УДК 004.089 DOI 10.18522/2311-3103-2022-4-122-143

В.В. Бова, Ю.А. Кравченко, С.И. Родзин

МЕТОДЫ И АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ ТЕКСТОВЫХ ДАННЫХ (ОБЗОР)

Рассматривается одна из важных задач искусственного интеллекта – машинная обработка естественного языка. Решение данной задачи на основе кластерного анализа позволяет выявлять, формализовывать и интегрировать большие объемы лингвистической экспертной информации в условиях информационной неопределенности и слабой структурированности исходных текстовых ресурсов, полученных из различных предметных областей. Кластерный анализ является мощным средством разведочного анализа текстовых данных, позволяющий провести объективную классификацию любых объектов, которые охарактеризованы рядом признаков и имеют скрытые закономерности. Проведен обзор и анализ современных модифицированных алгоритмов агломеративной кластеризации CURE, ROCK, CHAMELEON, неиерархической кластеризации PAM, CLARA и алгоритма аффинного преобразования, используемых на различных этапах кластеризации текстовых данных, эффективность которых проверяется экспериментальными исследованиями. В работе обоснованы требования к выбору наиболее эффективного метода кластеризации для решения задачи повышения эффективности интеллектуальной обработки лингвистической экспертной информации. Также в работе рассмотрены способы визуализации результатов кластеризации для интерпретации кластерной структуры и зависимостей на множестве элементов текстовых данных и графические средства их представления в виде дендограмм, диаграмм рассеивания, диаграмм сходства VOS и карт интенсивности. Для сравнения качества работы алгоритмов использовались внутренние и внешние метрики эффективности: «V-мера», «Adjusted Rand index», «Силуэт». На основании проведенных экспериментов выявлено, что необходимо использовать гибридный подход, в котором для первоначального выбора числа кластеров и распределения их центров использовать иерархический подход, основанный на последовательном объединении и максимизации близости данных ограниченной выборки, когда нет возможности выдвинуть гипотезу о начальном количестве кластеров. Далее подключать алгоритмы итерационной кластеризации, обеспечивающие высокую устойчивость по отношению к шумовым признакам и наличию выбросов. За счет гибридизации повышается эффективность работы алгоритмов кластеризации. Результаты исследований показали, что для повышения вычислительной эффективности и преодоления чувствительности при инициализации параметров алгоритмов кластеризации для оптимизации параметров модели обучения и поиска глобального оптимального решения необходимо использовать метаэвристические подходы.

Кластерный анализ текстовых данных; агломеративная кластеризация; метрики качества; неиерархическая кластеризация; метод аффинного преобразования; дендограммы; диаграммы рассеивания.

V.V. Bova, Y.A. Kravchenko, S.I. Rodzin

METHODS AND ALGORITHMS FOR TEXT DATA CLUSTERING (REVIEW)

The article deals with one of the important tasks of artificial intelligence – machine processing of natural language. The solution of this problem based on cluster analysis makes it possible to identify, formalize and integrate large amounts of linguistic expert information under conditions of information uncertainty and weak structure of the original text resources obtained from various subject areas. Cluster analysis is a powerful tool for exploratory analysis of text data, which allows for an objective classification of any objects that are characterized by a number of features and have hidden patterns. A review and analysis of modern modified algorithms for agglomerative clustering CURE, ROCK, CHAMELEON, non-hierarchical clustering PAM, CLARA and the affine transformation algorithm used at various stages of text data clustering, the effectiveness of which is verified by experimental studies, is carried out. The paper substantiates the requirements for choosing the most efficient clustering method for solving the problem of increas-

ing the efficiency of intellectual processing of linguistic expert information. Also, the paper considers methods for visualizing clustering results for interpreting the cluster structure and dependencies on a set of text data elements and graphical means of their presentation in the form of dendograms, scatterplots, VOS similarity diagrams, and intensity maps. To compare the quality of the algorithms, internal and external performance metrics were used: "V-measure", "Adjusted Rand index", "Silhouette". Based on the experiments, it was found that it is necessary to use a hybrid approach, in which, for the initial selection of the number of clusters and the distribution of their centers, use a hierarchical approach based on sequential combining and averaging the characteristics of the closest data of a limited sample, when it is not possible to put forward a hypothesis about the initial number of clusters. Next, connect iterative clustering algorithms that provide high stability with respect to noise features and the presence of outliers. Hybridization increases the efficiency of clustering algorithms. The research results showed that in order to increase the computational efficiency and overcome the sensitivity when initializing the parameters of clustering algorithms, it is necessary to use metaheuristic approaches to optimize the parameters of the learning model and search for a global optimal solution.

Text data cluster analysis; agglomerative clustering; quality metrics; non-hierarchical clustering; affine transformation method; dendograms; scatterplots.

Введение. Одним из актуальных направлений в области машинной обработки естественного языка (natural language processing (NLP) является кластерный анализ слабоструктурированных естественно-языковых (ЕЯ) текстовых данных, позволяющий проводить первоначальную обработку информации для ее структурирования, выделения семантических признаков, обобщения и сортировки.

Важной задачей кластерного анализа текстовых данных является поиск скрытых закономерностей в больших объемах информации. При ее решении возникает проблема извлечения и отбора признаков существенных для анализа лингвистической информации [1–3]. Необходимость выбора большого количества семантических признаков многомерных текстовых данных приводит к увеличению вычислительной и временной сложности алгоритма кластеризации. Но при этом исключение из пространства признаков шумовых объектов, может приводить к потере значимой для идентификации неявных закономерностей информации [4–6]. Определение связи шумовых признаков с целевой переменной в решаемой задаче обработки ЕЯ текстовых данных является фундаментальной проблемой алгоритмов кластеризации. Поэтому при выборе алгоритма необходимо основываться на максимальном сохранении информационных признаков, используемых для кластеризации.

Анализ научных исследований в области кластерного анализа текстовых данных [2–8] показывает, что в настоящее время присутствует существенное многообразие алгоритмов кластеризации, отличных по своей природе и происхождению, каждым из них можно получить различные разбиения исходного множества текстовых данных. По этой причине актуальной проблемой остается выбор и применение эффективного метода кластеризации для решения задач NLP. Для решения данных проблем можно сформулировать ряд требований, которым должен удовлетворять метод кластеризации текстовых данных: обеспечение высокой размерности пространства признаков данных; масштабируемость при работе с большим объемом текстовых данных; обеспечение смешанного типа измерений при выборе метода сходства (вычисления семантической близости) и метода объединения (агрегации связей на основе поиска скрытых закономерностей); выявление семантически однородных кластеров, содержащих объекты близкие по совокупности свойств и признаков; распределение объектов по всей совокупности взаимоисключающих кластеров.

В данной работе проводится сравнительный анализ современных модификаций известных алгоритмов кластеризации текстовых данных, наиболее подходящих для решения задачи повышения эффективности интеллектуальной обработки лингвистической экспертной информации.

1. Особенности выбора и классификация методов кластерного анализа текстовой информации. Построение моделей кластерного анализа данных, представленных на ЕЯ основывается на семантическом подходе. Семантическая кластеризация является многоэтапной процедурой, на каждом этапе которой должна решаться отдельная задача выбора наиболее эффективного алгоритма извлечения признаков и уменьшения размерности данных [3, 9–11]. Последовательность ключевых этапов кластерного анализа представлена на рис. 1. На первом этапе предварительной обработки текста происходит преобразование текстовых документов в структурированные данные на основе извлечения признаков, которые представлены в онтологии определенной предметной области, удаление избыточной информации и уменьшение размерности пространства признаков [4, 10, 11]. Для предварительной обработки применяются методы фильтрации – удаления спецсимволов и общеупотребительных слов, токкенизации – разбиения текста на лингвистические тематические единицы, стемминг, удаление стоп-слов, сокращение низкочастотных терминов [1, 9].

На втором этапе реализуются задачи выбора признаков, их скрытое семантическое индексирование и преобразование в векторную модель данных, которые можно подавать на вход алгоритмам машинного обучения. Подробный обзор методик векторизации текстовых документов приведен в работах [12–14].

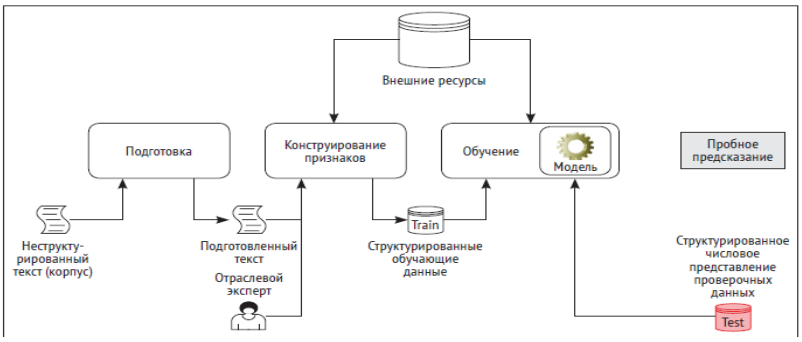

Рис. 1. Этапы кластерного анализа

Векторное представление модели текстовых данных характеризуется высокой размерностью и разреженностью семантических признаков, что требует применения улучшенных алгоритмов кластеризации для обработки данных большой размерности [4, 13].

В последнее время ведутся активные разработки новых масштабируемых алгоритмов кластеризации, способных обрабатывать текстовые данные большой размерности [3, 14–19]. Разработанные авторами модели кластерного анализа представляют собой гибридное решение на основе известных алгоритмов и эффективно решают задачи машинного перевода, распознавания речи и генерации реалистичных текстов. В работах [8, 9] представлена современная классификация модифицированных алгоритмов кластеризации. На рис. 2 приведена классификация основных типов репрезентативных алгоритмов кластеризации текстовых данных на основе иерархических и итерационных (неиерархических) методов.

