УДК 004.02:004.032.26

Р.М. Алгулиев, Р.М. Алыгулиев

АВТОМАТИЧЕСКОЕ РЕЗЮМИРОВАНИЕ ТЕКСТОВЫХ ДОКУМЕНТОВ С ПОМОЩЬЮ КЛАСТЕРИЗАЦИИ ПРЕДЛОЖЕНИЙ

Введение

Из всех видов информации, собранной в Интернете, наибольший интерес, как правило, представляют тексты. На сегодняшний день тексты — важнейший носитель информации, и они, очевидно, еще долго будут оставаться таковыми. Подавляющая часть научных статей, документации, on-line новостей и т.п. имеют текстовую форму, что обусловливает интерес к проблеме обработки и поиска текста. В настоящее время механизмы поиска в Интернете на основе ключевого слова возвращают сотни и даже тысячи документов, которые затрудняют поиск необходимой информации. Таким образом, с ростом количества текстовых документов, доступных в Интернете, обычные информационно-поисковые технологии становятся неудовлетворительными для отыскания релевантной информации. Поэтому возникает потребность в новой технологии, которая может помочь пользователю обрабатывать огромное количество информации и быстро идентифицировать самые релевантные документы.

В связи с ростом объема текстовых документов предоставление пользователю резюме каждого документа значительно облегчило бы задачу обнаружения нужной информации. Текстовой поиск и резюмирование — это две взаимодополняющие технологии.

Цель задачи автоматического резюмирования состоит в извлечении из документа информативных фрагментов (предложений, абзацев), отражающих его содержание [1]. В течение последних лет предложено множество методов резюмирования. Например, в работах [2, 3] для извлечения значимых абзацев предложен метод TRM (Text Relationship Map), идея которого заключается в представлении текста в виде графа, вершинами которого служат абзацы. Каждый абзац идентифицируется взвешенным вектором слов. Между абзацами вычисляется мера подобия, определенная скалярным произведением. Если мера подобия больше заданного порога, то эти вершины соединяются. Критерий включения абзаца в резюме определяется количеством ребер, связывающих его с другими абзацами. В работе [3] предложены четыре типа критериев выбора абзаца: bushy path, depth-first path, segmented bushy path, augmented segmented bushy path.

Большая часть работ посвящена определению степени релевантности предложения [4–10]. В частности, в [4] она определяется взвешенной комбинацией его локальной и глобальной характеристик. В [11] сформулирован метод определения локальной характеристики предложения, согласно которому вес слова определяется не по формуле TF*IDF (Term Frequency * Inverse Document Frequency), а по формуле TL*TF (Term Length * Term Frequency). Идея схемы взвешивания TL*TF [11, 12] базируется на том, что слова, появляющиеся часто, более короткие. Такие

слова не описывают контент документа, т.е. это стоп-слова. Наоборот, слова, появляющиеся редко, более длинные. Глобальная характеристика предложения определяется методом TRM. В работе [5] предлагаются два подхода: MCBA (Modified Corpus-Based Approach) и LSA+TRM (Latent Semantic Analysis+TRM). Первый подход основан на обучении, он учитывает некоторые особенности, включая позицию предложения в абзаце, позитивные и негативные ключевые слова, центральность предложения в документе и сходство предложения с заглавием. Второй подход с помощью LSA вычисляет семантическую матрицу документа, потом на основе этой матрицы конструирует семантический TRM.

В работе [6] для определения степени релевантности предложения газетных статей вводится комбинирование статистической и лингвистической особенностей. Статистическая особенность определяется стандартными методами информационного поиска, а лингвистическая — с учетом анализа резюме газетных статей. Текстовые резюме могут быть запрос-релевантными (query relevance summary) и общими (generic summary). Запрос-релевантное резюме представляет контент документа, близко связанный с запросом. Создание запрос-релевантного резюме — это, по существу, процесс извлечения из документа предложений (фрагментов), релевантных к запросу. Поэтому запрос-релевантное резюме часто формируется путем применения технологии информационного поиска, и большое количество методов резюмирования текстов относят к этой категории. Запрос-релевантное резюме полезно для того, чтобы ответить на вопрос, релевантен ли данный документ запросу пользователя, а если релевантен, то какая(ие) часть(и) документа релевантна(ы). Запрос-релевантное резюме не охватывает полный контент документа и, следовательно, не годится для его краткого обзора. Чтобы ответить на вопрос, к какой категории принадлежит документ и какие точки в документе ключевые, нужно создать общее (generic) резюме. С другой стороны, общее резюмирование позволяет обеспечить резюме с широким охватом контента документа.

