УДК 004.02

Новый метод резюмирования текстовых документов и оценка результата классификации в трех аспектах

Р. М. АЛГУЛИЕВ, Р. М. АЛЫГУЛИЕВ

Институт информационных технологий НАН Азербайджана, г. Баку
rasim@science.az, a.ramiz@science.az

В целях классификации текстовых документов предложен новый метод резюмирования, суть которого заключается в определении счета релевантности каждого предложения в документе. Счет релевантности предложения, определяемый сравнением его со всеми остальными предложениями в документе и заглавием, измеряется мерой косинуса. Результат классификации оценивается в трех аспектах: гомогенность и гетерогенность классов; мера подобия между результатом классификации и точной классификацией; надежность классификации.

Данная работа посвящена задачам классификации текстовых документов через резюмирование. Из всех видов информации, собранной на WWW, наибольший интерес, как правило, представляют текстовые данные. В Интернете каждый день появляются сотни новых текстовых документов, пополняющих и без того большое количество доступной текстовой информации. При этом применения текстового поиска не ограничиваются поиском web-страницы в Интернете, а имеют множество важных приложений в современном мире, таких как классификация текстовых объектов, создание автоматизированных справочных систем и т. д. В таких случаях остро стоит проблема интеллектуального анализа текстовых данных.

Для интеллектуального анализа данных (data mining) используют разные подходы [1]. Одним из таких подходов является классификация. Классификация заключается в разбиении выборки из текстовых документов на непересекающиеся группы в целях обеспечения максимальной "близости" (подобия) между соответствующими определенной смысловой тематике документами одной группы и максимального различия между группами [2].

Для классификации текстовых документов имеются разные подходы [1, 3]. Большинство методов классификации, которые основаны на использовании модели векторного пространства [4—6], анализируют отдельные слова в документах. Модель векторного пространства представляет документы как характеристический вектор слов, которые появляются во всем наборе документов. Каждый характеристический вектор содержит веса слов — обычно частоту их появления в наборе документов. Подобие между документами измеряют, используя одну из мер подобия — меру косинуса, Евклидову меру или меру Jaccard.

Чтобы достичь высокой точности классификации документов, приходится учитывать более информативные особенности слов. С этой целью, например, в работе [7] с помощью генетического алгоритма определяются веса HTML тегов, которые влияют на эффективность поиска информации. В работе [8] классификация документов проводится на уровне отдельных слов, но, в отличие от классических работ, здесь определяется релевантность каждого слова относительно их информативных особенностей, которыми являются частота появления слова в заглавии, выделенные (курсивом, жирным шрифтом, подчеркиванием) слова и позиция слова на странице. Позиция, влияющая на релевантность слова, в [8] определяется следующим образом: каждая страница делится на четыре части; первой и четвертой частям страницы присваивается вес, равный 3/4, а второй и третьей частям страницы — вес, равный 1/4. Отметим, что такое эвристическое определение весов не оправдывает себя в задаче классификации документов.

Для повышения точности классификации в работе [9] предложен алгоритм DIG (Document Index Graph), основанный на теории графов, который учитывает фразы и их веса. Здесь под фразой понимается последовательность слов, а не грамматически структурированное предложение. Предложенный в работе [10] алгоритм GIS (Generalized Instance Set) объединяет методы k-ближайших соседей и линейного классификатора.

В последние годы в классификации текстовых документов широко используется методика резюмирования [11], которую называют предварительной обработкой классификации. Методика резюмирования применяется для извлечения значимых контекстов [12], предложений [13—16] и параграфов [17, 18]. В работе [12] для резюмирования web-страницы исследован эффект контекста, который содержит информацию, извлеченную из содержания всех документов, связанных с данной страницей. Для извлечения важных параграфов из текста в работе [17] применена методика TRM (Text Relationship Map), а в работе [18] перед извлечением значимых параграфов сначала их кластеризируют методом k-средних. В работе [15] для резюмирования текста используются два метода определения счета релевантности предложения. В первом методе по статистическим χ^2-значениям ранжируются слова по категориям, а потом по формуле TF*IDF (Term Frequency, Inverse Document Frequency) с учетом категорий слов определяется значимость предложения. Во втором методе значимость предложения определяется мерой подобия между ним и заглавием.

В работе [13] для резюмирования газетных статей взвешенной комбинацией статистической и лингвистической особенностей предложения вычисляется его счет релевантности. Для увеличения точности классификации в работе [16] использовано четыре метода резюмирования web-страницы. По каждому методу вычисляется счет релевантности предложения. Результирующий счет релевантности предложения является суммой этих счетов.

В работе [14] извлечение значимых предложений проводится двумя методами. Суть первого метода заключается в следующем:

1) документ представляется как множество S предложений;

2) для каждого предложения $i \in S$ и документа в целом вычисляются векторы A_i и D взвешенных слов соответственно;

3) для каждого предложения $i \in S$ вычисляется счет релевантности, который определяется скалярным произведением векторов A_i и D;

4) выбирается предложение k, имеющее наибольший счет релевантности;

5) предложение k удаляется из множества S, слова, содержащиеся в этом предложении, исключаются из документа и заново вычисляется вектор D взвешенных слов для документа;

6) если число предложений в резюме достигает заранее определенного значения, операция завершается, в противном случае перейти к шагу 3.

Второй метод резюмирования основан на методике LSA (Latent Semantic Analysis).