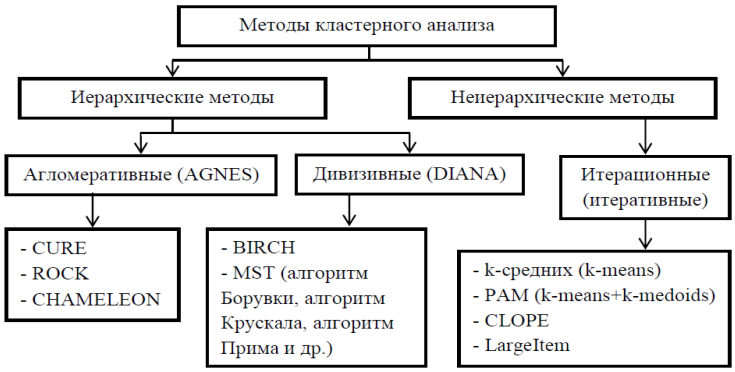

Рис. 2. Классификация методов кластерного анализа

Выбор определенного метода зависит от набора данных, к которому он применяется, требований к точности и скорости работы алгоритма, критериям оптимальности параметров кластеризации при решении задач NLP [19]. Также уточнения и обоснования требуют следующие вопросы:

♦- выбор числа кластеров и метода оптимизации параметров их первоначального формирования;

♦- применение на различных этапах кластеризации эффективных методов измерения сходства и вычисления семантической близости связанных текстовых данных в многомерном пространстве их информационных признаков;

♦- выбор метрик точности и качества кластеризации текстовых данных большой размерности, позволяющих определить семантическую однородность полученных кластеров;

♦- выбор способа визуализации результатов кластеризации с отражением семантических зависимостей на множестве элементов текстовых данных для графического представления связанных между собой элементов кластерной структуры, их анализа и уточнения параметров метода кластеризации.

2. Методы иерархической агломеративной кластеризации. Алгоритмы иерархической кластеризации рекурсивно находят систему вложенных разбиений анализируемой выборки данных на непересекающиеся кластеры, применяя агломеративные (объединяющие) или дивизионные (разделительные) стратегии [17]. В агломеративных алгоритмах каждый объект из множества наблюдений рассматривается как независимый кластер. На каждом шаге алгоритма вычисляется матрица расстояний на основе критериев сходства признаков и последовательно объединяются объекты, обладающие непустым набором общих семантических свойств или закономерностей, пока все объекты не будут составлять один кластер.

Агломеративные методы различаются используемыми мерами расстояния для вычисления семантической близости и алгоритмами объединения, использующие различные критерии связи [8, 19].

1. Одиночная связь (Single link). Условие объединения кластеров C_1 и C_2 по принципу ближайшего соседа представлено как:

$$D(C_1, C_2) = \min_{x \in C_1, y \in C_2} dist\ (x, y), \quad (1)$$

где *dist(x,y)* – мера семантической близости, используемая при создании матрицы расстояний, $X = (x_0, x_1, \ldots, x_n)$, $Y = (y_0, y_1, \ldots, y_n)$ – множество n векторов признаков объектов кластеров C_1 и C_2 соответственно.

2. Полная связь (Complete link). Два кластера объединяются на основе вычисления расстояния между их самыми дальними объектами по формуле:

$$D(C_1, C_2) = \max_{x \in C_1, y \in C_2} dist(x, y). \quad (2)$$

3. Метод попарного среднего (Average link). Расстояние между двумя кластерами рассчитывается как среднее арифметическое между всеми возможными парами объектов из каждого кластера согласно:

$$D(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1, y \in C_2} dist(x, y). \quad (3)$$

4. Центроидная связь (Centroids link). В этом методе объединяются два кластера с минимальным расстоянием до центроида, межкластерное расстояние между центрами тяжести определяется как:

$$D(C_1, C_2) = \min_{x \in C_1, y \in C_2} dist\left(\left(\frac{1}{|C_1|}\sum_{x \in C_1} \vec{x}\right), \left(\frac{1}{|C_2|}\sum_{y \in C_2} \vec{y}\right)\right). \quad (4)$$

5. Связь Уорда (Ward's metod). В данном методе критерием выбора пары кластеров для слияния является оптимальное значение целевой функции (ЦФ) связи, определяющей расстояние между двумя кластерами, которая вычисляется как сумма квадратов отклонений (ESS) от среднего вектора (центроида) после объединения кластеров. Способы вычисления расстояния между кластерами задаются формулой Ланса – Уильямса [7], а расстояния между объектами можно задавать любой метрикой семантической близости. Метод Уорда основан на объединении не максимально близких кластеров, а тех, слияние которых дает минимальный прирост внутрикластерной дисперсии.

Преимуществами иерархических методов является наглядность представления агломератов в древообразном представлении наблюдений, называемом дендрограммой, возможность получать сбалансированные кластеры произвольных форм и идентифицировать выбросы в наборе текстовых данных. К сложностям можно отнести: ограничение объема набора данных; выбор меры близости; отсутствие глобальной целевой функции, каждый шаг иерархической кластеризации является локальным решением; вычислительная и временная сложность значительно ограничивают способность алгоритмов обрабатывать текстовые данные большой размерности; систему полных разбиений, которая может являться излишней в контексте решаемой задачи NPL, поэтому в алгоритме необходимо задавать пороговое значение расстояния для усечения иерархии или максимальное количество объектов в кластере.

Рассмотрим модифицированные алгоритмы Cure, Chameleon, Rock, в которых методы иерархической кластеризации интегрированы с другими методами, преодолевающими ограничения агломеративных алгоритмов.

Алгоритм CURE (Clustering Using REpresentatives) выполняет агломеративную кластеризацию с использованием репрезентативных объектов - представителей окрестности семантических признаков текстовых данных каждого кластера [17]. Для работы с данными большой размерности CURE использует комбинацию случайной выборки набора данных и сегментирования распределенных объектов-представителей кластера [20]. Выбор представителей, принадлежащих кластеру начинается с наиболее удаленных, с последующим сжатием расстояния между ними и центром кластера на некоторую долю α. Объединение кластеров происходит на основе вычисления расстояния между наиболее близкими представителями двух кластеров. Если $\alpha = 0$, то алгоритм объединения приближается к методам кластеризации по принципу ближайшего соседа, в случае $\alpha = 1$, к центроидным методам. Такой подход позволяет ослабить влияние выбросов, фиксировать форму и размер кластера.

Исходными данными алгоритма являются: m объектов размерности n, k – количество результирующих кластеров, c – количество объектов-представителей кластера, α – параметр сдвига объектов к центроиду кластера, γ – доля объектов

выборки на наборе данных. Вычислительное ядро алгоритма представляет собой итерационное построение заданного числа кластеров, вычисление их центроидов и назначение объектов-представителей для каждого кластера [20].

Шаг 1. Построение дерева кластеров на основе случайной выборки репрезентативных объектов на исходном наборе данных [γ,m]. Каждый объект принимается за кластер, центроид которого равен самому объекту. Далее выполняются следующие итерации, пока количество кластеров не станет равно k:

1) для каждого i-го кластера вычисляется центроид μ_i по формуле: $\mu_i=\frac{1}{K_i}\sum_{m_j\in K_i} m_j$, где m_j – множество объектов-представителей, находящихся в K_i-м кластере;
2) выполняется поиск пары кластеров, расстояние между центроидами которых минимально и определяется по двум ближайшим объектам-представителям согласно формуле (1).

Шаг 2. Пары, имеющие наибольшее сходство объединяются и для образованного кластера определяются представительные объекты:

1) выбирается произвольный объект-представитель кластера;
2) пока количество представителей меньше c выбирается представителем объект, расстояние от которого, до уже выбранных является максимальным согласно формуле (2);
3) для каждого представителя выполняется сдвиг к центроиду соответствующего кластера в α раз.

Шаг 3. Пересчет наиболее близкого кластера для некоторого поднабора объединенных кластеров по формуле (4). Если полученные на этом шаге кластеры отвечают ЦФ, то завершить кластеризацию, иначе – перейти к шагу 3.

Для ускорения кластеризации, на шаге 2 CURE производит разбиение полученных объектов-представителей. Наборы n объектов сегментируются на p равных частей, и на них выполняется иерархическая кластеризация, пока количество агломератов каждой части не станет равно n/pq, где $q>0$.

Алгоритм CURE менее чувствителен к выбросам и позволяет получать семантически однородные кластеры сложной формы и различных размеров, хорошо масштабируем для больших данных без ущерба для качества кластеризации. Общая временная сложность алгоритма равна $O(n^2log(n))$, где n – количество объектов кластеризации, пространственная сложность этого алгоритма за счет использования динамических структур данных в виде дерева кластеров равна $O(n))$.

К недостаткам данного метода можно отнести то, что при объединении кластеров не учитывается их внутреннее сходство и необходимость задания параметров порога плотности и количества кластеров, от которых во многом зависят результаты анализа.

Алгоритм ROCK (RObust Clustering using linKs) – алгоритм агломеративной кластеризации, который использует стратегию поиска связей в категориальном наборе данных по принципу взаимного соседства [21]. В ROCK алгоритме сходство кластеров основано на количестве объектов из разных кластеров, имеющих общих соседей. Особенностью алгоритма является введение функции связи для поиска скрытых закономерностей в окрестностях векторного пространства признаков текстовых данных. Соседями являются объекты x_i и y_j, для которых значение функции сходства $sim(x_i,y_j)\geq\theta$ превышает заданный порог θ. Предполагается, что 1 соответствует абсолютной близости, и 0 – наоборот. Выбор функции sim и значения θ зависят от входных данных и особенностей реализации алгоритма. Значение функции связи $link(x_i,y_j)$ между x_i и y_j соответствует количеству общих соседей в окрестности поиска. Чем больше значение связи, тем больше вероятность, что объекты имеют одинаковую окрестность признаков и принадлежат одному кластеру. Кластеризация с использованием ROCK является более глобальной и не сводится только к локальному поиску объектов [17].