Исходя из этого соображения, в работе [7] предлагаются два метода общего резюмирования. Первый метод, используя стандартные методы информационного поиска, ранжирует предложения относительно их степеней релевантности, которые определяются скалярными произведениями взвешенных векторов документа и предложений. При этом основное усилие направлено на минимизацию избыточности в резюме, без широкого охвата контента документа, поскольку после выбора предложения с наибольшим значением степени релевантности оно удаляется из документа. После удаления предложения вектор взвешенных слов документа вычисляется заново; слова, содержащиеся в удаленном предложении, не присутствуют в вычислении данного вектора. Второй метод, используя LSA, идентифицирует семантически значимые предложения.

Работа [9] посвящена резюмированию web-страниц, при котором по известным четырем методам вычисляется степень релевантности каждого предложения и окончательная степень релевантности равняется сумме этих четырех степеней. В работах [8, 10] сначала кластеризуются абзацы и предложения, а затем определяются информативные предложения. Метод, предложенный в [8], в основном состоит из трех фаз. В первой фазе создается взвешенный вектор абзацев. Во второй фазе методом k-средних осуществляется кластеризация абзацев (разбиение на тематические разделы) и предлагается новый алгоритм определения количества кластеров, основанный на минимизации некоторой целевой функции. Наконец, в третьей фазе из каждого тематического раздела извлекается по одному предложению для включения в резюме. Анализ показывает, что резюме, созданное по этому алгоритму, не может охватывать главный контент документа. Это связано с тем, что нет четкого определения количества кластеров. В работе [10] кластеризация предложений реализуется методом иерархической кластеризации, который с вычислительной точки зрения более сложен, чем метод k-средних.

ISSN 0572-2691

В настоящей работе с целью обеспечения минимальной избыточности в резюме и максимально возможной степени охвата контента документа предлагается новый метод кластеризации предложений, основанный на решении задачи целочисленного квадратичного программирования с булевыми переменными. Этот метод, в отличие от других, например методов *k*-средних и иерархических, позволяет сбалансировать гомогенность и гетерогенность кластеров. Известно, что решение задач целочисленного квадратичного программирования требует больших вычислительных и временных ресурсов. Поэтому для решения таких задач целесообразно использовать эвристические методы — генетические алгоритмы, нейронные сети, муравьиные алгоритмы. Для решения задачи целочисленного программирования, соответствующей рассматриваемому случаю, разработан алгоритм синтеза нейронной сети. Определение количества кластеров — одна из сложных задач кластерного анализа. В данной работе предлагается также алгоритм пошагового определения количества кластеров. После кластеризации, во избежание избыточности в резюме, в каждом кластере, т.е. в каждом тематическом разделе, определяются информативные предложения и их количество.

1. Задача кластеризации предложений

В процессе интеллектуального анализа данных (data mining) кластеризация является одним из самых полезных подходов для обнаружения естественных групп в наборе данных. Для решения задачи кластеризации обычно используются традиционные алгоритмы, такие, как алгоритм *k*-средних, иерархическая кластеризация, алгоритм GEM (Gaussian Expectation-Maximization) и т.д. [13–16]. Из них широко распространение получил алгоритм *k*-средних. Это обусловлено тем, что этот алгоритм математически хорошо формулируется. Формулировка алгоритма *k*-средних как задачи математического программирования предложена в работах [17, 18]. В [19] алгоритм *k*-средних сформулирован в терминах негладкой и невыпуклой оптимизации. В настоящей работе для решения задач кластеризации применяются не традиционные методы, а методика, предложенная в [20].

Пусть документ d состоит из m предложений; представим его в виде набора предложений $d = (s_1, s_2, \ldots, s_m)$. Идея задачи кластеризации состоит в разбиении множества $d = (s_1, s_2, \ldots, s_m)$ на непересекающиеся кластеры $\mathcal{C} = (C_1, C_2, \ldots, C_q)$, $q \geq 2$, с целью обеспечения максимальной близости между предложениями одного кластера, соответствующими определенной смысловой тематике, и максимального различия между кластерами. Понятие «близость» определяется ниже.

Прежде чем перейти к формулированию метода с помощью модели векторного пространства, каждое предложение представим в виде взвешенного вектора слов $s_i = (w_{i1}, w_{i2}, \ldots, w_{in})$, которые появляются в документе, n — количество слов в документе d.

Вес w_{ij} слова j зависит от частоты его появления в конкретном предложении i и во всем наборе предложений (в документе) и определяется формулой TF * IDF:

$$w_{ij} = f_{ij} \log_2\left(\frac{m}{m_j}\right); \quad i = 1,\ldots, m; \quad j = 1,\ldots, n,$$

где m_j — количество предложений, в которых присутствует слово j.

