Обучение классификаторов на основе выделения фрагментов

Classifier training by passage recognition

Васильев В. Г. (vvg_2000@mail.ru)
ЛАН-ПРОЕКТ

В данной работе рассматривается новый метод обучения классификаторов, основанный на использовании результатов автоматического выделения фрагментов в текстах. Также приводится описание известных и нового метода выделения фрагментов в текстах. Проводятся эксперименты со стандартными тестовыми массивами, как на русском, так и на английском языке.

1. Введение

В настоящее время вопросы построения эффективных средств автоматической классификации текстов достаточно активно рассматриваются как в отечественных, так и в зарубежных работах. При этом в последнее время наибольшее распространение для построения классификаторов получил подход, основанный на использовании статистических или обучаемых методов. Данные методы опираются на использование эталонных массивов документов, в которых документы вручную разложены экспертами по заданным рубрикам, для автоматического построения статистических моделей и правил классификации.

Необходимо отметить, что в большинстве работ в области обучаемой классификации документы рассматриваются как неделимые цельные сущности и представляются в виде вектора весов информационных признаков. При таком подходе внутренняя структура документов и последовательность следования слов не учитывается, что является приемлемым при обработке относительно небольших или тематически однородных документов. В случае, когда документы являются политематическими или имеют сложную внутреннюю структуру, данный подход не подходит. В этой ситуации в ряде работ предлагается использовать подходы, основанные на поиске и классификации фрагментов в текстах. Рассмотрим их более подробно методы поиска фрагментов и методы классификации фрагментов, которые описываются в современной литературе.

Задача поиска фрагментов заключается в извлечении частей документа, которые релевантны запросу или информационной потребности пользователя [9]. Данная задача может решаться как самостоятельная задача, так и как вспомогательная задача при поиске документов, построении рефератов и поиске ответов на вопросы. Для ее решения разработан широкий набор подходов [1, 8, 9]: на основе вычисления TF-IDF весов у фрагментов, на основе вычисления функции правдоподобия запроса, на основе построения вероятностных моделей для оценки релевантности запроса, на основе использования обучаемых классификаторов, на основе использования скрытой марковской модели, а также на основе комбинировании нескольких подходов.

Задача классификации фрагментов заключается в извлечении фрагментов документа, которые соответствуют одной или нескольким заранее заданным рубрикам [2]. Данная задача несколько отличается от задачи поиска фрагментов, так как предполагает, что рубрики по которым осуществляется набор фрагментов заранее заданы. Для ее решения также разработан набор специализированных методов. Давайте рассмотрим наиболее типичные из них.

В работе [1] авторы предлагают использовать трехэтапный подход к распознаванию фрагментов. На первом этапе производится обучение классификатора с использованием обучающего множества, состоящего из полных текстов. На втором этапе производится разбиение текстов на фрагменты с использованием различных подходов. На третьем этапе осуществляется классификации всех фрагментов с использование классификатора обученного на первом шаге.

В работе [2] авторы используют двух этапную технологию. На первом этапе производится отнесение документа целиком к одной из рубрик с исполь-

зованием простейшего Байесовского классификатора. На втором этапе в тексте документа находится фрагмент, который наиболее соответствует соответствующей рубрике, путем вычисления максимума функции правдоподобия.

В работе [6] предлагается использовать скрытую марковскую модель. Выделение фрагментов происходит в два этапа. На первом этапе для каждой рубрики оцениваются параметры скрытой марковской модели с использованием обучающего множества, состоящего из целых документов. На втором этапе построенная модель используется для выделения фрагментов.

В ряде случаев классификация фрагментов является вспомогательной задачей для повышения качества классификации документов. В работе [10] авторы предлагают выполнять независимую классификацию фрагментов текста, а затем объединять полученные результаты для получения итоговой классификации документов целиком.

Наконец, в работе [7] фрагменты явным образом не выделяются, а используется специальный метод для вычисления весов терминов, который учитывает расположение терминов в тексте. В частности, вектор документа получается как взвешенная сумма векторов предложений.

Необходимо отметить, что во всех рассмотренных выше подходах обучение классификаторов производится на целых документах, т. е. на этапе обучения внутренняя структура документов не учитывается.

В данной работе рассматривается новый итерационный подход к обучению классификаторов, который основан на использовании результатов выделения фрагментов в текстах для итерационного переобучения классификатора. Для этих целей предлагается использовать следующую технологию.

На первом шаге, для каждой рубрики обучается бинарный классификатор с использованием обучающего множества, состоящего из полных документов.

На втором шаге, для каждой рубрики в каждом документе выделяются релевантные фрагменты с использованием построенных классификаторов.