Задача кластеризации сводится к максимизации суммы связей объектов, принадлежащих одному кластеру, и минимизации ее для объектов из разных кластеров. ЦФ разбиения на *k* кластеров имеет следующий вид [21]:

$$F_i = \sum_{i=1}^{k} n_i^{f(\theta)i} \times \sum_{xi,yj \in C_i} \frac{link\,(x_i,y_i)}{n_i^{1+2f(\theta)_i}}, \quad (5)$$

где C_i – кластер i размера n, ожидаемая сумма связей – $n_i^{1+2f(\theta)_i}$, ожидаемая сумма соседей – $n_i^{f(\theta)i}$, для функции $f(\theta)$ выбирается взвешенное представление $\frac{1+\theta}{1-\theta}$, что предупреждает отнесение объектов с низкими значениями связи к одному кластеру.

Для слияния кластеров вводится ЦФ качества (полезности), определяющая насколько выгодно объединять полученные кластеры:

$$Q(C_i, C_j) = \frac{link\,|C_i,C_j|}{(n_i+n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}. \quad (6)$$

Значение связи между кластерами представлено суммой попарных связей объектов, принадлежащих разным кластерам и определяется как:

$$link\,|C_i, C_j| = \sum_{x_i \in C_i, y_j \in C_j} link\,(x_i, y_i). \quad (7)$$

Алгоритм состоит из двух основных этапов. На входе имеется *n* объектов и *k* – количество кластеров для разбиений. Для каждого *n* задана функция схожести *sim*. На первом этапе каждый объект представляется отдельным кластером, для него вычисляется значение *link* и значение ЦФ для объединения с другим кластером по формуле (5). Для полученных решений создается глобальное дерево, которое содержит все подмножества кластеров, связь между которыми не нулевая.

На втором этапе выполняется цикл, на каждом шаге которого объединяются два подмножества кластеров с максимальным значением функции полезности согласно формуле (6), после чего вносятся соответствующие изменения в дерево кластеров. После объединения производится пересчет значений связи и качества для новообразованного кластера. Алгоритм завершает работу в двух случаях: когда получено *k* кластеров или, когда все связи между оставшимися кластерами равны 0.

К преимуществам алгоритма ROCK можно отнести высокую эффективность кластеризации и масштабируемость, эффективность работы с выбросами, ROCK подходит для данных произвольной формы и большой размерности. Недостатки: результат кластеризации чувствителен к параметрам алгоритма. Временная сложность алгоритма равна $O(n^2 log(n))$, где *n* – количество объектов.

Алгоритм CHAMELEON (hierarchical clustering using dynamic modeling) – гибридный алгоритм агломеративной кластеризации на основе графо-ориентированных и классических иерархических методов, использующий динамическую модель определения сходства между парами кластеров [22]. Существующие алгоритмы применяют статическую модель кластеризации и не используют информацию о характере отдельных кластеров по мере их слияния. Так алгоритм CURE не учитывает информацию о совокупной взаимосвязанности элементов в двух кластерах, а алгоритм ROCK – информацию о близости двух кластеров, определяемую сходством ближайших элементов в них. Ключевой особенностью Chameleon является то, что при вычислении сходства между кластерами используются их относительная взаимосвязь и относительная близость, а объединение наиболее похожей пары кластеров происходит с учетом нормализации их абсолютных значений. Chameleon позволяет выявлять стабильные кластеры в наборе текстовых данных, используя трехэтапный алгоритм [7, 22].

Первый этап заключается в построении графа $G=(X,E)$ по матрице сходства объектов (по множеству объектов $x_i,x_j \in E$ и мере их близости $dist(x_i,x_j)$) по принципу k ближайших соседей $dist(x_i,x_j) \in sorted(\{r|r=dist(x_i,x_k),k=1,...,n\})[1..k]$, причем ребра – взвешенное значение близости объекта с одним из k-ближайших соседей другого объекта $weight(e)=dist(x_i,x_j)$. Вершина графа, соответствующая некому объекту, соединяется с k вершинами, расстояние до которых минимально.

На втором этапе Chameleon использует алгоритм разделения графа для получения набора относительно малых подкластеров. Количество кластеров, на которые будет разбит граф, заранее неизвестно, и зависит от свойств объекта и выбранной метрики. Для разбиения графа создается множество $C=G$ и выполняется цикл со следующими операциями на каждой его итерации:

- выделить C_{gr} такой, что $|C_{gr}|=\max\limits_{1\le j\le |C|}|C_j|$;
- если $|C_{gr}||X|>min$, то разделить C_{gr} на 2 кластера C_1 и C_2, удалить C_{gr} из C, и добавить C_1 и C_2 в C;
- иначе процесс дальнейшего разбиения прекращается, переход на 3-й этап.

На третьем этапе выполняется агломеративная кластеризация для формирования результирующего множества подлинных кластеров путем многократного объединения полученных ранее подкластеров. Формируется начальный набор кластеров $C_{тек}=C$ и на каждой итерации слияния кластеров выполняется следующий цикл.

1. Для каждой пары кластеров $C_i,C_i \in C_{тек}$ вычисляются показатели: абсолютной взаимной связности $EC(C_i,C_j)$, рассчитывается как сумма весов ребер, соединяющих вершину 1-го кластера C_i с вершинами второго кластера C_j по формуле: $EC(C_i,C_j)=\sum_{e\in conn}(C_i,C_j)weight(e)$, где $conn(C_i,C_j)=\{(v,w)|v\in C_i,w\in C_j\}$ и внутренней связности кластера $EC(C_i)$ определяется как сумма весов ребер, разбивающих C_i на 2 равных подграфа: $EC(C_i)=\sum_{e\in mincut}(C_i)weight(e)$, где $mincut(C_i)$ – минимальный разрез C_i. Аналогичным образом вычисляются $EC(C_j)$.

2. Определяется относительная взаимная связность пары кластеров $RI(C_i,C_j)$ по формуле $RI(C_i,C_j)=2*|EC(C_i,C_j)|/(|EC_{Ci}|+|EC_{Cj}|)$. Нормализация с учетом EC_{Ci} используется для исключения тенденции к преимущественному объединению крупных кластеров, у которых значение взаимной связности достаточно велико, вследствие их размерности.

3. Далее вычисляется относительное взаимное сходство $RC_{C_i,C_j}=\overline{S_{EC_{(C_i,C_j)}}}/(\frac{|C_i|}{|C_i|+|C_j|}\overline{S_{EC_{(C_i)}}}+\frac{|C_i|}{|C_i|+|C_j|}\overline{S_{EC_{(C_j)}}})$, где $\overline{S_{EC_{(C_i,C_j)}}}=EC(C_i,C_j)|conn(C_i,C_j)|$ – абсолютное взаимное сходство пары кластеров и $\overline{S_{EC_{(C_i)}}}=EC(C_i)|mincut(C_i)|$ – среднее сходство между связанными объектами C_i.

4. Последовательное объединение пары кластеров происходит на основе вычисления ЦФ $F=\max RI(C_i,C_j)*RC\alpha(C_i,C_j)$, где α – заданный заранее параметр. Процесс слияния останавливается, когда количество кластеров, удовлетворяющих ЦФ менее одного.

Преимуществами Chameleon является высокая точность и эффективность результатов кластеризации. К недостаткам метода можно отнести то, что временная сложность резко возрастает с увеличением сложности графа. Общая временная сложность алгоритма равна $O(n^2)$, где n – количество объектов кластеризации.

Методы неиерархической (итерационной) кластеризации. Существующие методы неиерархической кластеризации отличаются подходами к разделению набора текстовых данных на определенное количество непересекающихся кластеров [8]. Первый заключается в идентификации неявной кластерной структуры, на основе «сгущения объектов» для определения границ кластеров с высокой локаль-

ной плотностью в многомерном пространстве исходных данных, а области низкой плотности соотносятся с шумовыми признаками. Кластер определяется локальным максимумом оценочной функции плотности. Этот подход реализован в плотностных алгоритмах DBSCAN, OPTICS, DBCLASD, позволяющих с высокой точностью идентифицировать кластеры произвольной формы и размера для данных большой размерности, искаженных шумом [7, 23].

Второй подход основан на разделении векторного пространства текстовых данных на оболочки смысловых кластеров, с минимальными ограничениями на признаки отбора, во избежание исключения скрытых закономерностей, поведение которых имеет характер шума [8]. Алгоритмы, поддерживающие данную идею кластеризации путем итеративных вычислений выполняют случайную инициализацию центроидов и определяют ближайшие вектора данных для каждого из них. После чего, выполняется итеративное смещение центров кластеров и пересчёт ближайших векторов данных, расчет суммарного квадратичного отклонения данных от центров полученных кластеров и выбор наилучшего локального решения, для которого ЦФ будет минимальной. К таким алгоритмам относятся k-means и его модификации K-modes, K-medoids:PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications), CLARANS (Randomized CLARA), отличающихся применением различных стратегий инициализации центров кластеров и метрик несходства для вычисления расстояния от каждого наблюдения до центров их кластеров [6–9].

Неиерархические методы проявляют более высокую устойчивость по отношению к шумам и выбросам, выбору метрики схожести, включению незначимых переменных в набор, участвующий в кластеризации, но слишком чувствительны к инициализации параметров центроидов, требуют задания начальных условий (количество образуемых кластеров, порог завершения кластеризации и др.), задача нахождения глобального оптимального решения для данных алгоритмов является NP-полной.