Функция f_{ij} частоты появления слова j в предложении i вычисляется следующим образом:

$$f_{ij} = \frac{n_{ij}}{n \operatorname{len}(s_i)},$$

здесь n_{ij} — количество появлений слова j в предложении i, $\operatorname{len}(s_i)$ — длина предложения s_i. Во избежание смещения, вызванного длиной (количеством слов) предложения, функция f_{ij} нормализована относительно длины предложения.

Отметим, что прежде чем вычислить вес слова в документе, необходимо выполнить следующие операции, цель которых уменьшить размерность задачи. Во-первых, в работу алгоритма заложено отбрасывание, т.е. исключение из документа определенных слов, которые часто называют «стоп-словами». Такими словами могут быть предлоги, суффиксы, причастия, междометия и частицы. Во-вторых, следует выделить корни слов.

В нашем случае предполагается, что текст написан на английском языке. Поэтому для удаления стоп-слов используем их список, предложенный в [21], а для выделения корня слов применяем алгоритм, разработанный Портером [22].

Для определения близости d_{ip} между предложениями s_i и s_p обычно используется евклидово расстояние:

$$d_{ip} = \sqrt{\sum_{j=1}^{n} (w_{ij} - w_{pj})^2}; \quad i, p = 1, \ldots, m.$$

2. Сведение задачи кластеризации к целочисленному квадратичному программированию

Известно, что качество кластеризации оценивается с точки зрения гомогенности (точки одного кластера должны быть близки) и гетерогенности (точки разных кластеров должны быть удалены друг от друга). Большинство методов кластеризации, которые также использованы в работах [8, 10], обеспечивают или гомогенность, или гетерогенность кластеров. Например, принцип работы алгоритма k-средних основывается на обеспечении гомогенности кластеров. Предложенный в данной статье метод кластеризации не только обеспечивает максимальную близость точек в кластерах (гомогенность), он еще и гарантирует, что сгруппированные в разные кластеры точки в максимально возможной степени будут отдалены друг от друга (гетерогенность).

Близость предложений в кластерах и удаленность предложений, отнесенных к разным кластерам, означает, что общая сумма расстояний между предложениями в пределах кластера должна быть минимальной, а общая сумма расстояний между предложениями, отнесенными к разным кластерам, — максимальной. Поэтому определим сумму S_k расстояний d_{ip} между предложениями s_i и s_p в кластере C_k:

$$S_k = \frac{1}{2} \sum_{s_i \in C_k} \sum_{s_p \in C_k} d_{ip}; \quad k = 1, \ldots, q; \quad i, p = 1, \ldots, m.$$

Просуммировав по k, получаем общую сумму расстояний между предложениями во всех кластерах C_k, $k = 1, \ldots, q$:

$$\sum_{k=1}^{q} S_k = \frac{1}{2} \sum_{k=1}^{q} \sum_{s_i \in C_k} \sum_{s_p \in C_k} d_{ip}; \quad i, p = 1, \ldots, m. \qquad (1)$$

ISSN 0572-2691

Теперь определим сумму S_{kl} расстояний d_{ip} между предложениями s_i и s_p, отнесенными к разным кластерам C_k и C_l, $k \neq l$:

$$S_{kl} = \frac{1}{2} \sum_{s_i \in C_k} \sum_{s_p \in C_l} d_{ip}; \quad k, l = 1, \ldots, q; \quad i, p = 1, \ldots, m.$$

Суммируя по k и l, $k \neq l$, находим общую сумму расстояний между предложениями, отнесенными к разным кластерам:

$$\sum_{k=1}^{q} \sum_{\substack{l=1 \\ l \neq k}}^{q} S_{kl} = \frac{1}{2} \sum_{k=1}^{q} \sum_{\substack{l=1 \\ l \neq k}}^{q} \sum_{s_i \in C_k} \sum_{s_p \in C_l} d_{ip}; \quad i, p = 1, \ldots, m. \tag{2}$$

На основании формул (1) и (2) сведем задачу кластеризации к такому виду, чтобы она одновременно обеспечивала и гомогенность, и гетерогенность кластеров:

$$\sum_{k=1}^{q} \sum_{s_i \in C_k} \sum_{s_p \in C_k} d_{ip} - \sum_{k=1}^{q} \sum_{\substack{l=1 \\ l \neq k}}^{q} \sum_{s_i \in C_k} \sum_{s_p \in C_l} d_{ip} \to \min. \tag{3}$$

