DOI 10.37882/2223-2966.2024.7-2.37

ИССЛЕДОВАНИЕ ТОЧНОСТИ РАБОТЫ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ ТЕКСТОВ, НАПИСАННЫХ НА ЕСТЕСТВЕННЫХ ЯЗЫКАХ[1]

STUDY OF THE ACCURACY OF CLUSTERING ALGORITHMS FOR TEXTS WRITTEN IN EUROPEAN LANGUAGES

M. Khairov
D. Sabirova
D. Novikova

Summary. This paper is devoted to investigating the problem of evaluating the accuracy of text clustering. To conduct the research, an expertly labeled dataset of 1800 texts was created, divided into three topics: IT innovations, education, and politics, as well as by text size.
The research included the steps of text processing, building vector models and applying different clustering algorithms such as K-means, Affinity Propagation and DBScan.
The results showed that K-means and Affinity Propagation algorithms achieved good results in text clustering accuracy (82 % and 85 %, respectively), while DBScan showed low accuracy (52 %) due to data features. In addition, K-means outperformed the other algorithms in terms of clustering completeness, showing 78 %.

Keywords: text clustering, text vector models, TF-IDF, K-means, Affinity Propagation, DBScan, clustering accuracy.

Хайров Марат Русланович
педагог ДО, Российский университет дружбы народов им. Патриса Лумумбы г. Москва
khayrov_mr@pfur.ru
Сабирова Динара Илхомовна
Лаборант, МИРЭА — Российский технологический университет
г. Москва
khayrov_mr@pfur.ru
Новикова Дарья Сергеевна
старший педагог ДО, Российский университет дружбы народов им. Патриса Лумумбы г. Москва
khayrov_mr@pfur.ru

Аннотация. Данная работа посвящена исследованию проблемы оценки точности кластеризации текстов. Для проведения исследований был создан размеченный экспертами датасет из 1800 текстов, разделенных на три тематики: IT инновации, образование и политика, а также по размерам текстов.
Исследование включало этапы обработки текстов, построения векторных моделей и применение различных алгоритмов кластеризации, таких как K-means, Affinity Propagation и DBScan.
Результаты показали, что алгоритмы K-means и Affinity Propagation достигли хороших результатов в точности кластеризации текстов (соответственно 82 % и 85 %), в то время как DBScan демонстрировал низкую точность (52 %) из-за особенностей данных. Кроме того, K-means превзошел другие алгоритмы по полноте кластеризации, показав 78 %.

Ключевые слова: кластеризация текстов, векторные модели текстов, TF-IDF, K-means, Affinity Propagation, DBScan, точность кластеризации.

Введение

Рубрикация и кластеризация текстов являются открытой проблемой 21-го века в области аналитики и машинного обучения. Кластеризация (рубрикация) текстов необходима почти во всех сферах жизнедеятельности, где используется тексты, написанные на естественном языке.

Кластеризация текста является важным инструментом в информационном поиске, для рекомендательных систем и области анализа настроений пользователей. Путем объединения схожих документов в кластеры поисковые системы предоставляют пользователям более точные и разнообразные результаты поиска. Кластеризация также позволяет персонализировать рекомендации и анализировать большие объемы текстовых данных для выявления тенденций.

Несмотря на их эффективность, применение алгоритмов кластеризации в области текстовой аналитики сопряжено с ограничениями, такими как чувствительность к выбросам и необходимость выбора подходящих алгоритмов и параметров в зависимости от характера данных.

[1] Работа выполнена при финансовой поддержке Российского научного фонда (РНФ), грант № 23–21-00153 «Анализ и моделирование динамики нестационарных временных рядов фрактальных процессов с реализацией памяти (последействия) и самоорганизацией на основе использования дифференциальных уравнений с дробными производными».

Литературный обзор

Существуют различные подходы к алгоритмам кластеризации. [1], пожалуй, наиболее популярными являются партиционный, иерархический и плотностной подходы. Алгоритмы, использующие условный подход, производят набор из K кластеров, не совпадающих друг с другом $S = \{S_1, S_2,\ldots,S_k\}$ сумма кардинальности которых равна кардинальности самого набора данных — то есть, $USl \in SSl \vee n$.

Иерархические алгоритмы идут дальше, получая также информацию о взаимосвязях между кластерами, что обычно требует больших вычислительных затрат.