Алгоритм кластеризации K-medoids: PAM (Partitioning Around Medoids). Алгоритм k-medoids является модификацией k-means и предназначен для решения задач выделения кластеров в случаях, когда проводится кластеризация объектов без использования свойств линейного пространства [5]. В отличие от k-means, центром кластера может быть не любой объект признакового пространства (центроид), а только, принадлежащий кластеризуемой выборке представитель – медоид, среднее несоответствие (сумма различий) которого по отношению ко всем другим объектам в кластере минимальна [8]. Принадлежность объекта x_c к соответствующему кластеру C определяется индексом наиболее близкого к нему медоида x_m.

$$M(C) = \ arg\, min_{x_m \in C} \sum_{x_c \in C} dist(x_c, x_m). \quad (8)$$

K-medoid использует произвольные меры сходства для вычисления близости *dist* и минимизирует абсолютную ошибку или «полное отклонение» *TD* суммы попарных различий между объектами в кластере C_i и медоидами m_i как:

$$TD = \min \sum_{i=1}^{k} \sum_{x_c \in C_i} dist(x_c, m_i). \quad (9)$$

В общем случае задачу полного разбиения алгоритмом *k*-medoids, применяющего стратегию полного перебора решить точно NP-сложно. Для решения задачи нахождения лучшего разбиения с использованием медоидов предназначена PAM-реализация алгоритма k-medoids [14]. PAM использует стратегию «жадного» поиска, преобразует каждый шаг алгоритма из детерминированных вычислений в задачу статистической оценки и уменьшает размерность выборки текстовых данных *n* до *O(n log n)*.

Входными данными алгоритма являются множество объектов $X=\{x_1,\ldots,x_n\}$ и их индексов $z=\{1,\ldots,n\}\subset Z$ с заданной функцией – метрикой расчета расстояний между объектами, $d_{ij}=dist(x_i,x_j)$, либо матрица расстояний $D\in Rn\times n$, характеризующая дистанции между каждой парой объектов $D=(d_{ij}),i\in z,j\in z$, а также число кластеров $k\leq n$, на которое необходимо разбить множество объектов. На выходе алгоритм выдает множество индексов объектов, принятых в качестве медоидов $m=\{m_1,\ldots,m_k\}$. PAM состоит из двух последовательных алгоритмов: BUILD для выбора начальной кластеризации и SWAP для улучшения кластеризации в сторону локального оптимума [16].

BUILD. Построение начального набора k медоидов путем их последовательного выбора. Первый из них выбирается как объект, сумма различий которого до всех других объектов минимальна согласно (8). Далее осуществляется последовательный выбор следующего медоида из оставшихся объектов по формуле (9) до тех пор, пока $m = m \cup \{arg\max_{i\in z|m} \sum_{j\in z|\{m\cup i\}} max(D_j - d_{ij}, 0)\}$, D_j – индекс первого ближайшего j-медоида. Выбранный набор медоидов характеризует конфигурацию кластеров, для которой может быть вычислена ее ЦФ стоимости.

SWAP. На каждой итерации производится перебор всех комбинаций пар медоид-немедоид и для них рассчитывается ЦФ стоимости смены текущей конфигурации как:

$$T_m(i,h) = \sum_{j\in o|\{m\cup h\}} C_{jih}, \qquad (10)$$

где C_{jih} вклад объекта j в перестановку медоида i и немедоида h. Данная операция производится $k(n-k)$ раз для каждой итерации. Из всех комбинаций выбирается та замена, стоимость которой максимальна. Критерий останова PAM $T_m(i,h)<0$.

PAM обладает высокой точностью результатов кластеризации, устойчив к шумовым признакам и выбросам. Временная сложность алгоритма $O(n^2)$. Ключевой проблемой PAM является большое время выполнения на данных большой размерности и неэффективный поиск новых кандидатов на места медоидов.

Алгоритмы CLARA (Clustering Large Applications) и CLARANS (Randomized CLARA). Рассмотрим модификации PAM алгоритма CLARA и CLARANS, которые позволяют генерировать оптимальный набор медоидов из выборки текстовых данных на основе рандомизированной инициализации (рис. 3), что обеспечивает ускорение в $O(n)$ раз на фазе «SWAP» и дают точные результаты кластеризации [18].

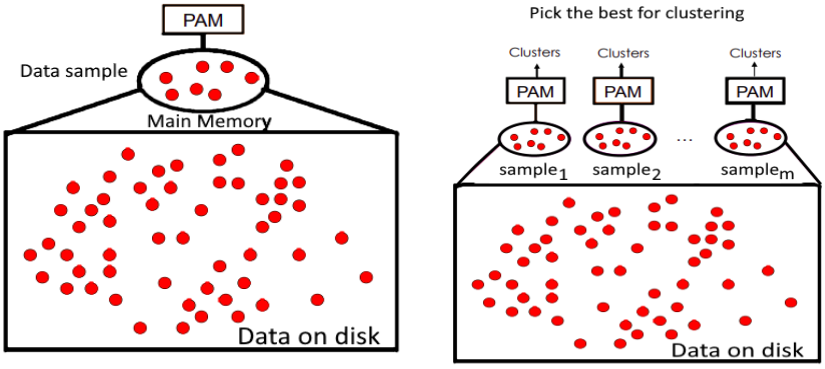

Рис. 3. Стратегия поиска медоидов алгоритмами CLARA и CLARANS

Вместо поиска медоидов для всего набора данных алгоритм CLARA случайным образом выбирает из исходного набора подмножество объектов данных фиксированного размера. Далее для создания оптимального набора медоидов выборки применяется алгоритм PAM, производится вычисление *k* репрезентативных объектов (медоидов), назначение каждого наблюдения из всего набора данных ближайшему медоиду, вычисление среднего (или суммы) различий наблюдений с их ближайшим медоидом и поиск центра кластера с наименьшей функцией стоимости из лучших *k* медоидов, кластеризованных для каждой выборки. CLARA повторяет процессы выборки и кластеризации заранее заданное количество раз. Окончательные результаты кластеризации соответствуют набору медоидов с минимальной стоимостью. Однако алгоритм может давать неправильные результаты кластеризации, если один или несколько выбранных медоидов отличаются от фактических лучших медоидов [8].

Алгоритм улучшает масштабируемость PAM, сокращает время вычислений при работе с большим набором данных, но качество кластеризации зависит от используемого метода формирования выборки, а на эффективность влияет ее размер и заданное количество результирующих кластеров. Временная сложность алгоритма равна $O(n^2 log(n))$.

Алгоритм CLARANS комбинирует методы PAM и CLARA, выполняя рандомизированный поиск только в подмножестве набора данных и формулирует задачу кластеризации как случайный поиск в графе [14]. Из исходного графа $G = (X,E)$, представляющего собой разбиение множества данных на число кластеров выбирается подмножество вершин S и кластеризуется подграф. Каждая вершина представлена набором признаков объектов данных, каждому из которых может быть назначен *k* медоид. Вершины подграфа являются соседями (т. е. соединенными дугой), если их наборы отличаются только одним объектом. Далее поиск оптимального набора медоидов осуществляется по алгоритму PAM [23].

Суть эвристики CLARANS заключается в сокращении множества для выбора нового медоида из всего графа до множества замен медоид-немедоид в каждом кластеризированном подграфе G_m. Это позволяет искать оптимальную замену не по всем вершинам графа, а только по одному кластеру, перебирая *s* соседних вершин внутри него [7]. Если $T_m(i,h)$ выбранного соседа согласно формуле (10) выше, чем у текущей вершины CLARANS переходит к этому соседу и продолжает процесс случайного выбора и сравнения. Поиск решения останавливается в том узле, где достигается локальный минимум. CLARANS минимизирует ту же целевую функцию, что и CLARA, путем изменения текущих медоидов таким образом, что сумма различий внутри кластера сводится к минимуму. На вычисление новых медоидов может влиять шум, поэтому выбор медоидов в CLARANS диктуется расположением преобладающей доли объектов внутри кластера и, следовательно, он менее чувствителен к наличию выбросов [8].

Алгоритм CLARANS устраняет недостатки как алгоритмов K-medoids, так и алгоритмов CLARA, он поддерживает баланс между вычислительными затратами и влиянием выборки данных на формирование семантически однородных кластеров. Временная сложность алгоритма равна $O(n^2)$.

4. Алгоритм кластеризации на основе аффинного преобразования. Алгоритм кластеризации Affinity propagation (AP), основанный на распространении сходства, является еще одним представителем алгоритмов неиерархической кластеризации. Основная идея: AP получает на вход матрицу схожести между элементами набора данных и возвращает набор меток, присвоенных этим элементам. Изначально AP рассматривает все объекты данных как потенциальные центры кластеров – «образцы», а сходство объекта данных из входного набора с «образцом» определяет

вероятность того, что этот объект будет соотнесен с центром кластера. Алгоритм АР не требует предварительного задания числа кластеров, и как *k*-medoids, находит «экземпляры» на входном наборе, которые представляют начальное представление центра кластера. Алгоритм использует жадную стратегию, которая максимизирует значение глобальной функции сходства на каждой итерации [24].

Пусть $X = (x_1,x_2,\ldots,x_n)$ кластеризуемый набор объектов данных без ограничений их внутренней структуры, $f=sim\ (x_i,\ x_i)$ – функция, количественно определяющая сходство между любыми двумя объектами по правилу: если $sim(x_i,\ x_i) > sim(x_i,\ x_k)$, то x_i больше похож на x_j, чем на x_k. Мера близости *sim* может вычисляться различными методами. Выполнение алгоритма сводится к итеративному выполнению двух этапов [8, 24].

1 этап – предварительная обработка каждого измерения и формирование матриц подобия: *R*– матрицы «ответственности» с элементами *r(i,k)* – значениями близости между парами, которые представляют пригодность экземпляра x_k быть образцом x_i, по сравнению с другими кандидатами для x_i и *A* – матрицы «доступности» с элементами *a*(*i*,*k*) значения которых представляют насколько «уместно» для x_i взять x_k как его экземпляр, принимая во внимание значения предпочтений других кандидатов данного «образца». Формализуем представления *r(i,k)* и *a(i,k)* как:

$$r(i,k) \leftarrow s(i,k) - max\ [a(i,k') + sim\ (i,k')\ \forall\ k' \neq k] \text{ и}$$
$$a(i,k) \leftarrow min\left[0, r(k,k) + \sum_{i \neq k,\, i' \notin \{i,k\}} r(i',k)\right]. \quad (11)$$

2 этап – обновление матриц *R* и *A*. Начальное значение этих матриц для *r(i,k)* и *a(i,k)* установлены в ноль, их расчет и обновление происходит на каждой итерации, и повторяется до сходимости. Как обсуждалось в работе [22], чтобы избежать числовых колебаний при обновлении значений, в итерационный процесс вводится коэффициент демпфирования λ:

$$r_{t+1}(i,k) = \lambda * r_t(i,k) + (1-\lambda) * r_{t+1}(i,k) \text{ и } a_{t+1}(i,k) = \lambda * a_t(i,k) + (1-\lambda) * a_{t+1}(i,k), \quad (12)$$

где *t* – время итерации. Итерации выполняются до тех пор, пока либо границы кластера останутся неизменными на протяжении ряда итераций, либо не будет достигнуто заданное число итераций. Из окончательных матриц извлекаются такие экземпляры диагоналей, чья «ответственность + доступность» положительна .