Введем булеву переменную x_{ik}:

$$x_{ik} = \begin{cases} 1, & \text{если } s_i \in C_k, \\ 0, & \text{если } s_i \notin C_k, \end{cases} \quad i = 1, \ldots, m; \quad k = 1, \ldots, q.$$

С учетом этого обозначения формулу (3) записываем в следующем виде:

$$\sum_{k=1}^{q} \sum_{i=1}^{m} \sum_{p=1}^{m} d_{ip} x_{ik} x_{pk} - \sum_{i=1}^{m} \sum_{k=1}^{q} \sum_{p=1}^{m} \sum_{\substack{l=1 \\ l \neq k}}^{q} d_{ip} x_{ik} x_{pl} \to \min. \tag{4}$$

Введем обозначение

$$d_{ip} e_{kl} = a_{ikpl},$$

где

$$e_{kl} = \begin{cases} 1, \text{ если } k = l, \\ -1, \text{ если } k \neq l. \end{cases}$$

Задачу (4) перепишем в компактном виде:

$$\sum_{i=1}^{m} \sum_{k=1}^{q} \sum_{p=1}^{m} \sum_{l=1}^{q} a_{ikpl} x_{ik} x_{pl} \to \min. \tag{5}$$

Из предположения $C_k \cap C_l = \varnothing$, $k \neq l$ (т.е. каждое из m предложений относится только к одному из q кластеров) следует, что должно выполняться условие

$$\sum_{k=1}^{q} x_{ik} = 1, \quad i = 1, \ldots, m. \tag{6}$$

С другой стороны, предполагается, что каждый кластер содержит хотя бы одно предложение

$$\sum_{i=1}^{m} x_{ik} \geq 1, \quad k = 1,\ldots, q, \tag{7}$$

где

$$x_{ik} \in \{0, 1\} \quad \forall i, k. \tag{8}$$

Итак, задача кластеризации предложений сведена к задаче целочисленного квадратичного программирования с булевыми переменными (5)–(8).

Задача (5)–(8) относится к задачам комбинаторной оптимизации. Многие из таких задач *NP*-полные, и их решение связано со слишком большими временными затратами. Поэтому для решения данного типа задач представляется целесообразным использовать нейронные сети, которые нашли эффективное применение в задачах комбинаторной оптимизации [20, 23].

3. Нейросетевая реализация задачи целочисленного квадратичного программирования

Использование нейронных сетей позволяет существенно сократить время решения задач (и в еще большей степени *NP*-полных задач). Поскольку метод нейронных сетей относятся к эвристическим, то их применение в общем случае не гарантирует глобальности найденного решения. Однако на практике зачастую требуется за определенное время найти один или несколько локальных экстремумов. В таком случае использование эвристических методов, в частности нейронных сетей, очень эффективно. Следует отметить, что решение конкретной задачи оптимизации на нейронной сети требует ее синтеза.

Для решения задачи оптимизации синтезируется тройка вида $\{\boldsymbol{N}, \boldsymbol{W}, \boldsymbol{B}\}$, где $\boldsymbol{N}$ — множество нейронов сети, $\boldsymbol{W}$ — матрица синаптических связей и $\boldsymbol{B}$ — вектор внешних смещений. Задача синтеза сети в общем случае состоит в определении всех компонентов данной тройки: вида и количества нейронов, значений внешних смещений, структуры матрицы связей и значений ее элементов. Считается, что тип и модель динамики нейроподобных элементов определены. Поэтому задача синтеза сети сводится к определению структуры сети, матрицы связей $\boldsymbol{W}$ и вектора смещений $\boldsymbol{B}$, удовлетворяющих целевому использованию синтезируемой сети.

Синтез нейронной сети для решения задачи оптимизации состоит из следующих этапов.

Этап 1. Нейросетевая интерпретация задачи.

Для нейросетевой интерпретации задачи оптимизации рассмотрим сеть бинарных нейронов, представляющих собой матрицу $\boldsymbol{Y} = \| y_{ik} \|$ размерностью $m \times q$. Каждой булевой переменной x_{ik} ставится в соответствие выходной сигнал y_{ik} ik-го нейрона. Возбужденное состояние нейрона $y_{ik} = 1$ в такой матрице соответствует тому факту, что предложение i отнесено к кластеру k.

Этап 2. Конструирование энергетической функции сети.

Второй этап процесса построения оптимизируемой сети заключается в конструировании энергетической функции сети. Построим эту функцию в виде суммы, отдельные слагаемые которой представляют собой выпуклые функции, принимающие минимальные значения на состояниях сети, удовлетворяющих рассмотренным ограничениям на состояниях сети и минимизирующих целевую функцию.