На третьем шаге, для каждой рубрики формируется специальное обучающее множество, которое состоит из фрагментов текстов, выделенных для данной рубрики на предыдущем шаге.

На четвертом шаге, осуществляется проверка критерия завершения работы и либо завершается обучение классификатора, либо повторяются шаги два и три.

Работа имеет следующую структуру. В разделе 2 рассматриваются базовые методы, которые используются для обучения классификаторов. В разделе 3 описываются методы выделения фрагментов. В разделе 5 дается описание новых алгоритмов обучения классификаторов и классификации текстов, с использованием информации о фрагментах. В разделе 5 приводятся результаты экспериментов с различными тестовыми массивами.

2. Используемые методы классификации

Для классификации текстов в настоящей работе используются два базовых метода: метод на основе машин опорных векторов (SVM) и метод на основе байесовского классификатора, в котором в качестве вероятностного распределения для отдельных рубрик используется распределение фон Мизеса-Фишера (VMF). Приведем необходимые определения.

Пусть $\Omega=\{\omega_1,...,\omega_k\}$ — множество из k рубрик и D — множество всех документов. В соответствии с работой [11] задачу классификации текстов будем рассматривать как k независимых задач, т. е. для каждой рубрики ω_j строится решающая функция вида

$$H_j(d) = \begin{cases} 1, & d \in \omega_j, \\ 0, & d \notin \omega_j. \end{cases}$$

Также будем считать, что каждый документ $d \in D$ представляется в виде вектора весов $x=(x_1,...,x_m)^T$, где x_l — вес признака $l=1,...,m$, m — размерность пространства признаков. При проведении экспериментов в данной работе в качестве признаков выступают как отдельные слова, так и словосочетания.

2.1. Метод машин опорных векторов

В методе SVM решающая функция $H_j(d)$, $j=1,...,k$, является линейной функцией следующего вида

$$H_j(d) = H_j(x) = \begin{cases} 1, & R_j(x) > 0, \\ 0, & R_j(x) < 0, \end{cases}$$

где $R_j(x)=w_j^T x+w_{0j}$, $R_j(x)$ — дискриминантная функция, $w_j=(w_{j1},...,w_{jm})^T$ — вектор весов и w_{j0} — смещение, которые определяют гиперплоскость разделяющую документы на классы.

Заметим, что $R_j(x)$ можно рассматривать как меру соответствия документа рубрике ω_j, так как $R_j(x)/\|w_j\|_2$ — расстояние от вектора x до разделяющей гиперплоскости.

Пусть $(x_1,y_1),...,(x_n,y_n)$ — обучающее множество для рубрики ω_j, $j=1,...,k$, где x_i — вектор признаков и $y_i \in \{-1,1\}$ — вектор идентификатор рубрик для документов d_i, $i=1,...,n$. Если документ $d_i \in \omega_j$, тогда $y_i=1$, в противном случае $y_i=-1$.

Параметры разделяющей гиперплоскости w_j и w_{j0} находятся путем максимизации расстояния от разделяющей гиперплоскости положительных и отрицательных примеров, в частности, это делается путем решения следующей оптимизационной задачи

$$\min_{w,w_0} w_j^T w_j + C\sum_{i=1}^{n}\xi_i,$$

$$\begin{cases} y_i\left(w_j^T x_i + w_{j0}\right) \geq 1-\xi_i, i=1,\dots,n, \\ \xi_i \geq 0, i=1,\dots,n, \end{cases}$$

где ξ_i, $i=1,...,n$, вспомогательные переменные, C — положительный параметр.

2.2. Байесовский метод

При использовании байесовского подхода решающая функция H_j для рубрики ω_j, $j=1,...,k$, имеет следующий вид

$$H_j(x) = \begin{cases} 1, & \dfrac{p_j f\left(x|\omega_j\right)}{\overline{p}_j f\left(x|\overline{\omega}_j\right)} > 1, \\ 0, & \dfrac{p_j f\left(x|\omega_j\right)}{\overline{p}_j f\left(x|\overline{\omega}_j\right)} \leq 1, \end{cases}$$

где $p_j=p(\omega_j)$ — вероятность рубрики ω_j, $f(x|\omega_j)$ — функции плотности для рубрики ω_j, $j=1,...,k$ $\overline{\omega}_j$ — обозначает множество документов, которые не относятся к рубрике ω_j, т. е. множество отрицательных примеров.

В данной работе в качестве $f(x \mid \omega_j)$ используется функция плотности распределения фон Мизеса-Фишера, которая определяется следующим образом

$$f\left(x\middle|\omega_j\right) = f\left(x|\mu_j,\kappa_j\right) = c_m\left(\kappa_j\right)e^{\kappa_j \mu_j^T x},$$

где $x \in R^m$, $\|x\|_2=1$, $\mu_j \in R^m$ среднее направление, $\|\mu_j\|_2=1$, $\kappa_j \geq 0$ мера концентрации, $m \geq 2$, $c_m(\kappa)$ нормализующий множитель.