Главный недостаток метода АР – его сложность. Алгоритм имеет временную сложность порядка $O(n^2T)$, где *n* – количество «образцов» и *T* – количество итераций до сходимости. Преимущества: простая и понятная идея алгоритма, нечувствительная к выбросам и количеству кластеров, не требующая предварительной настройки параметров.

5. Методы визуализации результатов кластеризации. Визуальный анализ многомерных текстовых данных это частный случай решения задачи нелинейного понижения размерности векторного пространства признаков. Методы визуализации позволяют осуществлять точный их перенос на плоскость или в трёхмерное пространство так, чтобы графическое представление наглядно отражало связанные между собой элементы кластерной структуры [25]. Из-за отсутствия однозначности в ее решении и предъявления различных требований к характеристикам получаемых визуализаций – остается актуальным вопрос о выборе метода визуального отображения многомерных данных (каждый из которых имеет свою целевую направленность, сильные и слабые стороны). Также очевидно, что совместное взаимодополняющее применение различных методов кластеризации и визуализации результатов позволяет провести более глубокий и многосторонний анализ исследуемых данных.

Дендограмма – это визуализатор представления результатов иерархической кластеризации в виде древовидной структуры, построенной по матрице мер близости между кластерами. Дендограмма является наиболее эффективной формой визуального представления и анализа структуры взаимосвязей между объектами при агломеративной кластеризации, а также наглядно демонстрирует в графическом виде последовательность их объединения или разделения [17]. Количество уровней дендрограммы соответствует числу этапов слияния кластеров. На каждом этапе построения дендограммы вычисляются расстояния между кластерами и производится пересчет расстояния между образованными кластерами $D(C_i,C_i)=dist\ (x,y)$. В зависимости от специфики решаемой задачи для вычисления расстояния $D(C_i,C_i)$ между кластерами используются различные функции связи, рассмотренные ранее в пункте 2.

Дендрограмма позволяет представлять зависимости между множеством объектов с любым числом заданных характеристик на двумерном графике, где по одной из осей откладываются все объекты, а по другой D_t – расстояние между кластерами, выбранными на шаге t для объединения (рис. 4). Для того, чтобы любой кластер мог быть представлен в виде непрерывного отрезка на оси объектов и ребра не пересекались (исключение самопересечений), на D_t накладывается ограничение монотонности.

Функция расстояния D_t является монотонной, если на каждом следующем шаге расстояние между кластерами не уменьшается: $D_2 \leqslant D_3 \leqslant \cdots \leqslant D_m$. и, для коэффициентов в формуле Ланса-Уильямса верна теорема Миллигана [7]. Следующим ограничением являются свойства растяжения $D_t \geq dist\ (\mu_x,\mu_y),\ \forall t$ и сжатия $D_t \leq dist\ (\mu_x,\mu_y),\ \forall t$, которые способствуют четкому отделению кластеров.

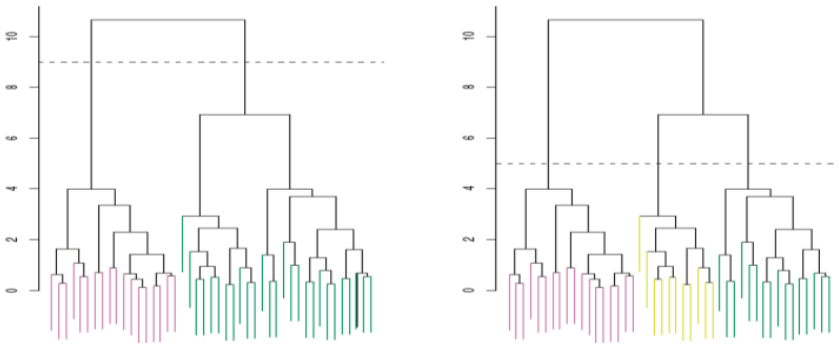

Рис. 4. Дендограмма с отсечением кластеров

Для определения числа кластеров находится интервал максимальной длины $|Dt+1-Dt|$. В качестве итоговых результатов выдаются кластеры, полученные на шаге t. При этом число кластеров равно $m-t+1$.

Для оценки производительности неиерархической кластеризации на наборах текстовых данных, их сравнительного анализа и интерпретации результатов используются средства непараметрической визуализации, поддерживающие следующие методы: t-SNE (t-distributed Stochastic Neighbor Embedding), Sammon, Isomap и локально линейное встраивание (LLE) [14, 25]. Данные алгоритмы предоставляют различные методы оценки внутренней связности и анализа скрытых закономерностей на множестве элементов кластерной структуры. Основными способами непараметрической визуализации на основе данных методов являются диаграммы рассеивания (разброса), диаграммы сходства и диаграммы интенсивности (тепловые карты).

Диаграммы рассеивания **– способ интерактивной** визуализации кластеризации многомерных текстовых данных путем проецирования их в низкоразмерное двух- или трехмерное пространство. На диаграмме каждому многомерному объекту соответствует точка, координаты которой равны значениям двух или более параметров этого наблюдения [13]. Если предполагается, что один из параметров зависит от другого, то сходные объекты моделируются близлежащими точками, а разнородные объекты моделируются удаленными точками с высокой вероятностью. Диаграмма рассеяния позволяет отражать несколько зависимостей: параметры одной переменной *X* откладываются по горизонтальной оси, а по вертикальной оси – значения нескольких переменных *Y*. Для каждой переменной *Y* используется разный цвет и вид точек (рис. 5, 6).

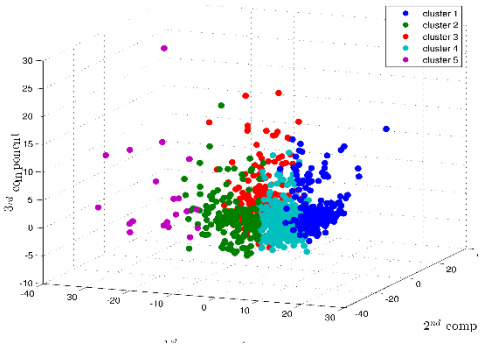

Рис. 5. Трехмерное представление диаграммы

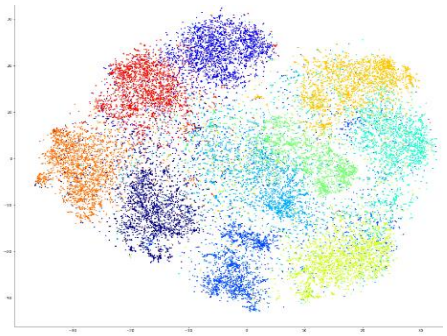

Рис. 6. Двухмерное представление диаграммы

Исследование диаграмм рассеяния позволяет определять формы зависимостей в связанных между собой данных кластерной структуры, проводить их анализ и уточнение параметров метода кластеризации. Другое важное преимущество диаграмм рассеивания состоит в том, что они позволяют находить «выбросы» (нетипичные данные), которые искусственным образом увеличивают или уменьшают («смещают») коэффициент корреляции.

Диаграмма сходства VOS ***(Visualization of Similarities)*** – это еще один инструмент исследования качества моделей кластеризации, основанный на визуализации зависимостей между многомерными данными [26]. Цель VOS – обеспечить низкоразмерную визуализацию, в которой объекты расположены таким образом, чтобы расстояние между любой парой объектов максимально точно отражало их сходство. Объекты распределяются по кластерам путем максимизации функции качества. Графическое представление результатов кластеризации текстовых данных показано на рис. 7, а визуальный анализ связности терминов для сформированных кластеров 2, 7, 8, 10 на рис. 8. На диаграммах VOS размер объекта характеризует общую мощность связей («total link strength»), а ширина линий – мощность связи («link strength») между двумя терминами.

Анализ диаграммы помогает исследовать следующие аспекты данных: различные уровни корреляции между точками данных необходимы для понимания взаимосвязи внутри данных; для заданных данных можно провести линию наилучшего соответствия и использовать ее для дальнейшего прогнозирования новых значений данных; точки данных, лежащие за пределами данного набора, могут быть легко идентифицированы, чтобы найти выбросы; группировку точек данных на диаграмме сходства можно определить как разные кластеры в данных.

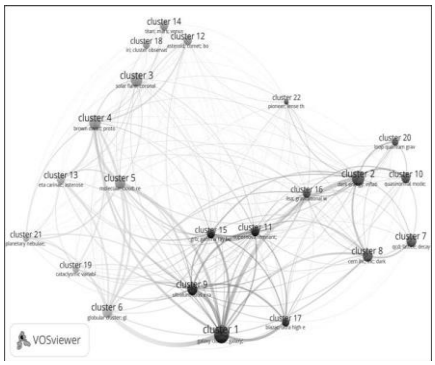

Рис. 7. Визуализация кластеров

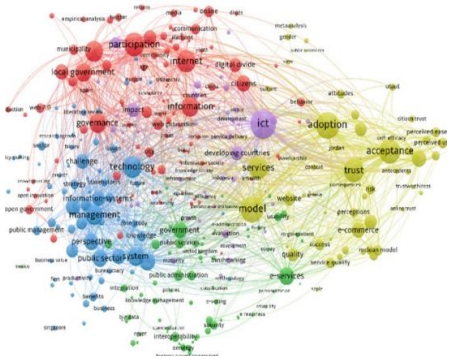

Рис. 8. Визуализация связей

Диаграммы интенсивности или тепловые карты (Term map) отображают точечные объекты в виде растровой поверхности, выделяя на цветовой шкале области с более высокой относительной плотностью точек. Тепловые карты используют алгоритм Isomap для вычисления и отображения относительной плотности распределения объектов в виде плавно меняющейся цветовой гаммы: от холодного (низкая плотность) до теплого (высокая плотность). Term map предлагают эффективный способ взвешивания плотности объектов на основе числового значения семантической близости данных в кластере [26]. Данный способ визуализации позволяет определять наличие статистически значимых выбросов и пространственной закономерности в распределении данных, находить кластеры среди точечных объектов в окружающем шуме на основе их плотностного распределения.