Сразу отметим, что методы, предложенные в работе [14], имеют несколько недостатков. Во-первых, при вычислении векторов взвешенных слов не учитываются их информативные особенности, во-вторых, счет релевантности предложения определяется между ним и документом в целом, что не позволяет точно определить счет релевантности предложения в документе. (Отметим, что такой же недостаток имеет методика, предложенная в работе [16]). И, наконец, с вычислительной точки зрения этот метод неэффективен, ибо на каждой итерации заново вычисляются вектор D взвешенных слов документа и счет релевантности предложений. Еще одним недостатком является то, что после удаления предложения k из множества S и исключения слов, содержащихся в этом предложении, из документа вектор взвешенных слов заново вычисляется только для документа. А это влияет на результат резюмирования.

Предлагаемая читателю работа посвящена классификации текстовых документов через резюмирование. Прежде чем резюмировать документ, в предложенной работе сначала определяется счет релевантности каждого слова относительно их информативности. После этого вычисляется счет релевантности каждого предложения путем его сравнения со всеми другими предложениями и заглавием документа. Наконец, согласно счету релевантности предложение включается в резюме. Кроме того, в работе оцениваются результаты резюмирования и классификации.

Вычисление релевантности слов

Так как основным носителем информации в текстовых документах являются слова, в задачах классификации сначала следует определить вес каждого слова в документе.

Выбор информативных признаков слов и определение их веса. Известно, что слова в документах могут встречаться в разных формах написания, что дает дополнительно существенную информацию о значимости слов. В целях повышения точности классификации следует учитывать эти информативные признаки.

По нашему мнению, такими информативными признаками могут быть:

- выделенные (курсивом, жирным шрифтом, подчеркиванием) слова;
- слова, написанные прописными буквами, и
- размер шрифта.

Введем следующие обозначения: $N(w, d)$ — общее число слов w в документе d; $N(s, d)$ —

общее число предложений s в документе d; $N^E(w, d)$ — общее число выделенных слов w в документе d; $N^U(w, d)$ — общее число слов w в документе d, написанных прописными буквами; $N^L(w, d)$ — общее число слов w в документе d, написанных крупными шрифтами; $N^S(w, d)$ — общее число слов w в документе d, написанных мелкими шрифтами; $N^R(w, d)$ — общее число слов w в документе d, написанных обычными шрифтами; $N^E(w_j, s_i)$ — число появлений j-го выделенного слова w_j в i-м предложении s_i ($i = 1, ..., N(s, d); j = 1, ..., N^E(w, d)$); $N^U(w_j, s_i)$ — число появлений j-го слова w_j, написанного прописными буквами, в i-м предложении s_i ($i = 1, ..., N(s, d); j = 1, ..., N^U(w, d)$); $N^L(w_j, s_i)$ — число появления j-го слова w_j, написанного крупными шрифтами, в i-м предложении s_i ($i = 1, ..., N(s, d); j = 1, ..., N^L(w, d)$); $N^S(w_j, s_i)$ — число появлений j-го слова w_j, написанного мелкими шрифтами, в i-м предложении s_i ($i = 1, ..., N(s, d); j = 1, ..., N^S(w, d)$); $N^R(w_j, s_i)$ — число появления j-го слова w_j, написанного обычными шрифтами, в i-м предложении s_i ($i = 1, ..., N(s, d); j = 1, ..., N^R(w, d)$).

Здесь под словами, написанными обычными шрифтами, понимаются те слова, которые не имеют указанных выше признаков.

Принимая во внимание введенные обозначения, определим следующие виды функции частот появления j-го слова w_j в i-м предложении s_i:

$$f_{ij}^{Lett} = \begin{cases} \dfrac{N^{Lett}(w_j, s_i)}{N^{Lett}(w, d)}, & \text{если } N^{Lett}(w, d) \neq 0; \\ 0 \text{ — в противном случае,} \end{cases}$$

где $Lett \in \{E, U, L, S, R\}$.

Переходим к определению веса каждого слова в предложении в зависимости от особенностей этого слова. Вес j-го слова w_j зависит от частоты его появления в i-м конкретном предложении s_i и в документе d. Для этого будем использовать формулу TF*IDF, которая присваивает наибольший вес среднечастотным словам и незначительные веса часто употребляемым словам общего назначения и редким словам.

Таким образом, веса каждого слова с учетом его особенностей определяются следующими формулами:

$$\omega_{ij}^{Lett} = f_{ij}^{Lett} \log_2 \left(\frac{N(s, d) + 0{,}5}{N^{Lett}(s, d, w_j) + 0{,}5} \right),$$

где $N^{Lett}(s, d, w_j)$ — число предложений s в документе d, в которых присутствует j-е слово w_j.

В последней формуле во избежание появления нуля под логарифмом введено смещение 0,5.

Поскольку определены веса каждого j-го слова w_j в i-м предложении s_i документа d с учетом его особенностей, найдем обобщенный вес каждого j-го слова w_j в i-м предложении s_i документа d, который вычисляется по следующей формуле

$$\omega_{ij} = \alpha_1 \omega_{ij}^E + \alpha_2 \omega_{ij}^U + \alpha_3 \omega_{ij}^L + \alpha_4 \omega_{ij}^S + \alpha_5 \omega_{ij}^R,$$

где коэффициенты $\alpha_k \in [0, 1]$ ($k = \overline{1, 5}$) и удовлетворяют следующему условию:

$$\sum_{k=1}^{5} \alpha_k = 1. \qquad (1)$$

Выбор коэффициентов α_k ($k = \overline{1, 5}$) осуществляется с помощью генетического алгоритма на обучающих выборках.