ISSN 0572-2691

Таким образом, слагаемое, обеспечивающее задачу минимизации (5), можно конструировать в виде

$$E_0 = -\frac{\lambda_0}{2}\sum_{i=1}^{m}\sum_{k=1}^{q}\sum_{p=1}^{m}\sum_{l=1}^{q} a_{ikpl} y_{ik} y_{pl}; \tag{9}$$

слагаемые, обеспечивающие выполнение ограничений (6)–(8), — в виде

$$E_1 = \frac{\lambda_1}{2}\sum_{i=1}^{m}\sum_{k=1}^{q} y_{ik}(1-y_{ik}) + \frac{\lambda_2}{2}\sum_{i=1}^{m}\left(\sum_{k=1}^{q} y_{ik} - 1\right)^2 +$$

$$+\frac{\lambda_3}{2}\left(\sum_{i=1}^{m}\sum_{k=1}^{q} y_{ik} - m\right)^2 + \frac{\lambda_4}{2}\sum_{k=1}^{q}\varphi^2\left(\sum_{i=1}^{m} y_{ik} - 1\right), \tag{10}$$

где $\lambda_0 \div \lambda_4$ — положительные константы, $\varphi(z) = z - |z|$ — функция, обладающая свойством $\varphi^2(z) = 2z\varphi(z)$.

Первое слагаемое в (10) соответствует бинарности переменных (8); второе — ограничению (6), где каждая строка матрицы $\boldsymbol{Y}$ содержит не более одной единицы; третье слагаемое соответствует тому, что в матрице $\boldsymbol{Y}$ содержится ровно m единиц; последнее слагаемое — ограничению (7). Следовательно, при ограничениях (6)–(8) выражение (10) принимает свое минимальное, равное нулю значение.

После суммирования выражений (9) и (10) и несложных преобразований получаем следующий вид энергетической функции нейронной сети:

$$E = E_0 + E_1 = -\frac{\lambda_0}{2}\sum_{i=1}^{m}\sum_{k=1}^{q}\sum_{p=1}^{m}\sum_{l=1}^{q} a_{ikpl} y_{ik} y_{pl} - \frac{\lambda_1}{2}\sum_{i=1}^{m}\sum_{k=1}^{q}\sum_{p=1}^{m}\sum_{l=1}^{q}\delta_{ip}\delta_{kl} y_{ik} y_{pl} +$$

$$+\frac{\lambda_1}{2}\sum_{i=1}^{m}\sum_{k=1}^{q} y_{ik} + \frac{\lambda_2}{2}\sum_{i=1}^{m}\sum_{k=1}^{q}\sum_{p=1}^{m}\sum_{l=1}^{q}\delta_{ip} y_{ik} y_{pl} - \lambda_2\sum_{i=1}^{m}\sum_{k=1}^{q} y_{ik} + \frac{\lambda_2}{2}m +$$

$$+\frac{\lambda_3}{2}\sum_{i=1}^{m}\sum_{k=1}^{q}\sum_{p=1}^{m}\sum_{l=1}^{q} y_{ik} y_{pl} - m\lambda_3\sum_{i=1}^{m}\sum_{k=1}^{q} y_{ik} +$$

$$+\frac{\lambda_3}{2}m^2 + \lambda_4\sum_{k=1}^{q}\left(\sum_{i=1}^{m} y_{ik} - 1\right)\varphi\left(\sum_{i=1}^{m} y_{ik} - 1\right), \tag{11}$$

δ_{ip} — символ Кронекера.

Этап 3. Определение параметров сети.

На этом этапе непосредственно определяются параметры нейронной сети — матрицы синаптических связей $\boldsymbol{W}$ и вектора внешних смещений $\boldsymbol{B}$ — путем сопоставления сконструированной энергетической функции $\boldsymbol{E}$ с ее канонической формой $\boldsymbol{E}_c$, которая конструируется в виде

$$E_c = -\frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{q}\sum_{p=1}^{m}\sum_{l=1}^{q} w_{ikpl} y_{ik} y_{pl} + \sum_{i=1}^{m}\sum_{k=1}^{q} b_{ik} y_{ik}. \tag{12}$$

Сопоставив выражения (11) и (12) и приравняв их линейные и квадратичные составляющие, находим параметры нейронной сети:

$$w_{ikpl} = \lambda_0 a_{ikpl} + \lambda_1 \delta_{ip}\delta_{kl} - \lambda_2 \delta_{ip} - \lambda_3,$$
$$b_{ik} = \frac{\lambda_1}{2} - \lambda_2 - m\lambda_3 + \lambda_4, \tag{13}$$

где $i, p = 1,\ldots, m;\ k, l = 1,\ldots, q.$

Отметим, что при определении параметров (13) слагаемые в (11), которые не зависят от состояния y_{ik} нейронной сети, не учитывались.