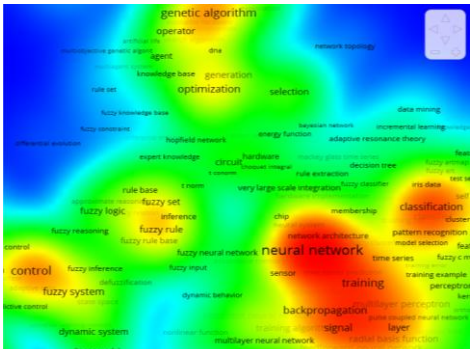

Рис. 9. Фрагмент Term map текстового корпуса

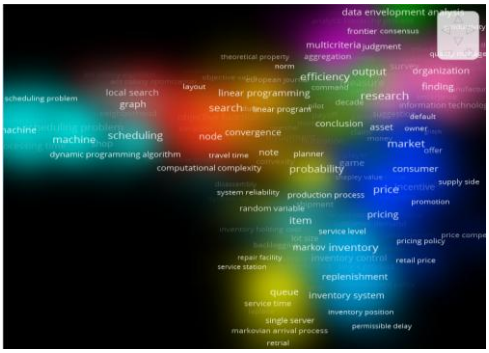

Рис. 10. Фрагмент Term map с сформированными кластерами

На сегодняшний день существует большое количество программных средств, предназначенных для решения задач визуального анализа многомерных текстовых данных в интерактивном режиме, среди них можно выделить: **VOSviewer**, Matplotlib, Tensorflow, **Ucinet**, CiteSpace и др.

6. Экспериментальные исследования. Для оценки эффективности кластеризации используют следующие группы методов: внешние метрики позволяют выполнить сравнение результатов о настоящем распределении объектов по кластерам на основе дополнительной (внешней) информации о кластеризуемом множестве: распределение по кластерам, количество кластеров и др. и внутренние меры оценивают качество структуры кластеров опираясь только на информацию о

близости данных [27]. Для оценки кластеризации будем использовать в качестве внешних мер – скорректированный индекс Ранда (Adjusted Rand Index - ARI) и V-меру, а внутренних – коэффициент «силуэта».

V-мера является средним гармоническим значением между двумя другими метриками: гомогенностью и полнотой [19]. Однородность характеризует степень соответствия тому, что каждый кластер содержит только объекты одного класса. Полнота показывает, в какой мере все объекты данного класса собраны в одном кластере. *V=2*Homogeneity*Completeness/(Homogeneity+Completeness).* Величина V-меры изменяется в диапазоне от 0 до 1. V-мера комбинирует в себе гомогенность и полноту таким образом, чтобы максимизация итоговой метрики не приводила к тривиальным решениям.

ARI – это метрика качества, построенная на основе меры близости, вычисляющей семантическое сходство между двумя кластерами. ARI измеряет степень соответствия между двумя распределениями данных, которые находятся как в одном кластере, так и в разных кластерах, с поправкой на случайность [12].

$$ARI = \frac{a+b}{C_2^{n_{samples}}}, \quad (13)$$

где a – количество объектов, которые фактически относятся к одному и тому же кластеру и предсказаны для одного кластера; b – количество объектов, которые и верны, и предсказаны в разных кластерах; в знаменателе количество комбинаций любых двух объектов, попадающих в один кластер, которые могут быть сформированы в наборе данных. Диапазон значений ARI составляет [0,1]. Чем больше значение, тем более согласован результат кластеризации с реальной ситуацией.

Коэффициент «силуэта» – метрика кластеризации, не требующая разметки, измеряет схожесть объектов своего кластера по сравнению с объектами других кластеров. Коэффициент определяется для каждого объекта выборки, а метрика определяется как средний коэффициент «силуэта» для всех объектов выборки [6].

$$Sil(C) = \frac{1}{N}\sum_{c_k \in C}\sum_{X_i \in c_k}\frac{z(x_i,c_k)-q(x_i,c_k)}{\max\{q(x_i,c_k),z(x_i,c_k)\}}, \quad (14)$$

где $q(x_i,c_k)$ – компактность распределения, определяется как среднее расстояние близости $x_i \in c_k$ до других объектов кластера c_k, $z(x_i,c_k)$ – отделимость, рассчитывается как среднее расстояние близости от $x_i \in c_k$ до объектов другого кластера $c_l: k \neq l$. Коэффициент принимает значения $-1 \leq Sil(C) \leq 1$ и максимизируется, когда кластеры «кучные» и хорошо отделены друг от друга.

Данные, используемые для экспериментального исследования, были получены из текстового корпуса базы данных научных публикаций Hindawi, находящейся в открытом доступе. В качестве предметной области для проверки результатов работы алгоритмов кластеризации была сформирована выборка из 11628 оригинальных исследовательских статей с термином «Knowledge managment» в названиях, аннотациях и терминах индексирования (ключевых словах). Для фильтрации текстов был разработан парсер текстов, реализующий возможность выполнять операцию нахождения основы слова с помощью алгоритмов стемминга и лемматизации. Составление векторов текстов в виде моделей «документ-термин с TF-IDF», «документ - ассоциативно-семантическая группа с TF-IDF» выполнялось алгоритмом Sem_group из библиотеки scikit-learn [13].

Для сравнения эффективности методов агломеративной и неиерархической кластеризации были выбраны алгоритмы: «k-medoids:PAM», «Affinity Propagation», «ROCK» и «CHAMELEON». Так как алгоритм «k-medoids:PAM» определяет необходимость задания количества кластеров, для него исследуемая выборка текстовых данных была размечена в полуавтоматическом режиме и со-

ставила 80 кластеров различных смысловых тематик. Алгоритмы «Affinity Propagation», «ROCK» и «CHAMELEON» без предварительной параметризации определили количество кластеров в процессе эксперимента. На следующем этапе проведены исследования по сравнению качества разбиения данными алгоритмами кластерного анализа на сформированной выборке текстовых данных, для которой был предварительно размечен словарь терминов на 1000 элементов и выявлено 59 наиболее часто встречающихся терминов. Результаты сравнения алгоритмов кластеризации по внешним и внутренним метрикам качества и времени работы представлены в табл. 1.

Таблица 1

Сравнение результатов кластеризации текстовых данных различными алгоритмами

Алгоритм кластеризации	Алгоритм векторизации	Время, с	V-мера	ARI	Коэфф. «силуэта»
k-medoids:PAM	tf_idf	397	0,721	**0,357**	0,544
k-medoids:PAM	sem_group	457	0,753	**0,360**	0,549
CHAMELEON	tf_idf	1378	0,714	0,329	**0,752**
CHAMELEON	sem_group	1056	0,741	0,336	**0,856**
ROCK	tf_idf	986	0,613	0,128	0,349
ROCK	sem_group	875	0,661	0,135	0,351
Affinity Propagation	tf_idf	**197**	**0,762**	0,208	0,434
Affinity Propagation	sem_group	**236**	**0,794**	0,334	0,536

Из представленных в табл. 1 результатов можно сделать вывод о том, что все алгоритмы показывают достаточно близкую точность работы. По временным затратам алгоритм «Affinity Propagation» показал лучший результат, он на 200 секунд быстрее «k-medoids:PAM». По метрике «V-мера», характеризующей семантическую однородность распределения статей в кластеры алгоритмы кластеризации «k-medoids:PAM» со значением 0,753 и «Affinity Propagation» со значением 0,762 представили качественное разбиение. Лучший результат измерения согласованности между результатами кластеризации и данными реальных категорий получен алгоритмом кластеризации «k-medoids:PAM» со значением «Adjusted Rand Index» – 0,360. Статистически значимый показатель локальной плотности по метрике «силуэт» показал агломеративный алгоритм CHAMELEON – коэффициент 0,856, что говорит о высоком уровне формирования семантически плотных кластеров и хорошем их разделении.

Алгоритм «k-medoids:PAM» производит кластеризацию текстовой выборки с выделением обобщенных тематических направлений. Алгоритмом «Affinity Propagation» определяются подтемы, так, что узко специфические тематики находятся в каждом кластере. Алгоритм «CHAMELEON» по результатам кластеризации близок к «k-medoids:PAM», и для всех алгоритмов векторизации точно распределяет темы статей по кластерам. Алгоритм «ROCK» показывает слабые результаты кластеризации, статьи по одному тематическому направлению оказываются в разных кластерах, а во многих кластерах определились статьи по нескольким различным тематикам. Это может быть объяснено тем, что в аннотациях слишком кратко изложена сущность текстового документа или используются специализированные термины.

Так же по результатам работы всех алгоритмов видно, что качество кластеризации при использовании модели представления текста «документ – ассоциативно-семантическая группа с TF-IDF» значительно улучшается.

По результатам проведенной кластеризации для визуального анализа и оценки семантической связности элементов кластерной структуры из словаря были выбраны 835 ключевых терминов с частотой упоминания, превышающей или равной пяти. Для снижения полученного пространства признаков использовался алгоритм t-SNE, который позволяет выполнять визуализацию элементов многомерного пространства путѐм снижения размерности до двух [25]. Так как построенные вектора текстов имеют большую размерность, длина их векторов зависит от длины набора терминов, которые выбраны в качестве основных признаков. Диаграмма сходства, построенная на основе t-SNE для выбранных терминов представлена на рис. 11. Графическое представление получено с использованием VOSviewer программного инструмента для построения и интерактивной визуализации результатов кластерного анализа текстовых данных [26].