Определение весов информативных признаков с помощью генетического алгоритма. Применению классического генетического алгоритма предшествует представление решения задачи в виде хромосомы, после чего становится возможным применение алгоритма [19, 20]. Исходя из специфики задачи, хромосомы представляем в виде комбинации коэффициентов (α_1, α_2, α_3, α_4, α_5).

Проверка того, насколько хорошим является i-е решение задачи, осуществляется с помощью вычисления функции приспособленности (fitness function) для каждой хромосомы $\alpha^i = (\alpha_1^i, \alpha_2^i, \alpha_3^i, \alpha_4^i, \alpha_5^i)$ ($i = 1, ..., N_{pop}$; N_{pop} — число хромосом в популяции).

Обычно функция приспособленности в явном виде содержит критерий оптимизации решаемой задачи. Исходя из этого соображения, в качестве функции приспособленности берем среднее значение $F_1^{summary}$, которое будет оп-

ределено ниже, по всей обучающей выборке размера N_{train}:

$$fitness(\alpha^i) = \frac{\sum_{j=1}^{N_{train}} F_1^{summary}(d_j)}{N_{train}}, \quad i = 1, ..., N_{pop},$$

где $F_1^{summary}(d_j)$ является F_1-критерием, соответствующим j-му документу d_j в обучающей выборке.

Тогда функция приспособленности всей популяции $\alpha = (\alpha^1, \alpha^2, ..., \alpha^{N_{pop}})^T$, т. е. конечная цель будет определяться таким образом:

$$F(\alpha) = \max_i fitness(\alpha^i), \quad i = 1, ..., N_{pop}. \quad (2)$$

Итак, определение коэффициентов α_k (k = = 1, ..., 5) сведено к задаче (2), (1), которая является задачей оптимизации с ограничением. Классический генетический алгоритм легко применяется к задачам оптимизации без ограничения. Однако применение классического генетического алгоритма к задачам оптимизации с ограничениями сталкивается с проблемой появления недопустимых решений, т. е. хромосом, которые нарушают ограничения (в нашем случае условие (1)).

При решении оптимизационной задачи с ограничениями генетическим алгоритмом важным является постоянная поддержка допустимости хромосом в течение работы алгоритма такими, чтобы они не нарушали ограничений. В этих целях вводят штрафную функцию, и оптимизационная задача с ограничениями приводится к задаче без ограничения. Эффективность использования штрафной функции в задачах оптимизации с ограничениями была показана в работе [21], где доказано, что введение штрафной функции приводит к быстрому нахождению решения и исключает преждевременное завершение генетического алгоритма.

После введения штрафной функции целевая функция будет иметь следующий вид:

$$E(\alpha) = F(\alpha) Penalty(\alpha_\Sigma).$$

Цель задачи состоит в том, чтобы найти оптимальное решение α, которое максимизирует функцию $E(\alpha)$. Штрафная функция $Penalty(\alpha_\Sigma)$ определяется так:

$$Penalty(\alpha_\Sigma) = \frac{1 + \min(1, \alpha_\Sigma)\max(1, \alpha_\Sigma)}{2\max(1, \alpha_\Sigma)},$$

где $\alpha_\Sigma = \sum_{k=1}^{5} \alpha_k$.

Легко проверить, что, если решение допустимое, т. е. $\alpha_\Sigma = 1$, то $Penalty(\alpha_\Sigma) = 1$. Следовательно, в этом случае функция $E(\alpha)$ будет равна целевой функции $F(\alpha)$. И когда решение является недопустимым, т. е. $\alpha_\Sigma < 1$ или $\alpha_\Sigma > 1$, то $Penalty(\alpha_\Sigma) < 1$. Отсюда следует, что штрафная функция, уменьшая значение функции $E(\alpha)$, заранее предотвращает появления недопустимых решений в следующих шагах генетического алгоритма.

Резюмирование документа

При резюмировании текстовых документов целью является автоматический выбор таких пассажей (здесь пассажами могут быть фразы, предложения или параграфы), которые адекватно отражают суть документа. В этом разделе предложен метод (так называемый "Предложение с предложением") извлечения значимых предложений из текста в резюме, основанный на определении счета релевантности каждого предложения. Суть предложенного метода заключается в том, что каждое предложение сравнивается со всеми остальными предложениями и заглавием, т. е. вычисляется мера близости между ними, со всеми остальными предложениями и с заглавием.

Определение счета релевантности предложения. Прежде чем вычислить меру близости между предложениями, каждое предложение s_i идентифицируем с характеристическим вектором $s_i = (\omega_{i1}, ..., \omega_{iN(w, d)})$ ($i = 1, ..., N(s, d)$) слов, которые появляются в документе d, а затем используем меру косинуса:

$$sim(s_i, s_j) = \cos(s_i, s_j) = \frac{\sum_{k=1}^{N(w, d)} \omega_{ik}\omega_{jk}}{\sqrt{\sum_{k=1}^{N(w, d)} \omega_{ik}^2}\sqrt{\sum_{k=1}^{N(w, d)} \omega_{jk}^2}},$$
$$i, j = 1, ..., N(s, d).$$

Далее определим суммарную меру близости между i-м и остальными предложениями:

$$sim_\Sigma(s_i) = \sum_{\substack{j=1 \\ j \neq i}}^{N(s, d)} sim(s_i, s_j).$$

Кроме того, известно, что одним из носителей информации в документе является заглавие. Поэтому, чтобы учесть вклад заглавия в резю-

мирование, определим меру близости между ним и i-м предложением:

$$sim_{title}(s_i) = sim(s_i, \boldsymbol{T}), \quad i = 1, ..., N(s, d),$$

где $\boldsymbol{T}$ — характеристический вектор слов, соответствующий заглавию.