Пусть теперь $(x_1,y_1),...,(x_n,y_n)$ — обучающее множество текстов для рубрики ω_j, $j=1,...,k$, где $x_i \in R^m$ — вектор признаков и $y_i \in \{0,1\}$ — идентификатор рубрики для документа d_i, $i=1,...,n$. При этом, если документ d_i относится к классу ω_j, то $y_i=1$, в противном случае $y_i=0$.

Для построения классификатора $H_j(x)$ для рубрики ω_j, $j=1,...k$, необходимо найти оценки параметров модели фон Мизеса-Фишера p_j, μ_j, κ_j и $\overline{p}_j, \overline{\mu}_j, \overline{\kappa}_j$. В работе [12] показывается, что оценки максимального правдоподобия для данных параметров имеют следующий вид

$$p_j^* = \frac{1}{n}\sum_{i=1}^{n} y_i, \overline{p}_j^* = \frac{1}{n}\sum_{i=1}^{n}(1-y_i),$$

$$\mu_j^* = \frac{r_j^*}{\left\|r_j^*\right\|_2}, \overline{\mu}_j^* = \frac{\overline{r}_j^*}{\left\|\overline{r}_j^*\right\|_2},$$

$$\kappa_j^* \approx \frac{r_j^* m - \left(r_j^*\right)^3}{1-\left(r_j^*\right)^2}, \overline{\kappa}_j^* \approx \frac{\overline{r}_j^* m - \left(\overline{r}_j^*\right)^3}{1-\left(\overline{r}_j^*\right)^2},$$

где $r_j^* = \sum_{i=1}^{n} y_i x_i, \overline{r}_j^* = \sum_{i=1}^{n}(1-y_i)x_i.$

Предварительные эксперименты с различными массивами текстов показали, что в большинстве случаем достаточно использовать фиксированные значения $\kappa_j^*=\overline{\kappa}_j^*=10$. В данном случае решающее правило для рубрики можно упростить и представить в следующем виде

$$H_j(x) = \begin{cases} 1, & R_j(x) > 0, \\ 0, & R_j(x) \leq 0, \end{cases}$$

где $R_j(x) = \kappa_0^{-1} log\frac{p_j}{\overline{p}_j} + \left(\mu_j - \overline{\mu}_j\right)^T x,$

$R_j(x)$ — дискриминантная функция, $\kappa_0=10$, $j=1,...,k$. Заметим, что чем больше значение $R_j(x)$, то тем ближе вектор x центру рубрики ω_j. Следовательно, $R_j(x)$ можно использовать в качестве меры соответствия документа к рубрике ω_j.

3. Выделение фрагментов

Рассмотрим теперь подходы к выделению фрагментов в текстах. Далее будем считать, что фрагмент это набор предложений и что на входе процедуры выделения фрагментов имеется документ, который $d \in D$ представляется с помощью матрицы $X \in R^{(m \times s)}$ и набор независимых классификаторов для каждой рубрики ω_j, $j=1,...,k$, где $X=(x_1,...,x_s)$, $x_t \in R^m$ — вектор весов признаков для предложения $t=1,...,s$, s — число предложений в документе d и m — общее число различных признаков во всех документах.

На выходе процедуры необходимо получить матрицу Λ, которая содержит веса предложений для каждой рубрики ω_j, $j=1,...,k$, где $\Lambda \in R^{(k \times s)}$, $\Lambda=(\lambda_1,...,\lambda_s)$, $\lambda_t=(\lambda_1 t,...,\lambda_k t)^T$, $\lambda_j t$ — вес предложения $t=1,...,s$ для рубрики ω_j, $j=1,...,k$.

Для выделения фрагментов в настоящей работе используются следующие четыре подхода: выделение фрагментов путем классификации предложений, выделении фрагментов путем классификации блоков текста, выделение фрагментов путем классификации иерархического покрытия и выделение фрагментов путем решения специальной оптимизационной задачи. Рассмотрим указанные подходы более подробно.

3.1. Выделение фрагментов путем классификации предложений

Данный подход является наиболее простым и используется во многих работах [1,2,10]. Вес $\lambda_j t$ предложения $t=1,...,s$ для рубрики ω_j, $j=1,...,k$ вычисляется следующим образом

$$\lambda_{jt} = R_j\left(\frac{x_t}{\|x_t\|_2}\right),$$

где $R_j(x)$ — дискриминантная функция для рубрики ω_j. Необходимо отметить, что перед вычислением веса предложения производится нормировка соответствующего вектора признаков таким образом, чтобы его длина была равна 1. Это необходимо из-за того, что обучение классификатора производится с использованием нормированных векторов.