Алгоритмы, основанные на плотности, определяют кластеры как области с более высокой плотностью, что позволяет создавать кластеры произвольной формы [2].

Рассматривая направление кластеризации данных, нельзя не затронуть тему построение векторной модели данных для дальнейшего использования ее в алгоритмах кластеризации.

Одной из самых известных векторных моделей данных является модель TF-IDF. При использовании алгоритма TF-IDF признаковому слову присваивается вес, основанный на том, насколько часто оно встречается в наборе документов.

Однако у данного алгоритма есть некоторые недостатки, которые разобрал в своей научной статье Qiangyi Li [3]. В ней он выделил три основных недостатка данной модели:

1. отсутствие информации о распределении внутри категории;
2. нет информации о распределении категорий. Алгоритм не учитывает распределение характерных слов по категориям при определении весов;
3. невозможность адаптации к перекошенным наборам данных. В коллекции документов количество документов в каждой категории почти никогда не бывает одинаковым.

В данной статье был предложен метод для подсчета IDF, где автор предлагает заменить ее на формулу.

$$IDF = -\log\left(1 - \frac{F(m_i)}{F(m_i) + F(o_i)}\right) = \log\left(1 + \frac{F(m_i)}{F(o_i)}\right) \quad (1)$$

В статье [4] авторы Ashwini K.S. и Shantala C.P. рассмотрели все эти вышеуказанные модели и произвели исследование точности кластеризации алгоритмами K-means. Более того, для авторы использовали метод локтя для подбора оптимального значения K.

Данный метод заключается в определении значения K, при котором при визуализации функции ошибки SSE происходит перегиб функции, после которого функция линейно уменьшается [5, 6].

Также стоит отметить, что евклидово расстояние, также известное как евклидова метрика, является часто используемой мерой расстояния в алгоритмах кластеризации, включая кластеризацию текста, и является мерой расстояния по умолчанию в алгоритме K-means. [7]

В другом исследовании [8] Fitri Andriyani и Yan Puspitatani исследуют точность кластеризации алгоритмами DBSCAN и K-means.

Методы предварительной обработки, используемые в данном исследовании — это токенизация, сложение регистров, удаление стоп-слов и стемминг. После предварительной обработки текстов были получены векторные модели TF-IDF данных отзывов. Кроме того, исследования точности кластеризации алгоритма DBScan проводилось путем подбора оптимального значения Eps.

Данные для анализа и подготовка векторных моделей

Для проведения исследования алгоритмов кластеризации был собран датасет текстов, состоящий из 1800 текстов, который в свою очередь был разбит на 600 текстов по 3-м тематикам:

1. 600 текстов на тему: IT инновации;
2. 600 текстов на тему: образование;
3. 600 текстов на тему: политика.

Кроме того, каждая тема разбита по 200 текстов на 3 группы:

1. 200 больших текстов;
2. 200 средних текстов;
3. 200 маленьких текстов.

Исследование началось с внедрения инструментов обработки естественного языка. Главная задача которого заключалась в очистке текстовых данных через процессы токенизации, нормализации и удаления мусора.

Токенизация разделяет большое количество текста на более мелкие фрагменты, известные как токены (предложения, слова).

Нормализация текста заключается в приведении каждого слова к его базовой форме. В русском языке базовыми формами считаются следующие морфологические формы:

1. для существительных — именительный падеж, единственное число;
2. для прилагательных — именительный падеж, единственное число, мужской род;

3. для глаголов, причастий, деепричастий — глагол в инфинитиве (неопределённой форме) несовершенного вида.

Для программной реализации данного этапы были применены готовые решения. В качестве основы была выбрана библиотека pymorphy2, а также библиотека nltk, из которой был взят словарь стоп-слов.

После обработки текстов необходимо было построить векторные модели текстов. Была выбрана векторная модель TF-IDF (частота слов — обратная частота документов).

Метод TF-IDF выделяет важность слова, основываясь на его частоте в конкретном документе и в других документах. Это позволяет задавать больший вес редко встречающимся словам по сравнению с часто встречающимися. Формула TF-IDF представлена в уравнении 2. [1, 9]

$$tf - idf(t,d,D) = tf(t,d) * idf(t,D) \quad (2)$$

где t — слово, d — конкретный документ, D — коллекция документов.