Таким образом, построена нейронная сеть, параметры которой определены с точностью до постоянных коэффициентов. Вопрос определения коэффициентов $\lambda_0 \div \lambda_4$ требует отдельного исследования.

4. Алгоритм определения количества кластеров

Отметим, что выбор оптимального количества кластеров — важный этап кластерного анализа [19, 24–26]. Априорно трудно определить, сколько кластеров представляет рассматриваемое множество. Здесь предлагается следующая стратегия: начиная с достаточно малого количества кластеров q его следует поэтапно увеличивать до тех пор, пока некоторый критерий завершения не будет удовлетворен. С точки зрения перспективы оптимизации это означает, что если решение соответствующей задачи оптимизации (5)–(8) не удовлетворительно, то должна рассматриваться задача (5)–(8) с $q+1$ кластером, и т.д. Таким образом, задача (5)–(8) должна решаться неоднократно с различным количеством кластеров.

Приведем алгоритм пошагового вычисления кластеров.

Введем функцию $F(x)$, которая определяется соотношением

$$F(x) = \frac{\sum\limits_{k=1}^{q}\sum\limits_{i=1}^{m}\sum\limits_{p=1}^{m} d_{ip}x_{ik}x_{pk}}{\sum\limits_{i=1}^{m}\sum\limits_{k=1}^{q}\sum\limits_{p=1}^{m}\sum\limits_{\substack{l=1\\ l\neq k}}^{q} d_{ip}x_{ik}x_{pl}},$$

где числитель соответствует первому, а знаменатель — второму слагаемому в формуле (4).

Шаг 1. Задается допуск $\varepsilon > 0$. Полагаем $k = 2$ и решаем задачу (5)–(8). Пусть F_2 — значение функции $F(x)$, соответствующее решению задачи (5)–(8).

Шаг 2. Полагаем $k = k+1$ и решаем задачу (5)–(8). Пусть F_{k+1} — значение функции $F(x)$, соответствующее решению задачи (5)–(8).

Шаг 3. Если $(F_k - F_{k+1})/F_2 < \varepsilon$, $k \geq 2$, то следует остановить алгоритм, в противном случае полагаем $k = k+1$ и переходим к шагу 2.

Легко показать, что для всех k выполняется условие $F_k \geq F_{k+1} > 0$. Таким образом, в результате получаем убывающую последовательность $\{F_k\}$, $F_k > 0$, $\forall k$. Следовательно, после k^* итераций критерий останова в шаге 3 будет удовлетворен.

Выбор допуска $\varepsilon > 0$ очень существенен в описанном алгоритме. Большое значение ε может привести к крупным кластерам, объединяющим другие кластеры, а малое — к появлению небольших искусственных кластеров. Выбор допуска ε определяется экспериментальным путем.

5. Выбор информативных предложений и оценка резюмирования

Следующий шаг после кластеризации — определение информативных предложений в каждом кластере. Информативность предложения определяется мерой близости, вычисляемой между ним и соответствующим кластерным центроидом, т.е. чем меньше евклидово расстояние между предложением и соответствующим кластерным центроидом, тем это предложение более информативно. Перед включением предложений в резюме они ранжируются в порядке возрастания их мер близости к соответствующему кластерному центроиду. Большинство текстовых документов обычно состоят из нескольких тем. Некоторые темы описываются многими предложениями и, следовательно, формируют главный контент документа. Другие темы могут только кратко упоминаться, чтобы дополнить главную тематику. Следовательно, количество предложений в разных кластерах различно. При этом количество предложений, выбираемых из разных кластеров, тоже различно. Такой подход позволяет в максимально возможной степени охватывать главный контент документа и избегать избыточности. В общем случае количество предложений, включаемых в резюме, зависит от коэффициента сжатия. Коэффициент сжатия a_{cmpr} определяется отношением длин резюме и документа

$$a_{\text{cmpr}} = \frac{\text{len}(\text{summ})}{\text{len}(\text{doc})}$$

и служит важным фактором, влияющим на качество резюме; здесь $\text{len}(\text{summ})$, $\text{len}(\text{doc})$ — длины резюме и документа соответственно. При малом значении коэффициента сжатия резюме более краткое и основная часть информации утрачивается, а при большом — резюме более подробное, однако оно содержит несущественные предложения. В работе [1] показано, что если коэффициент сжатия находится в интервале $[0{,}05;\ 0{,}3]$, то результат резюмирования считается приемлемым.