3.2. Выделение фрагментов путем классификации блоков текста

В данном подходе выделение фрагментов производится в два этапа.

На первом этапе текст разбивается на набор смежных блоков предложений с использованием метода TextTiling [3]. При проведении экспериментов использовались следующие параметры данного алгоритма: размер блока равен 4 предложениям, похожесть блоков вычисляется с использованием косинусной меры близости, сглаживание осуществляется с использованием алгоритма скользящего среднего, в котором размер окна равен 3. В результате получается вектор $b=(b_1,...,b_s)$, где $b_t \in \{1,B\}$ номер блока для предложения $t=1,...,s$, B — общее число блоков в тексте.

На втором этапе вычисляются веса предложений $\lambda_j t$, $t=1,...,s$, для рубрик ω_j, $j=1,...,k$ путем использования следующей формулы

$$\lambda_{jt} = u_{jb_t},$$

где u_{ji} — мера соответствия блока $i=1,...,B$ рубрике ω_j, $j=1,...,k$

$$u_{ji} = R_j\left(\frac{\sum_{t\in\{l=1,...,s|b_l=i\}} x_t}{\left\|\sum_{t\in\{l=1,...,s|b_l=i\}} x_t\right\|_2}\right).$$

Необходимо отметить, что также можно использовать и другие методы разделения текстов на блоки [3,4,5,6].

3.3. Выделение фрагментов путем классификации иерархического покрытия

В данном подходе вес предложения $\lambda_j t$, $t=1,...,s$, для рубрики ω_j, $j=1,...,k$, вычисляется путем нахождения суммы весов всех непрерывных фрагментов содержащих данное предложение. В частности, $\lambda_j t$ находится следующим образом

$$\lambda_{jt} = \sum_{y\in F, x_t\in y} R_j\left(\frac{y}{\|y\|_2}\right),$$

где F множество векторов признаков, соответствующих всем непрерывным фрагментам предложений в данном документе, т. е.

$$F = \left\{\sum_{t=l_1}^{l_2} x_t \;|1 \le l_1 \le l_2 \le s\right\}.$$

Заметим, что для вычисления весов для всех предложений для одной категории необходимо вычислить степень соответствия всех фрагментов данной рубрике, что требуется порядка $O(s^2)$ операций.

Для снижения вычислительной сложности можно воспользоваться иерархическим множеством фрагментов H вместо использования всего множества фрагментов F, где

$$H = H_0 \cup \left(\bigcup_{i=1}^{\lceil \log_2 s\rceil} H_i\right),$$

$$H_t = \left\{\sum_{t=1+l2^{i-1}}^{\min(l2^{i-1}+2^i,s)} x_t \;|l = 0, \dots, \left\lceil \frac{s}{2^{i-1}} - 1\right\rceil\right\},$$

$$H_0 = \{x_1,...,x_s\}.$$

Можно показать, что

$$|H| = |H_0| + \sum_{i=1}^{\lceil \log_2 s\rceil} |H_i| \le \log_2 s + 1 + 3s.$$

Таким образом, в данном случае вычислительная сложность порядка $O(s)$. Также можно показать, что для любого фрагмента $f \in F$ существует фрагмент $h \in H$ такой, что

$$\frac{|h\Delta f|}{|f|} \le \frac{1}{2},$$

где $|f|$ — число предложений во фрагменте f, $|f\Delta h|$ число различных предложений во фрагментах f и h. Иными словами, для любого фрагмента $f \in F$ существует фрагмент $h \in H$, который содержит, по крайней мере, половину общих предложений. В результате, выражение для вычисления весов предложений приобретает следующий вид

$$\lambda_{jt} = \sum_{h\in H, x_t\in h} R_j\left(\frac{h}{\|h\|_2}\right).$$

3.4. Выделения фрагментов с использованием оптимизационных методов

В данном новом подходе веса $\lambda_j=(\lambda_j 1,...\lambda_j s)$ предложений для рубрики ω_j, $j=1,...,k$ находятся путем решения следующей оптимизационной задачи

$$\max_{\lambda_j} R_j\big(z(\lambda_j)\big),$$

где $z(\lambda_j)$ — вектор документа, основанный на векторах предложений, который определяется следующим образом

$$z(\lambda_j) = \frac{\sum_{t=1}^{s} \lambda_{jt} x_t}{\left\| \sum_{t=1}^{s} \lambda_{jt} x_t \right\|_2}.$$

Заметим, что из определения следует, что $\|z(\lambda_j)\|_2=1$.