По формуле 3 можно рассчитать частоту вхождения слова в документ на обратную частоту документа.

$$tf - idf(t,d,D) = \frac{n_t}{\sum_k n_k * \log_{10} \frac{|D|}{\{d_i \in D | t \in d_i\}}} \quad (3)$$

где t — слово, d — конкретный документ, D — коллекция документов, n_t — количество вхождений слова в документ, $\sum_k n_k$ — общее количество слов в документе, $|D|$ — количество документов, $\{d_i \in D | t \in d_i\}$ — количество документов, в которых встречается слово t.

Алгоритм DBSCAN

Данный алгоритм кластеризации группирует тесно лежащие точки в некотором пространстве. [10]

На вход алгоритма поступает 2 значения:

1. ε-окрестность — это радиус окружности, который задается для нахождения соседей вокруг точки;
2. minPts — минимальное количество соседей, для добавления данной точки в список основных точек.
3. На рисунке 1 схематически показана ε-окрестность точки. Все точки, находящиеся в пределах ε-окрестности будут являться соседями данной точки.

Алгоритм [11] работает таким образом, что сначала проводится обход по всем точкам, а затем собираются

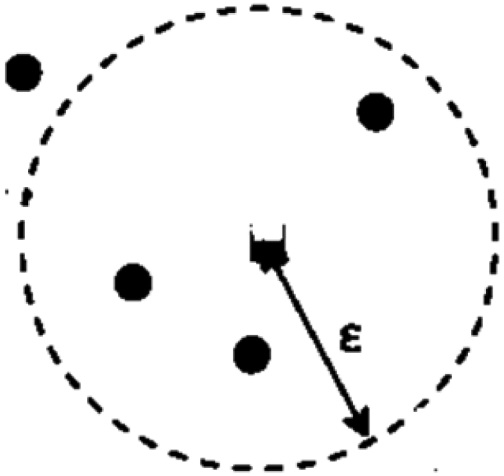

Рис. 1. ε-окрестность точки

соседи каждой точке. После этого создается список основных точек, в который войдут только те точки, у которых количество соседей не меньше, чем minPts. Следом будет создан список неиспользованных точек, из которого далее удалятся те точки, которые объединятся в кластер.

Затем алгоритм случайным образом выбирает из основных точек одну, которая также есть в списке неиспользованных точек, создает для этой точки новый кластер и добавляет в этот кластер соседей основной точки. Если среди этих соседей находятся еще точки из списка основных точек, то в кластер добавляются и их соседи. Так происходит пока в соседях основных точек, которые попали в кластер, не останется ни одной точки из списка основных точек, не добавленных в этот кластер.

Процедура с выбором случайной точки из списка основных точек продолжается до тех пор, пока в списке основных точек не останется ни одной точки из списка неиспользованных.

Алгоритм Affinity Propagation

Алгоритм Affinity Propagation (AP) [12, 13] принадлежит к классу неиерархических методов кластеризации и не требует заранее заданного числа кластеров. Он итеративно выбирает образцы из входных данных, которые наиболее репрезентативны для кластера, и максимизирует сходство между объектами и выбранными образцами для поиска кластеров.

Алгоритм получает два набора данных:

1. матрицу расстояний между объектами;
2. значения предпочтений, которые отражают степень пригодности каждого документа в качестве центра кластера.

Оба набора данных объединяются в матрицу расстояний S размерности $n \times n$, где n — количество объектов в коллекции.

Главная диагональ матрицы S соответствует предпочтениям. Расстояния могут быть опущены или установлены как минус бесконечность, если объекты не подходят для образования одного кластера.

Алгоритм кластеризации выполняется итерационно, каждая итерация состоит из двух шагов:

1. на первом шаге происходит расчет матрицы «ответственности» — R (формула 3);
2. на втором шаге расчет матрицы «доступности» — A (формула 4).
3. Изначально матрицы R и A инициализируются нулями.

$$r(i,k) = s(i,k) - \left(a(i,k') + s(i,k')\right) \quad (4)$$

где i, k — рассматриваемые объекты, k'— остальные образцы k, кроме текущего.

$$\left\{a(i,k) = \min\left\{0, r(k,k) + \sum_{i \in \{i,k\}} \max\left(0, r(i,k)\right)\right\}\right. \quad (5)$$

для $i \neq k$, где i, k — рассматриваемые объекты

Итоговые образцы извлекаются по формуле 6 из матриц R и A:

$$c_i = \arg\left(a(i,k) + r(i,k)\right) \quad (6)$$

где i, k — рассматриваемые объекты, c_i — образец для объекта i.