Так как каждое предложение и заглавие имеют разный вклад в резюмирование, в окончательный счет релевантности i-го предложения они будут входить с соответствующими весами:

$$score(s_i) = \beta_1 sim_{\Sigma}(s_i) + \beta_2 sim_{title}(s_i), \quad i = 1, ..., N(s, d),$$

где $\beta_1, \beta_2 \in [0, 1]$ и $\beta_1 + \beta_2 = 1$. Эти веса тоже определяются с помощью генетического алгоритма.

Следующим этапом является включение предложений в резюме в зависимости от их счета релевантности. Прежде чем включить предложения в резюме, их ранжируют в порядке убывания их счетов релевантности. После ранжирования, начиная с предложения с наивысшим счетом, в резюме включают те предложения, счет релевантности которых больше заданного порога $score(s_i) > \theta$ ($i = 1, ..., N(s, d)$). Процесс продолжается до тех пор, пока коэффициент сжатия $rate_{comp}$ удовлетворяет заданному ограничению. В работе [22] было показано, что, если коэффициент сжатия находится в интервале [0,05; 0,3], то резюме считается приемлемым:

$$rate_{comp} = \frac{len_{summary}}{len_{document}},$$

где $len_{summary}$, $len_{document}$ — длины (число слов) резюме и документа соответственно.

Оценка резюмирования. Для оценки результата резюмирования используем F_1-критерии. Пусть $N^{rel}_{document}$ — число релевантных предложений в документе, $N^{rel}_{summary}$ — число релевантных предложений в резюме, $N_{summary}$ — число предложений в резюме, $P_{summary}$ — точность резюмирования, $R_{summary}$ — полнота. Отсюда следует, что

$$P_{summary} = \frac{N^{rel}_{summary}}{N_{summary}};$$

$$R_{summary} = \frac{N^{rel}_{summary}}{N^{rel}_{document}};$$

$$F_1^{summary} = \frac{2P_{summary}R_{summary}}{P_{summary} + R_{summary}}.$$

Классификация документов

Задача классификации документов состоит в автоматическом отнесении нового документа к какому-либо известному системе классу. Существуют многочисленные методы решения задачи классификации. В нашей работе использованы наиболее широко распространенные методы: наивный байесовский (Naïve Bayes), k-ближайших соседей (k-Nearest Neighbor) и Rocchio.

Наивный байесовский метод. Вероятностная модель классификации для определения принадлежности документа d_i ($i = 1, ..., N$; N — число документов) классу c_k ($k = 1, ..., K$; K — число классов) использует вероятность

$$P(c_k|d_i) = \frac{P(c_k)P(d_i|c_k)}{P(d_i)},$$

где $P(c_k)$ — априорная вероятность появления документов класса c_k; $P(d_i)$ рассчитывается по формуле полной вероятности события d_i.

Учитывая, что, как и в случае поиска текста, величина $P(d_i)$ является константой в процессе классификации, и ею можно пренебречь. Для вычисления $P(d_i|c_k)$ обычно используют предположение о независимости слов (и в этом заключается "наивность" метода) в документах и классах, что приводит к формуле

$$P(c_k|d_i) = P(c_k) \prod_{j=1}^{NV} P(w_j|c_k)^{N(w_j, d_i)}. \quad (3)$$

Здесь $N(w_j, d_i)$ — число появлений слова w_j в документе d_i; NV — общее число слов в наборе документов $D = (d_1, d_2, ..., d_N)$; $P(w_j|c_k)$ — вероятность появления j-го слова w_j документа d_i в классе c_k, которая вычисляется по формуле

$$P(w_j|c_k) = \frac{\sum_{d_i \in D_k} N(w_j, d_i)}{\sum_{w_j \in V} \sum_{d_i \in D_k} N(w_j, d_i)},$$

где V — словарник (общее число слов) во множестве документов D; D_k — множество документов в классе c_k.

Многие из слов w_j появляются только в нескольких классах, следовательно, вероятность $P(w_j|c_k)$ будет нулевой для тех классов c_k, где слово w_j не появляется. Тогда из формулы (3) следует, что вероятность $P(c_k|d_i)$ также будет нулевой, если документ d_i будет содержать хотя бы

одно такое слово. Во избежание этого на практике пользуются оценкой Лапласа:

$$P(w_j|c_k) = \frac{1 + \sum\limits_{d_i \in D_k} N(w_j, d_i)}{NV + \sum\limits_{w_j \in V} \sum\limits_{d_i \in D_k} N(w_j, d_i)},$$

$$i = 1, ..., N; \quad k = 1, ..., K; \quad j = 1, ..., NV.$$

Перейдя в формуле (3) к логарифмическому масштабу и разделив полученное равенство на $N(w, d_i)$ — общее число слов в документе d_i, получим

$$P(c_k|d_i) = \frac{\log(P(c_k))}{N(w, d_i)} + \sum_{j=1}^{NV} P(w_j, d_i)\log(P(w_j|c_k)),$$

где было принято $P(w_j|d_i) = \frac{N(w_j, d_i)}{N(w, d_i)}$.

Согласно вероятностной модели документ d_i ($i = 1, ..., N$) принадлежит к тому классу c_k ($k = 1, ..., K$), для которого значение $P(c_k|d_i)$ максимально.