Рассмотрим теперь процедуру нахождения приближенного решения данной задачи для случая, когда в качестве решающей функции используется классификатор на основе машин опорных векторов. В данном случае оптимизационная задача имеет следующий вид

$$\max_{\lambda_j} w_j^T z(\lambda_j) + w_{0j}.$$

Прямой нахождение решения данной оптимизационной задачи является достаточно сложным. По этой причине рассмотрим следующую упрощенную задачу

$$\max_{\|\xi\|_2=1} w_j^T \xi + w_{0j}.$$

Несложно показать, что в данном случае максимум достигается, когда $\xi=w_j/\|w_j\|_2$.

Таким образом, можно найти приближенное решение исходной задачи путем нахождения решения следующего уравнения

$$z(\lambda_j)= \xi,$$

которое можно переписать следующим образом

$$\frac{\sum_{t=1}^{s} \lambda_{jt} x_t}{\left\| \sum_{t=1}^{s} \lambda_{jt} x_t \right\|_2} = \frac{w_j}{\|w_j\|_2}.$$

Последнее равенство эквивалентно следующему равенству

$$\sum_{t=1}^{s} \lambda_{jt} x_t = w_j.$$

В матричной форме оно имеет следующий вид

$$X\lambda_j^T = w_j.$$

Необходимо отметить, что матрица X имеет m строк и s столбцов, где m — число различных признаков и s число предложений. В большинстве случаев $m \gg s$. Более того, в большинстве случаев матрица X является сильно разряженной. По этой причине приведенное выше равенство обычно не имеет точного решения и для нахождения решения необходимо пользоваться приближенными методами, ориентированными на работу с большими разряженными матрицами.

В данной работе для этих целей используется алгоритм LSQR [13], который как раз ориентирован для работы с такими данными. При его использовании решение уравнения находится путем итерационного решения задачи следующей задачи наименьших квадратов

$$\min_{\lambda_j} \left\| X\lambda_j^T - w_j \right\|_2^2.$$

Пусть $\lambda_j^*=(\lambda_{j1}^*,...,\lambda_{js}^*)$ соответствующее решение последней задачи. Тогда, если $\lambda_{jt}^*>0$ можно считать, что предложение $t=1,...,s$ соответствует «положительной» рубрике ω_j. По этой причине в качестве предложений соответствующих рубрике можно рассматривать те, которые соответствуют положительным элементам вектора λ_j^*.

Рассмотрим теперь выделение значимых предложений в том случае, когда используется байесовский классификатор на основе распределения фон Мизеса-Фишера. В данном случае исходная оптимизационная задача имеет следующий вид

$$\max_{\lambda_j} \left(\mu_j - \overline{\mu}_j\right)^T z(\lambda_j) + \kappa_0^{-1} log \frac{p_j}{\overline{p}_j}.$$

Если обозначить $w_j^T = \left(\mu_j - \overline{\mu}_j\right)^T$ и

$w_{j0} = \kappa_0^{-1} log \frac{p_j}{\overline{p}_j}$, тогда исходная оптимизационная задача будет полностью совпадать с исходной оптимизационной задачей для SVM классификатора. Следовательно, решение ее можно находить аналогичным образом.

4. Обучение классификаторов

Рассмотрим теперь алгоритмы обучения классификаторов и классификации документов на основе фрагментов.

4.1. Обучение классификаторов с использованием фрагментов

Пусть $\Omega=\{\omega_1,...,\omega_k\}$ множество из k заранее заданных рубрик и $d_1,...,d_n$ обучающее множество документов. Каждый документ d_i, $i=1,...,n$, в обучающем множестве представляется с помощью пары $\langle X_i,c_i\rangle$, где $X_i=(x_{1i},...,x_{(si\,i)})$ — матрица, столбцы которой являются векторами весов информационных признаков для предложений документа, s_i — число предложений в документе и $c_i=(c_{1i},...,c_{ki})^T$ вектор с идентификаторами рубрик, причем $c_{ji}\in\{0,1\}$ признак отнесения документа d_i к рубрике ω_j экспертом.

Пусть $\Lambda_i=(\lambda_{jti})_{k\times si}$, $i=1,...,n$ множество матриц таких, что λ_{jti} вес предложения $t=1,...,s_i$ для рубрики $j=1,...,k$ в документе с номером $i=1,...,n$. Тогда алгоритм обучения классификатора принимает следующий вид.