На основании изложенного определим количество N_k информативных предложений, отобранных из каждого кластера k, которое вычисляется формулой

$$N_k = \left[\frac{\text{len}(C_k)\, a_{\text{cmpr}}}{\text{len}_{\text{avg}}} \right], \quad k = 1, \ldots, q,$$

где $\text{len}(C_k)$ — длина кластера C_k, $\text{len}_{\text{avg}} = \text{len}(\text{doc})/m$ — средняя длина предложений в документе, $[\cdot]$ означает целую часть числа.

Для оценки результата резюмирования используется F_1-критерий. Пусть N_d^{rel} — количество релевантных предложений в документе, N_s^{rel} — количество релевантных предложений в резюме, N_s — общее количество предложений в резюме, P — точность, R — полнота. Тогда

$$P = \frac{N_s^{\text{rel}}}{N_s}, \quad R = \frac{N_s^{\text{rel}}}{N_d^{\text{rel}}}, \quad F_1 = \frac{2PR}{P+R}.$$

Заключение

В связи с увеличением количества текстовых информаций в Интернете возникает потребность в автоматических методах резюмирования. Цель задачи автоматического резюмирования заключается в извлечении из текста нескольких обобщающих фрагментов (предложений, абзацев), отражающих его контент. Если документ состоит из нескольких тематических разделов, то проблема создания ре-

зюме, охватывающего все темы документа, становится трудной. Для решения проблемы предлагается подход, позволяющий выявить тематические разделы и информативные предложения в документе. Этот подход основан на кластеризации предложений, которая сведена к задаче целочисленного квадратичного программирования с булевыми переменными. При резюмировании одна из сложных задач состоит в выявлении тематических разделов в документе, что непосредственно связано с определением количества кластеров. Количество кластеров в традиционных алгоритмах в основном задается заранее, и во многих случаях это не практично. Например, количество тематических разделов в документе заранее не известно. В работе [8] сделана попытка определения количества кластеров. Анализ показывает, что такое определение не может гарантировать оптимальность найденного количества кластеров. Для решения этой проблемы предлагается новый алгоритм, позволяющий найти оптимальное количество кластеров, т.е. латентных тематических разделов в документе. Преимущество этого алгоритма в том, что определение количества кластеров непосредственно связывается с целевой функцией, обеспечивающей точную кластеризацию. Последний шаг при резюмировании — определение информативных предложений и их количества. Для широкого охвата контента документа и устранения избыточности в работе предложены критерий информативности и алгоритм определения количества предложений для включения в резюме. Количество выбранных из каждого кластера предложений регулируется заранее задаваемым параметром a_{cmpr}.

Р.М. Алгулієв, Р.М. Алигулієв

АВТОМАТИЧНЕ РЕЗЮМУВАННЯ ТЕКСТОВИХ ДОКУМЕНТІВ ЧЕРЕЗ КЛАСТЕРИЗАЦІЮ РЕЧЕНЬ

Для забезпечення мінімальної надмірності в резюме і максимально можливого відображення контенту документа запропоновано метод автоматичного резюмування текстових документів, який базується на кластеризації речень. Кластеризація речень застосовується з метою визначення тематичних розділів та інформативних речень. Кількість кластерів (тематичних розділів) визначається за допомогою спеціально розробленого алгоритму. Описано алгоритм синтезу нейронної мережі для розв'язання задачі кластеризації.

R.M. Alguliev, R.M. Alyguliev

AUTOMATIC TEXT DOCUMENTS SUMMARIZATION THROUGH SENTENCES CLUSTERING

The generic document summarization method based on sentences clustering is proposed. The proposed approach allows revealing topical sections in a document which is one of difficult problems in the document summarization. The clustering problem is formulated as a binary quadratic integer programming problem. For solving the binary quadratic integer programming problem the synthesis algorithm of a neural network is described.

1. *Mani I., Maybury M.T.* Advances in automated text summarization. — Cambridge : MIT Press, 1999. — 442 p.
2. *Salton G., Singhal A., Mitra M., Buckley C.* Automatic text structuring and summarization // Information Processing and Management. — 1997. — **33**, N 2. — P. 193–207.
3. *Mitra M., Singhal A., Buckley C.* Automatic text summarization by paragraph extraction // Proc. of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. Madrid, Spain. July 7–12, 1997. — P. 39–46.