Отметим, что в рамках рассмотренного подхода возможна другая интерпретация полученных результатов. Каждый документ и каждый класс могут быть представлены как вероятностные распределения на множестве всех известных системе слов. Для определения близости документа d_i классу c_k обычно пользуются информационными мерами, такими как мера Kullback—Leibler:

$$score(d_i, c_k) = \frac{\log(P(c_k))}{N(w, d_i)} + \sum_{j=1}^{NV} P(w_j|d_i)\log\left(\frac{P(w_j|c_k)}{P(w_j|d_i)}\right).$$

Метод *k*-ближайших соседей. Этот метод применяют во многих задачах для определения класса, которому принадлежит документ. Метод основан на использовании при классификации уже имеющегося набора классифицированных документов. Для каждого нового документа d_{N+1}, поступающего в систему, определяются k ближайших к нему документов, уже отнесенных к одному из классов.

При использовании метода k-ближайших соседей для классификации документов исследователю приходится решать проблему выбора метрики для определения близости объектов. Если используется Евклидово расстояние

$$dist(d_i, d_j) = \sqrt{\sum_{l=1}^{NV} (\omega_{il} - \omega_{jl})^2}, \quad i, j = 1, ..., N,$$

то для каждого из классов c_k вычисляется сумма расстояний от нового документа d_{N+1} до каждого из k документов, отнесенных ранее к этому кассу:

$$dist_\Sigma(d_{N+1}, c_k) = \sum_{d' \in O_k(d_{N+1}) \cap D_k} dist(d_{N+1}, d') = \sum_{i \in I_k(d_{N+1})} dist(d_{N+1}, d_i)e_{ik},$$

где $O_k(d_{N+1})$, $I_k(d_{N+1})$ — элементы и их индексы k-ближайших соседей документа d_{N+1} соответственно;

$$e_{ik} = \begin{cases} 1, \text{ если } i\text{-й документ относится} \\ \qquad k\text{-му классу;} \\ 0 \text{ — в противном случае.} \end{cases}$$

Новый документ d_{N+1} включается в класс c_k с наименьшим значением $dist_\Sigma(d_{N+1}, c_k)$.

Если используется мера косинуса, то классификатор k-ближайших соседей каждому кандидату — классу c_k — присваивает счет релевантности, определяемый следующей формулой:

$$score(d_{N+1}, c_k) = \sum_{d' \in O_k(d_{N+1}) \cap D_k} \cos(d_{N+1}, d') = \sum_{i \in I_k(d_{N+1})} \cos(d_{N+1}, d_i)e_{ik}.$$

Документ d_{N+1} будет отнесен к тому классу c_k, для которого значение $score(d_{N+1}, c_k)$ максимально.

Метод Rocchio. Одним из распространенных методов линейной классификации является алгоритм Rocchio, в соответствии с которым для вычисления вектора весов некоторой классификации используется формула

$$v_{kj}^{avg} = \frac{1}{N_k} \sum_{i=1}^{N} v_{ij}e_{ik},$$

$$j = 1, ..., N(w, d_i); \quad k = 1, ..., K,$$

где v_{kj}^{avg} — средний вес j-го слова w_j в документах класса c_k; N_k — число документов класса c_k; v_{ij} — вес j-го слова w_j в i-м документе d_i.

При классификации документов методом k-ближайших соседей требуется сохранить все элементы выборки, т. е. элементы матриц $\boldsymbol{V} = \|v_{ij}\|$ и $\boldsymbol{E} = \|e_{ik}\|$. Отсюда следует, что при большом количестве документов использование классификатора k-ближайших соседей с точки зрения памяти неэффективно. Для решения этой

проблемы в работе [23] предложен адаптивный метод Rocchio:

$$v_{kj}^{avg}(\tau) = \frac{N_k^{\tau-1}}{N_k^{\tau}} v_{kj}^{avg}(\tau - 1) + \frac{1}{N_k^{\tau}} \sum_i v_{ij} e_{ik},$$

где N_k^{τ} — число документов класса c_k, собранных за время τ. Второй член в правой части равенства является суммой весов j-го слова w_j в документах класса c_k, которые получены в интервале времени между $\tau - 1$ и τ.

Оценка классификации

В общем случае результат классификации оценивают с трех точек зрения:

1) качество классификации сначала оценивается в терминах гомогенности — документы одного класса "подобны" друг другу — и гетерогенности классов — документы разных классов "не подобны" друг другу;

2) при оценке классификации полагаются на степень общности между ее результатом и эталоном (точной классификацией);

3) оценивается надежность классификации или вероятность того, что классы структурированы неслучайно.

Гомогенность и гетерогенность классов. Гомогенность измеряет "подобие" документов в классе c_k. Например,

$$hom_1(c_k) = \frac{1}{N_k(N_k - 1)} \sum_{\substack{d_i, d_j \in c_k \\ d_i \neq d_j}} sim(d_i, d_j).$$

Это выражение описывает гомогенность класса c_k со средней попарной близостью документов. Альтернативное определение вычисляет гомогенность класса c_k относительно "центроида" c_k^0 класса, т. е.

$$hom_2(c_k) = \frac{1}{N_k} \sum_{d_i \in c_k} sim(d_i, c_k^0),$$
$$k = 1, ..., K; \quad i = 1, ..., N.$$

Гетерогенность определяет различие между классами c_p и c_q ($p \neq q$). Например,

$$het_1(c_p, c_q) = \frac{1}{N_p N_q} \sum_{\substack{d_i \in c_p \\ d_j \in c_q}} sim(d_i, d_j)$$

или

$$het_2(c_p, c_q) = sim(c_p^0, c_q^0); \quad p, q = 1, ..., K.$$

Поскольку определения гомогенности и гетерогенности основаны на вычислении подобия документов, то качество классификации повышается с увеличением значения гомогенности класса c_k и уменьшением значения гетерогенности между классом c_k и другими классами.