1. Положить $iter=1$, $\lambda_{jti}=1$ for $i=1,...,n$, $j=1,...,k$, $t=1,...,s_i$.
2. Вычислить вектора информационных признаков z_{ij} для каждого документа d_i, $i=1,...,n$, и рубрики ω_j, $j=1,...,k$, путем использования следующей формулы

$$z_{ij}=\frac{\sum_{t=1}^{s_i}\lambda_{jti}x_{ti}}{\left\|\sum_{t=1}^{s_i}\lambda_{jti}x_{ti}\right\|_2}$$

3. Построить для каждой рубрики ω_j, $j=1,...,k$, бинарный классификатор H_j путем использования векторов документов $z_{1j},...,z_{nj}$ и эталонных индикаторов принадлежности документов к данной рубрике $c_{j1},...,c_{jn}$.
4. Для каждой рубрики ω_j, $j=1,...,k$, выделить фрагменты во всех документах и перевычислить значения весов Λ_i, $i=1,...,n$ путем использования одного из методов выделения фрагментов и классификаторов H_j, обученных на предыдущем шаге.
5. Если $iter$ меньше заданного порога, то перейти к шагу 2, в противном случае завершить работу алгоритма.

4.2. Классификация текстов с использованием фрагментов

Теперь давайте рассмотрим алгоритм классификации документов, который применяется совместно с описанным алгоритмом обучения классификаторов.

Пусть имеется документ $d\in D$, который представлен с помощью матрицы $X\in R^{m\times s}$ и классификаторы H_j для каждой рубрики ω_j, $j=1,...,k$, где $X=(x_1,...,x_s)$, $x_t\in R^m$ — вектор весов информационных признаков для предложения $t=1,...,s$, s число предложений в документе d и m число различных информационных признаков во всех документах D.

Пусть $\Lambda\in R^{k\times s}$ обозначает матрицу весов предложений для рубрик ω_j, $j=1,...,k$, где λ_{jt} вес предложения $t=1,...,s$ для рубрики ω_j, $j=1,...,k$. Тогда алгоритм классификации текстов имеет следующий вид.

1. Для каждой рубрики ω_j, $j=1,...,k$, выделить соответствующие ей фрагменты в документе и вычислить веса Λ путем использования одного из подходов к выделению фрагментов и классификатора H_j.
2. Для каждой рубрики ω_j, $j=1,...,k$, вычислить вектора весов документов $z_j\in R^m$ путем использования следующего выражения

$$z_j=\frac{\sum_{t=1}^{s}\lambda_{jt}x_t}{\left\|\sum_{t=1}^{s}\lambda_{jt}x_t\right\|_2}.$$

3. Для каждой рубрики ω_j, $j=1,...,k$, выполнить классификацию векторов документов z_j путем использования классификатора H_j.

Таким образом, в приведенном алгоритме классификации для каждой рубрики строится специальное представление документов, что позволяет адаптироваться к особенностям каждой категории при классификации документов.

5. Эксперименты

5.1. Тестовые массивы текстов

Рассмотрим теперь результаты предварительных экспериментов по оценке качества разработанных методов классификации документов.

При проведении экспериментов использовались два английских массива текстов “Reuters-21578” (http://www.daviddlewis.com/resources/testcollections/reuters21578/) и “20 News Groups” (http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html), и один русский массив текстов “ROMIP-2004” (http://romip.ru/en/index.html). В Таблице 1 приведены основные характеристики данных массивов.

Таблица 1. Основные характеристики массивов текстов

Массив	Число документов	Число рубрик	Размер массива
Reuters-21578	13476	123	16.2 Mb
20 News Groups	16330	20	46 Mb
ROMIP 2004	6931	173	281 Mb

Также при проведении экспериментов использовались сокращенные варианты данных

массивов: “Reuters-21578-10”, “20 News Groups Mini”, “ROMIP 2004 Mini”. Данные сокращенные массивы более подходят для интенсивных экспериментов ввиду меньшего размера и более сбалансированной структуры. Более того данные массивы значительно более часто используются в работах других исследователей, чем исходные полные массивы [1,2].

Массив “Reuters-21578-10” состоит из десяти наибольших рубрик “acq”, “corn”, “crude”, “earn”, “grain”, “interset”, “money-fx”, “ship”, “trade”, “wheat” из массива “Reuters-21578”.

Массив “20 News Groups Mini” является подмножеством массива “20 News Groups” и содержит 2000 случайным образом отобранных документов. В каждой рубрике при этом ровно по 100 документов.

Массив “ROMIP-2004 Mini” является подмножеством массива “ROMIP 2004”. Он включает десять наибольших рубрик со следующими номерами: 9001149, 9001227, 9001355, 9001423, 9001557, 9001613, 9001755, 9001759, 9001716900, 901800651. В таблице 2 приведены основные характеристики данных массивов.