Количество итоговых образцов равняется количеству кластеров, каждый образец является центром своего кластера.

Алгоритм k-means

K-means один из популярных алгоритмов кластеризации, который разбивает набор значений на k заданных кластеров, где каждое значение принадлежит к кластеру с ближайшим средним значением (центроиду). [14]

Данный алгоритм работает следующим образом:

1. алгоритм начинает с первоначального предположения о центрах кластеров. Эти центроиды обычно выбираются случайным образом;
2. каждый объект относится к кластеру, центроид которого находится ближе всего к нему. Для вычисления расстояния между объектами применяется евклидово расстояние (формула 7);
3. центроиды кластеров обновляются, путем нахождения среднего объекта (центра) в уже сформированном кластере (формула 8).

$$d = \sum_{i}^{N_A} \left(A_{i_2} - A_{i_1}\right)^2 \quad (7)$$

$$\left(a';b';\ldots;z'\right) = \left(\frac{\sum_{i}^{n_a} a_i}{n_a}; \frac{\sum_{i}^{n_b} b_i}{n_b}; \ldots; \frac{\sum_{i}^{n_z} z_i}{n_z}\right) \quad (8)$$

Шаги 2 и 3 повторяются до того момента, пока не будут распределены все точки по кластерам и центроиды не перестанут изменяться.

Данный алгоритм имеет множество недостатков, самый главный из них, это то, что он предполагает, что кластеры сферические и имеют одинаковую форму, тем самым он очень чувствителен к выбросам, так как из-за них центроиды кластеров будут смещены от оптимального значения, тем самым кластеры будут сформированы неточно. И не менее важно, что количество кластеров, которые необходимо сформировать пользователь выбирает сам.

Исследование точности и полноты кластеризации

Перед началом проведения исследования и кластеризации текстов необходимо определить, что будет считаться точностью и полнотой кластеризации.

В данном исследовании точность кластеризации рассчитывалась, как количество элементов одного кластера, правильно кластеризованных (распределенных) в этот кластер, деленное на общее количество элементов в данном кластере.

Полнота была рассчитана для каждого кластера отдельно, где полнота кластера — это количество правильно кластеризованных элементов данного кластера деленное на общее количество элементов кластера. После чего полученные значения полноты складываются и делятся на количество кластеров, составляющих один большой кластер.

Данный расчет необходим из-за того, что после кластеризации алгоритмами DBSCAN и Affinity-Propagation количество кластеров значительно больше изначального.

Результаты кластеризации текстов малого размера алгоритмами DBSCAN, Affinity Propagation и K-means представлены на рисунках 2, 3, 4, а все результаты исследования приведены в таблице 1.

Для алгоритма DBScan Значение ε-окрестности и minPts равны 2,6 и 3 соответственно, а для алгоритма K-means, $k = 3$.

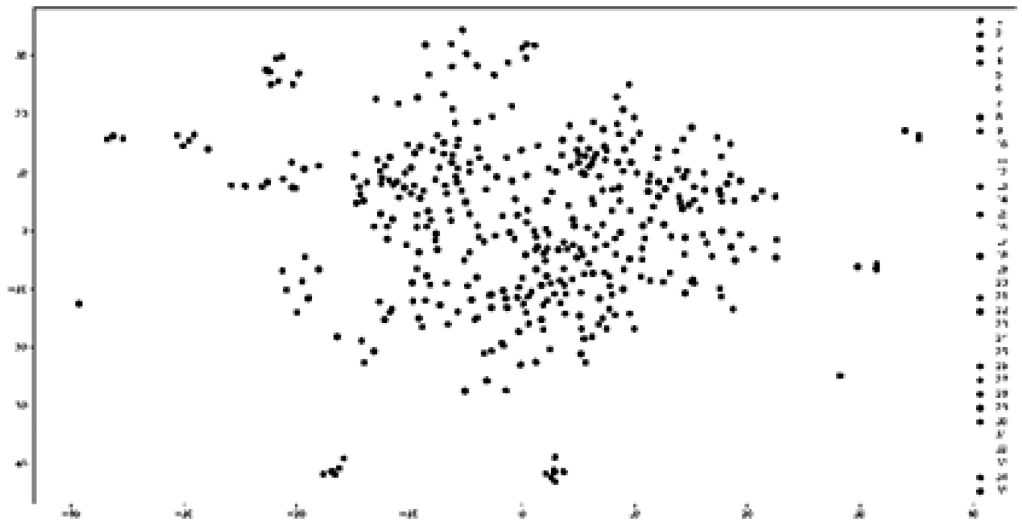

Рис. 2. Визуализация результатов кластеризации с использованием алгоритма DBScan на текстах малого размера