Если при классификации $\boldsymbol{C} = (c_1, c_2, ..., c_K)$ гомогенность каждого класса и гетерогенность между парами классов уже определены, то можно вычислить среднюю оценку гомогенности и гетерогенности классификации $\boldsymbol{C}$, пользуясь следующими формулами [24]:

$$hom^{avg} = \frac{1}{N} \sum_{c_k \in C} N_k hom_2(c_k);$$

$$het^{avg} = \frac{1}{\sum_{p \neq q} N_p N_q} \sum_{p \neq q} N_p N_q het_2(c_p, c_q)$$

соответственно.

Сравнение с точной классификацией. Пусть известна точная классификация $\boldsymbol{C}^+ = (c_1^+, c_2^+, ..., c_{K^+}^+)$ документов $\boldsymbol{D} = (d_1, d_2, ..., d_N)$. Результат классификации оценивается степенью общности между классификациями $\boldsymbol{C}$ и $\boldsymbol{C}^+$.

Количественная характеристика степени общности двух классификаций определяется с использованием понятия меры подобия, применяемого в теории автоматической классификации. Прежде чем выбрать меру подобия, сначала для каждой классификации $\boldsymbol{C}$ и $\boldsymbol{C}^+$ вводят соответствующие бинарные матрицы $\boldsymbol{B} = \|b_{ij}\|$ и $\boldsymbol{B}^+ = \|b_{ij}^+\|$ $(i, j = 1, ..., N)$, элементы которых определяются аналогичным образом:

$$b_{ij} = \begin{cases} 1, \text{если в } \boldsymbol{C} \text{ документы } d_i \text{ и } d_j \\ \quad \text{относятся к одному классу;} \\ 0 \text{ — в противном случае;} \end{cases}$$

$$b_{ij}^+ = \begin{cases} 1, \text{если в } \boldsymbol{C}^+ \text{ документы } d_i \text{ и } d_j \\ \quad \text{относятся к одному классу;} \\ 0 \text{ — в противном случае;} \end{cases}$$

В теории автоматической классификации существует множество функций для вычисления мер подобия между объектами, описанными в виде бинарных векторов (коэффициент Jaccard,

индекс Folkes и Mallows, мера Minkowski, индекс Huberts, индекс Rand и т. д.) [25]. Ниже приведено несколько видов функций подобия:

$$S_1 = \frac{p_{11}}{p_{11} + p_{10} + p_{01}};$$

$$S_2 = \sqrt{\frac{p_{11}^2}{(p_{11} + p_{10})(p_{11} + p_{01})}};$$

$$S_3 = \frac{p_{11} + p_{00}}{p_{11} + p_{10} + p_{01} + p_{00}};$$

$$S_4 = \frac{1}{p_{11} + p_{10} + p_{01} + p_{00}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} b_{ij} b_{ij}^{+};$$

$$S_5 = \frac{p_{11}}{\min(p_{11} + p_{10} p_{11} + p_{01})};$$

$$S_6 = \sqrt{\frac{p_{10} + p_{01}}{p_{11} + p_{01}}};$$

$$S_7 = \frac{2p_{11}}{2p_{11} + p_{10} + p_{01}}.$$

Здесь p_{11} — число пар документов (d_i, d_j), причем в обеих классификациях $\boldsymbol{C}$ и $\boldsymbol{C}^+$ документы d_i и d_j относятся к одному классу. Величину p_{11} вычисляют на основе бинарных матриц $\boldsymbol{B}$ и $\boldsymbol{B}^+$ следующим образом:

$$p_{11} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} x_{ij}.$$

где $x_{ij} = \begin{cases} 1, \text{если } b_{ij} = 1 \text{ и } b_{ij}^{+} = 1; \\ 0 \text{ — в противном случае;} \end{cases}$

p_{10} — число пар документов (d_i, d_j), в которых документы d_i и d_j в классификации $\boldsymbol{C}$ принадлежат к одному классу, а в классификации $\boldsymbol{C}^+$ относятся к разным классам. Величину p_{10} вычисляют как:

$$p_{10} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} y_{ij},$$

где $y_{ij} = \begin{cases} 1, \text{если } b_{ij} = 1 \text{ и } b_{ij}^{+} = 0; \\ 0 \text{ — в противном случае;} \end{cases}$

p_{01} — количество пар документов (d_i, d_j), в которых документы d_i и d_j в классификации $\boldsymbol{C}$ относятся к разным классам, а в классификации $\boldsymbol{C}^+$ принадлежат к одному классу. Величину p_{01} вычисляют как:

$$p_{01} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} z_{ij},$$

где $z_{ij} = \begin{cases} 1, \text{если } b_{ij} = 0 \text{ и } b_{ij}^{+} = 1; \\ 0 \text{ — в противном случае;} \end{cases}$

p_{00} — число пар документов (d_i, d_j), в которых документы d_i и d_j в обеих классификациях $\boldsymbol{C}$ и $\boldsymbol{C}^+$ относятся к разным классам. Величину p_{00} вычисляют по формуле

$$p_{00} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} t_{ij},$$

где $t_{ij} = \begin{cases} 1, \text{если } b_{ij} = 0 \text{ и } b_{ij}^{+} = 0; \\ 0 \text{ — в противном случае.} \end{cases}$

Для функций S_1—S_5 было доказано, что чем больше значение этих индексов, тем больше подобие между результатами классификаций $\boldsymbol{C}$ и $\boldsymbol{C}^+$. Анализ функций подобия S_1—S_6 показывает, что они дают сравнимые результаты, однако наиболее точно степень общности двух классификаций отражает функция S_7.