Таблица 2. Основные характеристики сокращенных тестовых массивов текстов

Массив	Число документов	Число рубрик	Размер
Reuters-21578-10	8592	10	8.9 Mb
20 News Groups Mini	1954	20	4.7 Mb
ROMIP 2004 Mini	1704	10	113 Мб

Для оценки качества работы классификаторов в настоящей работе использовались стандартные показатели: точность, полнота, F-мера. При вычислении данных показателей для всего массива в целом использовалось макроусреднение. Вычисление значений показателей производилось в соответствии со следующей схемой.

1. Разбиение эталонного массива текстов на обучающее и тестовое множество случайным образом в заданной пропорции.
2. Обучение классификатора на обучающем множестве с использованием различных методов.
3. Классификация документов из тестового множества с использованием построенных классификаторов и оценка качества классификации.

Необходимо отметить, что все алгоритмы классификации и обработки текстов были реализованы на языке MATLAB в виде специальных функций и классов.

Далее при описании экспериментов будут использоваться следующие сокращенные обозначения: SVM — стандартный классификатор SVM, который обучен на полных документах; VMF — стандартный байесовский классификатор на основе распределения фон Мизеса-Фишера, который обучен на полных документах; SVM-SENT — классификатор SVM, обученный на фрагментах, выделенных с помощью классификации предложений; SVM-TILE — классификатор SVM, обученный на фрагментах, выделенных путем классификации блоков предложений; SVM-HIER — классификатор SVM, обученный на фрагментах, выделенных путем классификации иерархического покрытия документа; SVM-LS — классификатор SVM, обученный на фрагментах, выделенных путем решения специальной оптимизационной задачи; VMF-SENT — классификатор VMF, обученный на фрагментах, выделенных с помощью классификации предложений; VMF-TILE — классификатор VMF, обученный на фрагментах, выделенных путем классификации блоков предложений; VMF-HIER — классификатор VMF, обученный на фрагментах, выделенных путем классификации иерархического покрытия документа; VMF-LS — классификатор , обученный на фрагментах, выделенных путем решения специальной оптимизационной задачи.

5.2. Результаты экспериментов

Рассмотрим теперь результаты предварительных экспериментов с сокращенными массивами текстов. В данных экспериментах исходное эталонное множество документов разделялось на обучающее и тестовое множества в пропорции 75% на 25%. В таблицах 3 и 4 приведены показатели качества классификации для методов SVM и SVM-LS для массивов “20 News Groups Mini” и “ROMIP 2004 Mini”, соответственно.

Из приведенных таблиц можно заметить, что качество работы метода SVM-LS значительно выше качества работы стандартного метода SVM.

Анализ результатов обработки массива 20 News Groups показывает, что повышение качества классификации возможно связано с тем, что оригинальные документы содержат служебные заголовки почтовых сообщений, которые являются малоинформативными, а также рассуждения участников новостных групп на отвлеченные темы, которые не относятся напрямую к основной теме новостной группы.

Анализ результатов обработки массива ROMIP 2004 Mini показывает, что повышение качества классификации, скорее всего, связано с политематическим содержанием нормативно-правовых документов. Также необходимо отметить, что достигнутые показатели качества классификации не уступают тем, которые получены другими авторами в рамках семинара РОМИП (в данных работах F-мера принимает значения от 0,1 до 0,5 в зависимости от состава используемых рубрик).

Таблица 3. Качество работы классификаторов для массива 20 News Groups Mini

Method	Precision	Recall	F-measure
SVM	0.94	0.27	0.40
SVM-LS	0.96	0.89	0.92

Таблица 4. Качество работы классификаторов для массива ROMIP 2004 Mini

Method	Precision	Recall	F-measure
SVM	0.67	0.29	0.36
SVM-LS	0.76	0.39	0.50

Анализ качества работы всех рассмотренных алгоритмов на массиве ROMIP 2004 Mini приведен в таблице 5, из которой можно заметить, что в наибольшей степени повышение качества классификации происходит при использовании алгоритма SVM.

Таблица 5. Качество работы классификаторов для массива ROMIP 2004 Mini

Method	F-measure	Method	F-measure
SVM	0.37	VMF	0.45
SVM-SENT	0.39	VMF-SENT	0.47
SVM-HIER	0.38	VMF-HIER	0.46
SVM-TILE	0.39	VMF-TILE	0.46
SVM-LS	0.50	VMF-LS	0.37

Предварительные эксперименты с другими массивами также показали, что повышение качества классификации в основном происходит при относительно небольшом размере обучающего множества. Для проверки данной гипотезы была проведена серия экспериментов, в которых в качестве параметра выступал размер обучающего множества. В частности, на рисунках 1 и 2 приведены зависимости качества классификации от размера обучающего множества для различных алгоритмов на массивах “20 News Groups” и “Reuters-21578”.

