i)$ , найти первый элемент, который
 
$$\langle d_{j_i}, \text{similarity}(d_{j_i}, r_i) \rangle \in Y(r_i) \wedge \langle d, r_x \rangle \in T_{\text{teach}} \wedge r_x \neq r_i \quad (24)$$
  - 3.4. если самый первый элемент  $Y(r_i)$  удовлетворяет критерию 24, семантический образ рубрики  $r_i$  построен неверно
  - 3.5. если в  $Y(r_i)$  нет элемента, удовлетворяющего критерию 24,  $l(r_i) = 0$
  - 3.6. иначе – запомнить пороговое значение меры похожести документа рубрике  $l(r_i) = \text{similarity}(d_j, r_i) + \xi_2$ ,  $\xi_2$  – некоторая константа
4. Конец алгоритма.

Как видно из алгоритма, обратная связь от полнотекстового поиска отсекает термины и даже целые семантические образы рубрик, результат классификации по которым оказался некорректным. Более того, при  $\xi_1 > 0$  и  $\xi_2 > 0$  справедливо следующее утверждение:

$$\text{Classify}(\mathbf{R}, \mathbf{H}, \text{Teach}(\mathbf{R}, \mathbf{H}, T_{\text{teach}}), \mathbf{D}_{\text{teach}}) \subset T_{\text{teach}}$$

то есть, точность классификации документов обучающей выборки равна 1.

## **Эксперимент ROMIP**

Разработанная на базе данных алгоритмов программа классификации ML Классификатор 2.0 принимала участие в сравнительном тестировании классификаторов ROMIP (цвет - *lime*). Задача, которая была поставлена перед системами классификации при тестировании, имела следующие особенности:

- классифицировались сайты, а не одиночные документы;
- обучающая выборка содержала сайты, которые были отнесены сразу к нескольким рубрикам.

Пример: сайт *dgien.narod.ru* отнесен одновременно к «Дом и семья»>Другое», «Знакомства»>Другое» и «Общество и политика»>Государство»

- обучающая выборка содержала некорректно классифицированные сайты

Некоторая ее часть (320 сайтов) была подана для оценки аксессуарам, и оказалось, что 119 из них (37%) имеют пометку *notrelevant*.

- с точки зрения системы классификации, многие страницы сайтов являются дублями, причем часто случалось так, что сайты, которым принадлежали такие страницы, относились к разным рубрикам.

Из 205545 страниц обучающей выборки только 182022 были уникальными (с точки зрения программы индексирования), причем в 11192 случаях дублирующиеся страницы принадлежали разным сайтам. В 1181 случае сайты, которым принадлежали дублирующиеся страницы, относились к разным рубрикам.

Для того, чтобы запустить в программе ML Классификатор 2.0 режим обучения по сайтам ROMIP, ее пришлось модифицировать так, чтобы выделение терминов из страниц обучающей выборки выполнялось в два этапа. На первом этапе из каждой страницы сайта выделялись слова и словосочетания, для каждого из них вычислялся вес. На втором этапе веса всех терминов, которые встречались на двух и более страницах, комбинировались, а затем из наиболее «весомых» терминов строились представления документов (сайтов) обучающей выборки.

Для каждого сайта выбирались не более 5000 терминов, причем никаких ограничений на взаимосвязи между терминами не накладывались. Пример: для сайта <http://aerofitness.narod.ru/> были выделены термины «гантели аква аэробики» и «аква аэробика».

Тестирование показало, что данный способ дает слишком большой приоритет словам и словосочетаниям, которые присутствуют на всех страницах сайта, даже если они являются элементами дизайна и навигации. Обнаружилась также, что данный вариант недостаточно эффективно учитывает изменения словосочетаний при подсчете весов. Так, например, при расчете веса словосочетания «гантели аква аэробики» никак не учитывался вес «аква аэробика». По всей видимости, следующая версия классифицирующей системы будет использовать полнотекстовый поиск не только при обучении и классификации, но и на этапе выделения терминов.

На следующем этапе было выполнено обучение с использованием алгоритмов, рассмотренных в данной статье. В результате для всех 164 рубрик были построены семантические образы. Ручной проверки и «чистки» результатов обучения не проводилось.

Прогон автоматического классификатора выполнялся дважды – один раз с исходными семантическими образами рубрик, а второй раз – с расширенными. Для каждой рубрики был вручную выбран 1 или 2 поисковых запроса, а затем они были дополнены ассоциативно связанными запросами [5]. Для всех добавленных запросов была

определена рубрика, которой соответствуют наибольшее количество найденных в обучающей выборке документов, и вычислены по-роговые значения веса.

В первом случае полнота и точность классификации были, соответственно, равны 0.06 и 0.27, а во втором - 0.08 и 0.20 (режим weak). Полный список показателей эффективности показан на следующей таблице (x/y – x – исходный набор признаков, y - расширенный):

рубрика	полнота	точность	из чего в основном состояло множество признаков рубрики
192	0.09 / 0.09	0.50 / 0.42	многословные термины ( <i>лиги чемпионата России фолейболу, чемпионата России волейболу команд</i> )
197	0.10 / 0.10	0.75 / 0.60	много специальных терминов ( <i>цигун чжоу, хаберэрцена, школы винь чунь</i> )
242	0.01 / 0.01	0.07 / 0.02	в семантические образы рубрик попало много фрагментов исходных текстов и рефератов.
168	0.06 / 0.05	0.36 / 0,27	специальные термины ( <i>прокторпус, стрюверита мальгайская республика</i> )
220	0.04 / 0.03	0.25 / 0.10	фамилии ( <i>Шитякова, Барашина, Кривицкая</i> )
263	0.04 / 0.10	0.60 / 0.19	термины по радиоэлектронике и типичные размерности ( <i>10мкФ, например</i> )
247	0.04 / 0.08	0.21 / 0.29	торговые марки стройматериалов ( <i>gasparini</i> ) и части URL одного из сайтов.
106	0.04 / 0.02	0.33 / 0.10	из-за ошибки в программе в признаки попали фрагменты JavaScript.
262	0.10 / 0.11	0.29 / 0.24	термины полиграфической тематики, размерности баннеров и названия фирм-заказчиков ( <i>ставропольжилкомхоз</i> и т.д.)
111	0	0	очень много текстов из нави-