Надежность классификации. Пусть N — общее число документов, из которых M документов являются релевантными. Далее, пусть n — необходимое число документов, которые должны быть классифицированы в классы c_k ($k = 1, ..., K$), и $P_m(N, M, n)$ — вероятность классификации n документов такая, что m документов из них будут "релевантными". Тогда вероятность $P_m(N, M, n)$ можно вычислить по формуле гипергеометрического распределения:

$$P_m(N, M, n) = \frac{C_M^m C_{N-M}^{n-m}}{C_N^n}, \qquad (4)$$

где $C_a^b = \frac{a!}{b!(a-b)!}$ — биномиальный коэффициент.

Гипергеометрическое распределение связано с выбором без возвращения, а именно: формула (4) указывает вероятность получения ровно m

релевантных документов в случайной выборке объема n из генеральной совокупности, содержащей N документов, среди которых M "релевантных" и $N - M$ — "нерелевантных" документов. При этом вероятность (4) определена лишь для

$$\max(0,\ M + n - N) \leqslant m \leqslant \min(n,\ M).$$

Однако определение (4) можно использовать при всех $m \geqslant 0$, так как можно считать, что $C_a^b = 0$ при $b > a$, поэтому равенство $P_m(N,\ M,\ n) = 0$ нужно понимать как невозможность получить в выборке m релевантных документов. Сумма значений $P_m(N,\ M,\ n)$, распространенная на все выборочное пространство, равна 1. Если обозначить $\frac{M}{N} = p$, то формулу (4) можно переписать в иной форме:

$$P_m(N,\ M,\ n) = C_n^m \frac{A_{Np}^m A_{Nq}^{n-m}}{A_N^n},$$

где $A_a^b = C_a^b b!$ и $p + q = 1$.

Если p постоянно и $N \to \infty$, то имеет место биномиальное приближение

$$P_m(N,\ M,\ n) \sim C_n^m p^m q^{n-m}.$$

Среднее значение гипергеометрического распределения не зависит от числа N и совпадет со средним np соответствующего биномиального распределения. Дисперсия гипергеометрического распределения $\sigma^2 = npq\frac{N-n}{N-1}$ не превышает дисперсию биномиального закона $\sigma^2 = npq$.

Пусть $N_{rel}(N,\ M,\ n)$ — ожидаемое число релевантных документов в классе c_k. Тогда

$$N_{rel}(N,\ M,\ n) = \sum_{m=0}^{M} mP_m(N,\ M,\ n).$$

Если учесть соотношение $C_M^m = \frac{M}{m} C_{M-1}^{m-1}$, то получим

$$N_{rel}(N,\ M,\ n) = \sum_{m=0}^{M} M\frac{C_{M-1}^{m-1} C_{N-M}^{n-m}}{C_N^n} =$$

$$= \frac{M}{C_N^n} \sum_{m=0}^{M} C_{M-1}^{m-1} C_{N-M}^{n-m}.$$

Из равенства $\sum_{m=0}^{M} C_{M-1}^{m-1} C_{N-M}^{n-m} = C_{N-1}^{n-1}$ следует, что

$$N_{rel}(N,\ M,\ n) = \frac{M}{C_N^n} C_{N-1}^{n-1}$$

и, следовательно,

$$N_{rel}(N,\ M,\ n) = \frac{Mn}{N}.$$

В результате мы можем вычислить точность и полноту классификации:

$$P_{class} = \frac{N_{rel}(N,\ M,\ n)}{n} = \frac{M}{N};$$

$$R_{class} = \frac{N_{rel}(N,\ M,\ n)}{M} = \frac{n}{N};$$

$$F_1^{class} = \frac{2P_{class}R_{class}}{P_{class} + R_{class}}.$$

Заключение

Огромное количество информации, с которой нам приходится иметь дело, — это текстовая информация. Книги, журналы, руководства, web-страницы, электронные и обычные письма — все это примеры того, насколько текст важен для людей. В связи с развитием компьютерной техники и телекоммуникационных технологий количество текстовой информации постоянно растет. Поэтому возникает необходимость в средствах, обеспечивающих обработку этой информации и облегчающих доступ к ней с учетом требований конкретных приложений. В качестве таких средств могут выступать средства автоматической классификации текстовых документов. Задача автоматической классификации текстов в настоящее время интересует многих специалистов по всему миру. В отличие от других задач классификации, при обработке текста приходится сталкиваться с такими проблемами, которые затрудняют применение некоторых наиболее часто используемых методов. Специфика задачи состоит в том, что количество признаков, по которым происходит классификация, здесь велико, а сами их значения изменяются незначительно.

С учетом изложенного выше в данной статье предложен метод рзюмирования текстовых документов, который учитывает информативные признаки слов и значимости предложений.