ISSN 0572-2691

4. *Kruengkrai C., Jaruskulchai C.* Generic text summarization using local and global properties of sentences // Proc. of the IEEE/WIC Int. Conf. on Web Intelligence (WI'03). Halifax, Canada. October 13–17, 2003. — P. 201–206.
5. *Yeh J-Y., Ke H-R., Yang W-P., Meng I-H.* Text summarization using a trainable summarizer and latent semantic analysis // Information Processing and Management. — 2005. — **41**, N 1. — P. 75–95.
6. *Goldstein J., Kantrowitz M., Mittal V., Carbonell J.* Summarization text documents: sentence selection and evaluation metrics // Proc. of the 22-d Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley, USA. August 15–19, 1999. — P. 121–128.
7. *Gong Y., Liu X.* Generic text summarization using relevance measure and latent semantic analysis // Proc. of the 24-th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. New Orleans, USA. September 9–12, 2001. — P. 19–25.
8. *Hu P., He T., Ji D., Wang M.* A study of Chinese text summarization using adaptive clustering of paragraphs // Proc. of the 4-th Int. Conf. on Computer and Information Technology (CIT'04). Wuhan, China. September 14–16, 2004. — P. 1159–1164.
9. *Web-page* classification through summarization / D. Shen, Z. Chen, Q. Yang, H.J. Zeng, B. Zhang, Y. Lu, W.Y. Ma // Proc. of the 27-th Annual Int. Conf. on Research and Development in Information Retrieval. Sheffield, UK. July 25–29, 2004. — P. 242–249.
10. *Delort J.-Y., Bouchon-Meunier B., Rifqi M.* Enhanced web document summarization using hyperlinks // Proc. of the 14-th ACM Conf. on Hypertext and Hypermedia. Nottingham, UK. August 26–30, 2003. — P. 208–215.
11. *Luhn H.P.* The automatic creation of literature abstracts // IBM J. of Research and Development. — 1958. — **2**, N 2. — P. 159–165.
12. *Banko M., Mittal V., Kantrowitz M., Goldstein J.* Generating extraction-based summaries from hand-written summaries by aligning text spans // Proc. of the 4-th Conf. of the Pacific Association for Computational Linguistics (PACLING'99). Waterloo, Canada. August 25–28, 1999. — P. 36–40.
13. *Grabmeier J., Rudolph A.* Techniques of cluster algorithms in data mining // Data Mining and Knowledge Discovery. — 2002. — **6**, N 4. — P. 303–360.
14. *Halkidi M., Batistakis Y., Vazirgiannis M.* On clustering validation techniques // J. of Intelligent Systems. — 2001. — **17**, N 2–3. — P. 107–145.
15. *Jain A.K., Murty M.N., Flynn P.J.* Data clustering : A review // ACM Computing Surveys. — 1999. — **31**, N 3. — P. 264–323.
16. *Jiang D., Tang C., Zhang A.* Cluster analysis for gene expression data: a survey // IEEE Transactions on Knowledge and Data Engineering. — 2004. — **16**, N 11. — P. 1370–1386.
17. *Mangasarian O.L.* Mathematical programming in data mining // Data Mining and Knowledge Discovery. — 1997. — **1**, N 2. — P. 183–201.
18. *Bradley P.S., Fayyad U.M., Mangasarian O.L.* Mathematical programming for data mining: formulations and challenges // INFORMS J. on Computing. — 1999. — **11**, N 3. — P. 217–238.
19. *New* algorithms for multi-class diagnosis using tumor gene expression signature / A.M. Bagirov, B. Ferguson, S. Ivkovic, G. Saunders, J. Yearwood // Bioinformatics. — 2003. — **19**, N 14. — P. 1800–1807.
20. *Алгулиев Р.М., Алыгулиев Р.М., Алекперов Р.К.* Подход к оптимальному назначению заданий в распределенной системе // Проблемы управления и информатики. — 2004. — № 5. — С. 140–145.
21. ftp://ftp.cs.cornell.edu/pub/smart/english.stop.
22. *Porter M.* An algorithm for suffix stripping // Program. — 1980. — **14**, N 3. — P. 130–137.
23. *Нейроматематика.* Кн. 6. Уч. пособие для вузов / Под общей редакцией А.И. Галушкина. — М. : ИПРЖР, 2002. — 448 с.
24. *Kim D.-W., Lee K.H., Lee D.* On cluster validity index for estimation of the optimal number of fuzzy clusters // Pattern Recognition. — 2004. — **37**, N 10. — P. 2009–2025.
25. *Kothari R., Pitts D.* On finding the number of clusters // Pattern Recognition Letters. — 1999. — **20**, N 4. — P. 405–416.
26. *Sun H., Wang S., Jiang Q.* FCM-based model selection algorithms for determining the number of clusters // Pattern Recognition. — 2004. — **37**, N 10. — P. 2027–2037.

Получено 29.06.2005
После доработки 06.03.2008