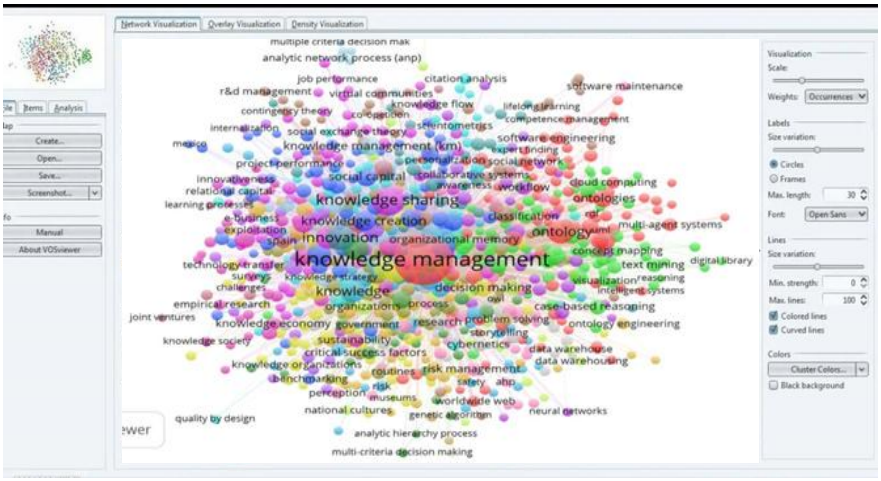

Рис. 11. Диаграмма VOS

Интерпретация кластеров основана на встречающихся в них ключевых словах, которые тесно связаны внутри одних и тех же кластеров, но слабо связаны между разными кластерами. На основе визуального анализа частоты встречаемости терминов определена мощность (числовое значение близости данных) семантической связи между элементами кластеров и плотность каждого кластера по количеству связанных элементов, что позволило выделить шумовые признаки и выбросы.

Заключение. В работе рассмотрена одна из важных задач искусственного интеллекта - кластерный анализ слабоструктурированных текстовых данных. В статье проведен обзор и анализ модифицированных алгоритмов агломеративной кластеризации CURE, ROCK, CHAMELEON, неиерархической кластеризации k-medoids:PAM, CLARA, CLARANS и алгоритма аффинного преобразования. Выявлены их достоинства и недостатки, а также рассмотрены особенности их реализации. Рассмотрены методы интерактивной визуализации результатов кластеризации, оценки внутренней связности и анализа скрытых закономерностей на множестве элементов кластерной структуры: дендограммы, диаграммы рассеивания, диаграммы сходства VOS и диаграммы интенсивности. Данные способы графического представления позволяют работать с многомерными наборами данных для снижения размерности векторного пространства признаков. Проведены эксперимен-

тальные исследования рассмотренных алгоритмов кластеризации текстовых данных. Сравнительный анализ алгоритмов показал, что наилучшее качество при наименьшем времени работы показывают неиерархические алгоритмы – «k-medoids:PAM» и «Affinity Propagation» с использованием модели векторизации «документ – лексико-семантическая группа с TF-IDF». Проведенный обзор и анализ показал, что для повышения эффективности кластеризации текстовых данных необходимо использовать гибридные подходы и метаэвристические алгоритмы.

Исследование выполнено за счет гранта Российского научного фонда № 22-21-00316, https://rscf.ru/project/22-21-00316/ в Южном федеральном университете.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Junkai Yi, Yacong Zhang, Xianghui Zhao, Jing Wan* A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network // Mathematical Problems in Engineering. – 2017. – Vol. 2017. – 13 p.
2. *Yujia Sun, Jan Platoš*. High-Dimensional Text Clustering by Dimensionality Reduction and Improved Density Peak // Wireless Communications and Mobile Computing. – 2020. – Vol. 2020. Article ID 8881112. – 16 p.
3. *Бова В.В., Кулиев Э.В., Щеглов С.Н.* Метод семантической кластеризации распределенных ресурсов знаний с динамическими компонентами на основе контентной фильтрации // Информатика, вычислительная техника и инженерное образование. – 2019. – № 1 (34).
4. *Bova V., Kureichik V., Leshchanov D.* The model of semantic similarity estimation for the problems of big data search and structuring // 11th IEEE International Conference AICT 2017. – P. 27-32.
5. *Zhang W., Tang X., Yoshida T.* TESC: an approach to TExt classification using Semi-supervised Clustering // Knowledge-Based Systems. – 2015. – Vol. 75. – P. 152-160.
6. *Wei T., Lu Y., Chang H., Zhou Q., Bao X.* A semantic approach for text clustering using WordNet and lexical chains // Expert Systems with Applications. – 2015. – Vol. 42, No. 4. – P. 2264-2275.
7. *Xu D., Tian Y.* A Comprehensive Survey of Clustering Algorithms // Ann. Data. Sci. – 2015. – No. 2. – P. 165-193.
8. *Sabhia Firdaus, Md. Ashraf Uddin* A Survey on Clustering Algorithms and Complexity Analysis // International Journal of Computer Science Issues. – 2015. – Vol. 12. Issue 2. – P. 62-85.
9. *Sara Saad Soliman, Maged F. El-Sayed, Yasser F. Hassan* Semantic Clustering of Search Engine Results // The Scientific World Journal. – 2015. – Vol. 2015. Article ID 931258. – 9 p.
10. *Kravchenko Y.A., Rodzin S.I., Kuliev E.V., Bova V.V.* Simulation of the semantic network of knowledge representation in intelligent assistant systems based on ontological approach // Communications in Computer and Information Science this link is disabled. – 2021. – 1396 CCIS. – P. 241-252.
11. *Kravchenko Y., Bova V.* Assessment of ontological structures semantic similarity based on a modified cuckoo search algorithm // IOP Conference Series: Materials Science and Engineering. – 2020. – No. 12018.
12. *Отраднов К.К., Раев В.К.* Экспериментальное исследование эффективности методик векторизации текстовых документов и алгоритмов их кластеризации // Вестник РГРТУ. – 2018. – № 64. – С. 73-84.
13. *Zhou S., Xu X., Liu Y., Chang R., Xiao Y.* Text similarity measurement of semantic cognition based on word vector distance decentralization with clustering analysis // IEEE Access. – 2019. – Vol. 7. – P. 107247-107258.
14. *Krömer P., Platoš J.* Cluster analysis of data with reduced dimensionality: an empirical study // Intelligent Systems for Computer Modelling. – Springer. Cham, 2016. – P. 121-132.
15. *Бова В.В., Щеглов С.Н., Лещанов Д.В.* Модифицированный алгоритм EM-кластеризации для задач интегрированной обработки больших данных // Известия ЮФУ. Технические науки. – 2018. – № 4 (165). – С. 197-211.

16. *Jingdong Yan, Wuwei Liu*. An Ensemble Clustering Approach (Consensus Clustering) for High-Dimensional Data // Security and Communication Networks. – 2022. – Vol. 2022. Article ID 5629710. – 9 p.
17. *Olson C.F.* Parallel algorithms for hierarchical clustering // Pattern Analysis & Machine Intelligence IEEE Transactions on. – 2016. – Vol. 12, No. 11. – P. 1088-1092.
18. *Xueli X.U., Xuejing Z.* Application of sparse spectral clustering algorithm in high-dimensional data // Journal of University of Science and Technology of China. – 2017. – Vol. 47. – P. 311-319.
19. *Пархоменко П.А., Григорьев А.А., Астраханцев Н.А.* Обзор и экспериментальное сравнение методов кластеризации текстов // Тр. ИСП РАН. – 2017. – Т. 29. – Вып. 2. – С. 161-200.
20. *Guha S., Rastogi R. Shim K*. CURE: an efficient clustering algorithm for large databases // ACM SIGMOD. Rec. – 2017. – Vol. 27. – P. 73-84.
21. *Guha S., Rastogi R. Shim K.* ROCK: a robust clustering algorithm for categorical attributes // Proceedings of the 15th international conference on data engineering. – 2016. – P. 512-521.
22. *Karypis G. Han E. Kumar V.* Chameleon: hierarchical clustering using dynamic modeling // ACM SIGMOD Rec. – Aug. 1999. – Vol. 32. – P. 68-75.
23. *Huu Hiep Nguyen.* Clustering Categorical Data Using Community Detection Techniques // Computational Intelligence and Neuroscience. – 2017. – Vol. 2017. Article ID 8986360. – 11 p.
24. *Yancheng He, Qingcai Chen, Xiaolong Wang, Ruifeng Xu*. An adaptive affinity propagation document clustering // The 7th International Conference on Informatics and Systems. – 2010.
25. *Laurens van der Maaten, Geoffrey Hinton* Visualizing data using t-SNE // Journal of Machine Learning Research. – 2008. – No. 9. – P. 2579-2605.
26. *Van Eck N.J., Waltman L*. Software survey: VOSviewer, a computer program for bibliometric mapping // Scientometrics. – 2015. – Vol. 84, No. 2. – P. 523-538.
27. *Бова В.В., Лещанов Д.В.* Метод оценки эффективности семантической кластеризации гипертекстовых динамических структур на основе DOM-фильтра // Тр. конгресса IS&IT. – 2018. – Т. 2. – С. 59-70.