Представленный в данной статье метод резюмирования состоит из следующих шагов:

- выбираются информативные признаки слов, которые, по нашему мнению, влияют на точность результата резюмирования и, следовательно, на результат классификации;
- с помощью генетического алгоритма определяются веса информативных признаков, влияющие на релевантность слов;
- с помощью косинусоидальной метрики вычисляется суммарная мера подобия между каждым предложением и остальными предложениями;
- вычисляется мера подобия между каждым предложением и заглавием;
- определяется взвешенный счет релевантности каждого предложения;
- ранжируются счета релевантности предложений;
- начиная с высшего, в резюме включаются предложения, счет релевантности которых больше заданного порога, и процесс продолжается до тех пор, пока коэффициент сжатия удовлетворяет заданному ограничению;
- наконец, оцениваются результаты резюмирования и классификации.

СПИСОК ЛИТЕРАТУРЫ

1. **Chakrabarti S.** Data mining for hypertext: a tutorial survey // ACM SIGKDD Explorations, January 2000. V. 1. N 2. P. 1—11.
2. **Jain A. K., Murty M. N., Flynn P. J.** Data clustering: a review // ACM Computing Surveys, September 1999. V. 31. N 3. P. 264—323.
3. **Толчеев В. О.** Модели и методы классификации текстовой информации // Информационные технологии. 2004. № 5. С. 6—14.
4. **Salton G., Wong A., Yang C. S.** A vector space model for automatic indexing // Communications of the ACM. November 1975. V. 18. N 11. P. 613—620.
5. **Salton G., McGill M. J.** Introduction to modern information retrieval. New York. USA. McGraw-Hill, 1986. 400 p.
6. **Salton G.** Automatic text processing: the transformation, analysis and retrieval of information by computer. Boston. USA. Addison Wesley, 1989. 530 p.
7. **Kim S., Zhang B. T.** Genetic mining of HTML structures for effective web-document retrieval // Applied Intelligence. May—June 2003. V. 18. N 3. P. 243—256.
8. **Fresno V., Ribeiro A.** An analytical approach to concept extraction in HTML environments // Journal of Intelligent Information Systems (Special Issue on Web Content Mining). May 2004. V. 22. N 3. P. 215—235.
9. **Hammouda K. M., Kamel M. S.** Efficient phrase-based document indexing for web-document clustering // IEEE Transactions on Knowledge and Data Engineering. October 2004. V. 16. N 10. P. 1279—1296.
10. **Lam W., Han Y.** Automatic textual document categorization based on generalized instance sets and a metamodel // IEEE Transactions on Pattern Analysis and Machine Intelligence, May 2003. V. 25. N 5. P. 628—633.
11. **Automatic** text structuring and summarization / G. Salton, A. Singhal, A. Mitra, C. Buckley // Information Processing and Management. March 1997. V. 33. N 2. P. 193—207.
12. **Delort J.-Y., Bouchon-Meunier B., Rifqi M.** Enhanced web-document summarization using hyperlinks // Proceedings of the 14th ACM Conference on Hypertext and Hypermedia. Nottingham. United Kingdom. August 26—30. 2003. P. 208—215.
13. **Summarization** text documents: sentence selection and evaluation metrics / J. Goldstein, M. Kantrowitz, V. Mittal, J. Carbonell // Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99). Berkeley. USA. August 15—19. 1999. P. 121—128.
14. **Gong Y., Liu X.** Generic text summarization using relevance measure and latent semantic analysis // Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01). New Orleans. Louisiana. USA. September 9—12. 2001. P. 19—25.
15. **Ko Y., Park J., Seo J.** Improving text categorization using the importance of sentences // Information Processing and Management. January 2004. V. 40. N 1. P. 65—79.
16. **Web-page** classification through summarization / D. Shen, Z. Chen, Q. Yang, H. J. Zeng, B. Zhang, Y. Lu, W. Y. Ma // Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval (SIGIR'04). Sheffield. United Kingdom. July 25—29. 2004. P. 242—249.
17. **Mitra M., Singhal A., Buckley C.** Automatic text summarization by paragraph extraction // Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. Madrid. Spain. July 7—12. 1997. P. 39—46.
18. **A study** of Chinese text summarization using adaptive clustering of paragraphs / P. Hu, T. He, D. Ji, M. Wang // Proceedings of the 4th International Conference on Computer and Information Technology (CIT'04). Wuhan. China. IEEE Computer Society. September 14—16. 2004. P. 1159—1164.
19. **Goldberg D. E.** Genetic algorithms in search, optimization and machine learning. Boston. USA. Addison Wesley. 1989. 432 p.
20. **Michalewicz Z.** Genetic algorithms + data structures = evolution programs. Berlin. Springer-Verlag. 1996. 387 p.
21. **Olsen A. L.** Penalty functions and the knapsack problem // Proceedings of the First IEEE Conference on Evolutionary Computation. Orlando. USA. June 27—29. 1994. V. 2. P. 554—558.
22. **Mani I., Maybury M. T.** Advances in automated text summarization. Cambridge. MIT Press. 1999. 442 p.
23. **Liu F., Yu C., Meng W.** Personalized web search for improving retrieval effectiveness // IEEE Transactions on Knowledge and Data Engineering. January 2004. V. 16. N 1. P. 28—40.
24. **Shamir R., Sharan R.** Click: a clustering algorithm with applications to gene expression analysis // Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00). San Diego. USA. August 19—23. 2000. P. 307—316.
25. **Halkidi M., Batistakis Y., Vazirgiannis M.** On clustering validation techniques // Journal of Intelligent Systems. December 2001. V. 17. N 2, 3. P. 107—145.