Проведенные эксперименты подтвердили гипотезу о том, что качество классификации более сильно увеличивается при маленьких объемах обучающих выборок. Также было установлено, что наиболее эффективным является метод SVM-LS.

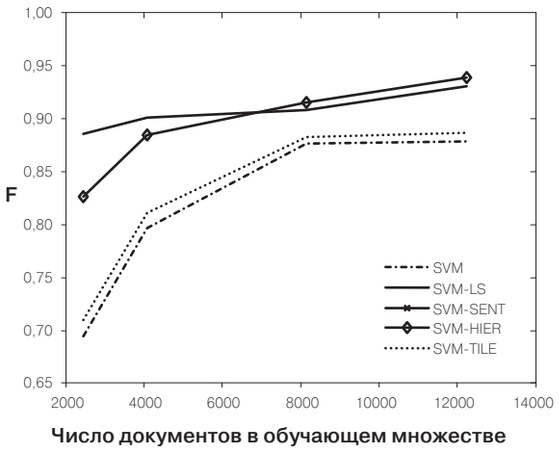

Рис. 1. Качество работы SVM классификаторов на массиве 20NG

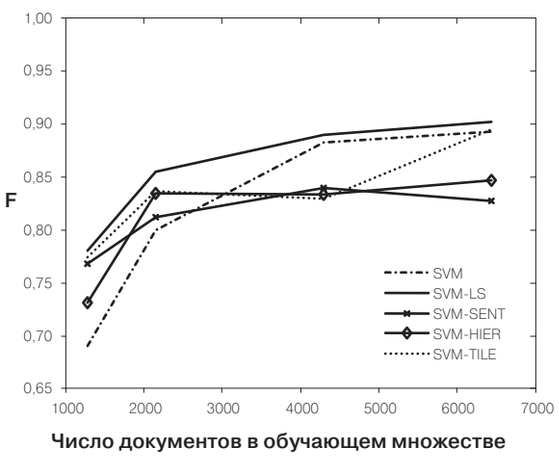

Рис. 2. Качество работы SVM классификаторов на массиве Reuters-21578-10

6. ВЫВОДЫ

Таким образом, в настоящей работе предложен новый подход к обучению классификаторов, основанный на использовании результатов автоматического выделения фрагментов в текстах. Экспериментально показано, что в ряде случаев данный подход может быть достаточно эффективным значительно повышать качество классификации. Также в работе рассмотрен новый метод выделения фрагментов, основанный на решении специальной оптимизационной задачи и нахождении весов предложений в тексте.

Литература

1. *Mengle S., Goharian N.* Passage detection Using Text Classification. Journal of the American Society for Information Science and Technology, 60(4), 2009, pp. 814–825.
2. *Murtagh Y. B., McClean S., Anderson T.* Text Passage classification using supervised Learning, 1999. — 13 p. (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.130.4741)
3. *Hearst M.* TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. Computational Linguistics , 23 (1), 1997. pp. 33–64.
4. *Choi F., Wiemer-Hasting P., Moore J.* Latent semantic Analysis for Text Segmentation. Proceedings of NAACL'01, Pittsburgh, PA, 2001. — pp. 109–117.
5. *Blei D., Moreno P. J.* Topic Segmentation with an Aspect Hidden Markov Model. SIGIR'01, September 9–12, 2001, New Orleans, Louisiana, USA. — 6 p.
6. *Denoyer L., Zaragoza H., Gallinary P.* HMM-based Passage Models for Document Classification and Ranking. 23-s BCS European Annual Colloquium on Information Retrieval, 2001.
7. *Ko Y., Park J., Seo J.* Improving text categorization using the importance of sentences. Information Processing and Management 40, 2004. — pp. 65–79.
8. *Extraction* of Coherent Relevant Passages Using Hidden Markov Model. ACM Transactions on Information Systems, Vol. 24, No. 3, 2006. pp. 295–319.
9. *Wade C., Allan J.* Passage Retrieval and Evaluation. CIIR Technical Report IR-396, 2005. — 9p. (http://ciir.cs.umas.edu/pubfiles/ir-396.pdf)
10. *Kim J., Kim M.* An evaluation of passage-based text categorization. Journal of Intelligent Information Systems, 23, 2004. pp. 47–65.
11. *Sebastiani F.* Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 2002. — pp. 1–47.
12. *Text* Mining: classification, clustering, and applications / Srivastava A., Sahami M. CRC Press, 2009. — 290 p.
13. *Paige C. C., Saunders M. A.* LSQR: An Algorithm for Sparse Linear Equations And Sparse Least Squares. ACM Trans. Math. Soft., Vol.8, 1982. — pp.43–71.