			гационных блоков
113	0.31 / 0.39	0.70 / 0.60	названия продуктов для животных, породы собак, типичные фразы ( <i>среднего роста высота в холке</i> )
159	0	0	в признаки попали в основном фрагменты текстов песен и подписи к фотографиям
249	0	0	фрагменты календаря (20.01, 21.01 и т. д.), названия сайтов и имена файлов с графическими изображениями.
177	0	0	фрагменты анкеты
182	0.03 / 0.03	0.11 / 0.05	названия факультетов
209	0	0	в поисковые запросы попали номера рейсов, телефоны и даты
194	0.20 / 0.33	0.56 / 0.47	много одно-двухсловных терминов ( <i>иппон кумитэ, удары ногой, ояма</i> )

Во всех рубриках семантические образы содержали некоторое количество «мусора» - фрагментов навигации (например, *ни вт ср чт пт сб вс*), имен файлов и частей URL. В некоторых рубриках таких терминов было большинство. В остальных термины получше, по ним можно было без труда догадаться о названии рубрики и содержании приписанных к ним документов.

Как видно из таблицы, добавление ассоциативных терминов в автоматическом режиме увеличило полноту ценой потери точности.

Результаты обучения часто искажались за счет гостевых книг, форумов и сборников анекдотов. Так, например, одним из самых характерных слов для сайтов рубрики *мотоциклы* были *водкинг* и *пивинг*, что, конечно, понятно с человеческой точки зрения, но не подходит для точной классификации.

## **Заключение**

По результатам тестирования можно сделать следующие выводы:

- 1) Нынешняя версия системы ML Классификатор 2.0 может быть использована для классификации сайтов только при условии ручной коррекции семантических образов рубрик. Коррекция подразумевает удаление ошибочно выбранных поисковых за-

просов и добавление в признаки рубрик дополнительных запросов.

- 2) Для того, чтобы приспособить систему для обучения и классификации в автоматическом режиме, необходимо внести в нее следующие изменения:
  - необходимо выявлять на классифицируемых сайтах фрагменты, которые являются элементами дизайна и навигации. Такие фрагменты нужно исключать из рассмотрения или обрабатывать в особом режиме;
  - дублирующиеся страницы сайтов обучающей выборки должны выявляться и исключаться из рассмотрения до начала обучения;
  - при оценке соответствия сайта рубрике аксессоры начинают просмотр со стартовой страницы. Таким образом, сайт приписывается в основном к рубрикам, которым соответствует стартовая страница, а также те, на которые она ссылается. Для того, чтобы результат автоматической классификации был ближе к таким результатам, имеет смысл внести в систему учет расстояния (в гиперссылках) от корневой страницы сайта до каждой из анализируемых страниц.

## *Литература*

- [1] *Березкин Д.В., Шабанов В.И., Андреев А.М.* Методы выделения терминов из текста. // Современные информационные технологии. Межвузовская юбилейная научно-техническая конференция аспирантов и студентов. - Изд-во МГТУ им Н. Э. Баумана, - 2001. - с. 117-127.
- [2] *Солтон Дж.* Динамические библиотечно-информационные системы. - М.: Мир, - 1979. - 550 с.
- [3] *Солтон Дж.* Автоматическая обработка, хранение и поиск информации. - М.: Советское радио. - 1973. - 560 с.
- [4] *Харин Н. П.* Метод ранжирования выдачи, учитывающий автоматически построенные ассоциативные отношения между терминами // НТИ. Сер. 2. – 1990. - №9, с. 19-23.
- [5] *Шабанов В.И., Власова А.Е.* Алгоритм формирования ассоциативных связей и его применение в поисковых системах. // Труды международной конференции Диалог`2003 по компьютерной лингвистике и интеллектуальным технологиям. – 2003. - с. 603-609
- [6] *C. Chekuri, M.H. Goldwasser;* Web Search Using Automatic Classification. // Proceedings of WWW-96. – 1996.

- [7] *V. Dasigi, R. Manu*. Neural Net Learning Issues in Classification of Free Text Documents. // AAAI spring symposium on Machine Learning in Information Access – 1996.
- [8] *J. Fuernkranz*. A study using n-gram features for Text Categorization // Tech report OEFAl-TR-98-30 – 1998.
- [9] *A. McCallum, R. Rosenfeld, T. Mitchell, A. Y. Ng*. Improving text classification by shrinkage in a hierarchy of classes. // In *Proc. ICML-98*. – 1997 – c. 359-367
- [10] *D. Koller, M. Sahami*. Hierarchically classifying documents using very few words. // In *Proc. ICML-97*. – 1997 – c. 170-176
- [11] *A. McCallum, K. Nigam*. A comparison of Event Models for Naïve Bayes Classification; // In AAAI-98 Workshop on Learning for Text Categorization. – 1998 – 8 c.
- [12] *R. Papka, J. Allan*. Document classification using Multiword features // Proceedings of the Conference in Information and Knowledge Management.- 1998 – 8 c.