REFERENCES

1. *Junkai Yi, Yacong Zhang, Xianghui Zhao, Jing Wan* A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network, *Mathematical Problems in Engineering*, 2017, Vol. 2017, 13 p.
2. *Yujia Sun, Jan Platoš.* High-Dimensional Text Clustering by Dimensionality Reduction and Improved Density Peak, *Wireless Communications and Mobile Computing*, 2020, Vol. 2020. Article ID 8881112, 16 p.
3. *Bova V.V., Kuliev E.V., Shcheglov S.N.* Metod semanticheskoy klasterizatsii raspredelennykh resursov znaniy s dinamicheskimi komponentami na osnove kontentnoy fil'tratsii [The method of semantic clustering of distributed knowledge resources with dynamic components based on content filtering], *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Informatics, computer engineering and engineering education], 2019, No. 1 (34).
4. *Bova V., Kureichik V., Leshchanov D.* The model of semantic similarity estimation for the problems of big data search and structuring, *11th IEEE International Conference AICT 2017*, pp. 27-32.
5. *Zhang W., Tang X., Yoshida T.* TESC: an approach to TExt classification using Semi-supervised Clustering, *Knowledge-Based Systems*, 2015, Vol. 75, pp. 152-160.
6. *Wei T., Lu Y., Chang H., Zhou Q., Bao X.* A semantic approach for text clustering using WordNet and lexical chains, *Expert Systems with Applications*, 2015, Vol. 42, No. 4, pp. 2264-2275.
7. *Xu D., Tian Y.* A Comprehensive Survey of Clustering Algorithms, *Ann. Data. Sci.*, 2015, No. 2, pp**.** 165-193.
8. *Sabhia Firdaus, Md. Ashraf Uddin* A Survey on Clustering Algorithms and Complexity Analysis, *International Journal of Computer Science Issues*, 2015, Vol. 12. Issue 2, pp. 62-85.
9. *Sara Saad Soliman, Maged F. El-Sayed, Yasser F. Hassan* Semantic Clustering of Search Engine Results, *The Scientific World Journal*, 2015, Vol. 2015. Article ID 931258, 9 p.
10. *Kravchenko Y.A., Rodzin S.I., Kuliev E.V., Bova V.V.* Simulation of the semantic network of knowledge representation in intelligent assistant systems based on ontological approach, *Communications in Computer and Information Science this link is disabled*, 2021, 1396 CCIS, pp. 241-252.

11. *Kravchenko Y., Bova V.* Assessment of ontological structures semantic similarity based on a modified cuckoo search algorithm, *IOP Conference Series: Materials Science and Engineering*, 2020, No. 12018.
12. *Otradnov K.K., Raev V.K.* Eksperimental'noe issledovanie effektivnosti metodik vektorizatsii tekstovykh dokumentov i algoritmov ikh klasterizatsii [Experimental study of the effectiveness of methods of vectorization of text documents and algorithms for their clustering], *Vestnik RGRTU* [Vestnik of RSREU], 2018, No. 64, pp. 73-84.
13. *Zhou S., Xu X., Liu Y., Chang R., Xiao Y.* Text similarity measurement of semantic cognition based on word vector distance decentralization with clustering analysis, *IEEE Access*, 2019, Vol. 7, pp. 107247-107258.
14. *Krömer P., Platoš J.* Cluster analysis of data with reduced dimensionality: an empirical study, *Intelligent Systems for Computer Modelling*. Springer. Cham, 2016, pp. 121-132.
15. *Bova V.V., Shcheglov S.N., Leshchanov D.V.* Modifitsirovannyy algoritm EM-klasterizatsii dlya zadach integrirovannoy obrabotki bol'shikh dannykh [Modified EM clustering algorithm for integrated big data processing tasks], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2018, No. 4 (165), pp. 197-211.
16. *Jingdong Yan, Wuwei Liu.* An Ensemble Clustering Approach (Consensus Clustering) for High-Dimensional Data, *Security and Communication Networks*, 2022, Vol. 2022. Article ID 5629710, 9 p.
17. *Olson C.F.* Parallel algorithms for hierarchical clustering, *Pattern Analysis & Machine Intelligence IEEE Transactions on*, 2016, Vol. 12, No. 11, pp. 1088-1092.
18. *Xueli X.U., Xuejing Z.* Application of sparse spectral clustering algorithm in high-dimensional data, *Journal of University of Science and Technology of China*, 2017, Vol. 47, pp. 311-319.
19. *Parkhomenko P.A., Grigor'ev A.A., Astrakhantsev N.A.* Obzor i eksperimental'noe sravnenie metodov klasterizatsii tekstov [Review and experimental comparison of text clustering methods], *Tr. ISP RAN* [Proceedings of ISP RAS], 2017, Vol. 29, Issue 2, pp. 161-200.
20. *Guha S., Rastogi R. Shim K.* CURE: an efficient clustering algorithm for large databases, *ACM SIGMOD. Rec.*, 2017, Vol. 27, pp. 73-84.
21. *Guha S., Rastogi R. Shim K.* ROCK: a robust clustering algorithm for categorical attributes, *Proceedings of the 15th international conference on data engineering*, 2016, pp. 512-521.
22. *Karypis G. Han E. Kumar V.* Chameleon: hierarchical clustering using dynamic modeling, *ACM SIGMOD Rec.*, Aug. 1999, Vol. 32, pp. 68-75.
23. *Huu Hiep Nguyen.* Clustering Categorical Data Using Community Detection Techniques, *Computational Intelligence and Neuroscience*, 2017, Vol. 2017. Article ID 8986360, 11 p.
24. *Yancheng He, Qingcai Chen, Xiaolong Wang, Ruifeng Xu.* An adaptive affinity propagation document clustering, *The 7th International Conference on Informatics and Systems*, 2010.
25. *Laurens van der Maaten, Geoffrey Hinton* Visualizing data using t-SNE, J*ournal of Machine Learning Research*, 2008, No. 9, pp. 2579-2605.
26. *Van Eck N.J., Waltman L.* Software survey: VOSviewer, a computer program for bibliometric mapping, *Scientometrics*, 2015, Vol. 84, No. 2, pp. 523-538.
27. *Bova V.V., Leshchanov D.V.* Metod otsenki effektivnosti semanticheskoy klasterizatsii gipertekstovykh dinamicheskikh struktur na osnove DOM-fil'tra [A method for evaluating the effectiveness of semantic clustering of hypertext dynamic structures based on a DOM filter], *Tr. kongressa IS&IT* [Proceedings of the Congress IS&IT], 2018, Vol. 2, pp. 59-70.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Гатчин.

Бова Виктория Викторовна – Южный федеральный университет; e-mail: vvbova@sfedu.ru; г. Таганрог, Россия; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

Кравченко Юрий Алексеевич – e-mail: yakravchenko@sfedu.ru; кафедра систем автоматизированного проектирования; доцент.

Родзин Сергей Иванович – e-mail: srodzin@sfedu.ru; тел.: 88634371673; кафедра МОП ЭВМ; профессор.

Bova Victoria Victorovna – Southern Federal University; e-mail: vvbova@sfedu.ru; Taganrog, Russia; phone: +78634371651; the department of computer aided design; associate professor.

Kravchenko Yury Alekseevich – e-mail: yakravchenko@sfedu.ru; the department of computer aided design; associate professor.

Rodzin Sergey Ivanovich – e-mail: srodzin@sfedu.ru; phone: +78634371673; the department of software engineering; professor.

УДК 004.896 DOI 10.18522/2311-3103-2022-4-143-157

Б.К. Лебедев, О.Б. Лебедев, Е.О. Лебедева

ЭВОЛЮЦИОННЫЙ ПОПУЛЯЦИОННЫЙ МЕТОД РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ*

Рассматривается эволюционный популяционный метод решения транспортной задачи на основе метаэвристики кристаллизации россыпи альтернатив. Исследуется закрытая (или сбалансированная) модель транспортной задачи: сумма груза у поставщиков равно общей сумме потребностей в пунктах назначения. Цель оптимизации – минимизация стоимости (достижение минимума затрат на перевозку) или расстояний и критерий времени (затрачивается минимум времени на перевозку). В основу метаэвристики кристаллизации россыпи альтернатив положена стратегия, основанная на запоминании и повторении прошлых успехов. Стратегия делает упор на «коллективную память», под которой подразумевается любой вид информации, которая отражает прошлую историю развития и хранится независимо от индивидуумов. В качестве кода решения транспортной задачи рассматривается упорядоченная последовательность D_k маршрутов. Объектами являются маршруты, альтернативами – множество позиций P в списке, где n_p – число позиций в списке $D_к$. Множество объектов $D_к$ соответствует множеству всех маршрутов. Множество альтернативных состояний P объекта соответствует множеству альтернативных вариантов размещения объекта списке $D_к$. Работа популяционного эволюционного алгоритма кристаллизации россыпи альтернатив опирается на коллективную эволюционную память, называемую россыпью альтернатив. Под россыпью альтернатив решения в работе называется структура данных, используемая в качестве коллективной эволюционной памяти, несущая информацию о решении, включающую сведения о реализованных альтернативах агентов в данном решении и о полезности решения. Разработан конструктивный алгоритм формирования опорного плана путем декодирования списка $D_к$. На каждом шаге t решается задача выбора очередного в последовательности D_k маршрута и определения количества груза, перевозимого из пункта отправления A_i в пункт назначения B_j по этому маршруту. Разработанный алгоритм является популяционным, реализующим стратегию случайного направленного поиска. Каждый агент является кодом некоторого решения транспортной задачи. На первом этапе каждой итерации l конструктивным алгоритмом на базе интегральной россыпи альтернатив формируется n_k кодов решений D_k.Формирование каждого кода решения D_k выполняется последовательно по шагам путем последовательного выбора объекта и позиции. Для построенного кода решения D_k рассчитывается оценка решения ξ_k и оценка полезности δ_k. Формируется индивидуальная россыпь альтернатив R_k и переход к построению следующего кода решения. На втором этапе итерации производится суммирования интегральной россыпи альтернатив, сформированной на предыдущих итерациях от l до (l-1), со всеми индивидуальными россыпями альтернатив, сформированных на итерации l. На третьем этапе итерации l производится снижение всех интегральных оценок полезности $r^*_{\alpha\beta}$ интегральной россыпи альтернатив $R^*(l)$ на величину δ^*. Алгоритм решения транспортной задачи был реализован на языке C++ в среде Windows. Сравнение значений критерия, на тестовых примерах, с

* Работа выполнена при финансовой поддержке гранта РФФИ № 20–07–00260 А.