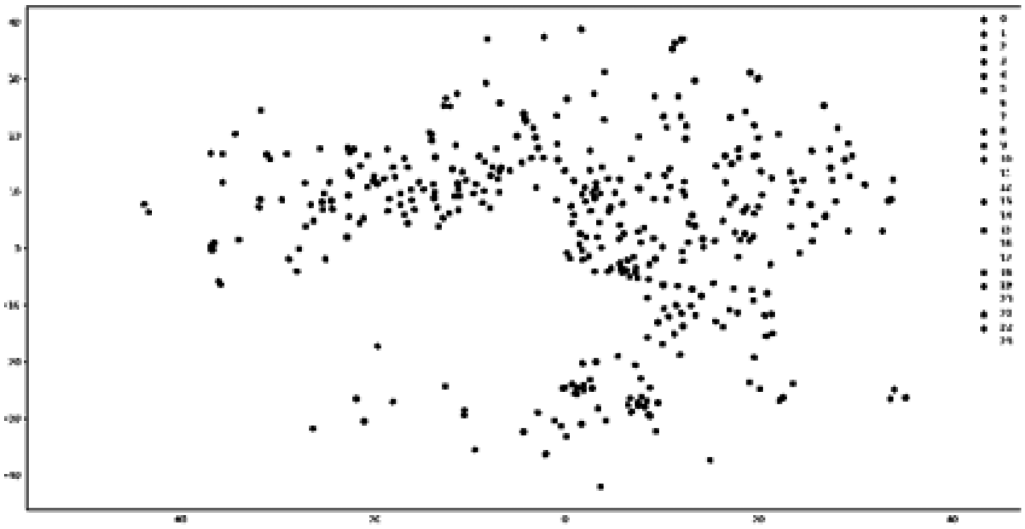

Рис. 3. Визуализация результатов кластеризации с использованием алгоритма Affinity Propagation на текстах малого размера

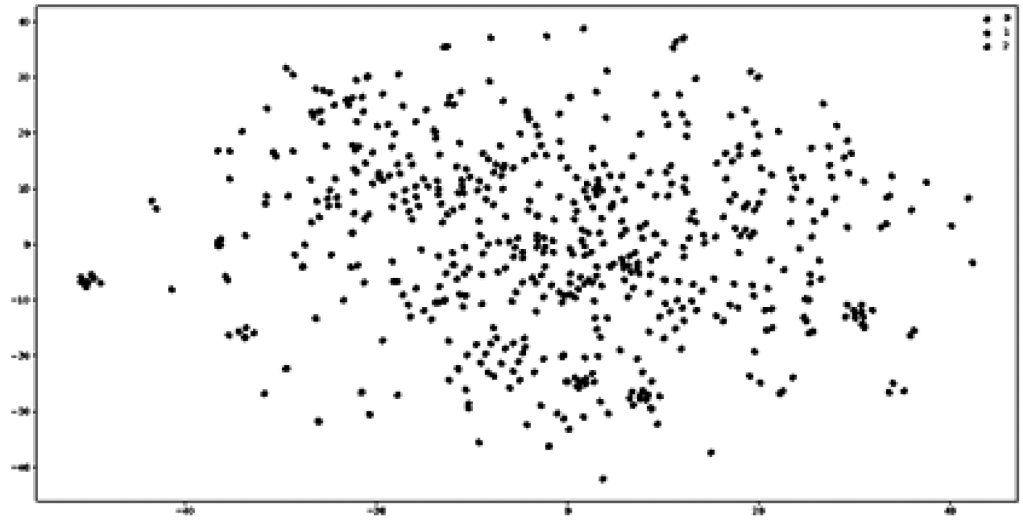

Рис. 4. Визуализация результатов кластеризации с использованием алгоритма K-means на текстах малого размера

Таблица 1.

Результаты кластеризации

Дата-сет	Тексты	точность / полнота	Алгоритмы		
			DBScan	Affinity Propagation	K-means
Маленькие	IT инновации	точность	0,82	0,91	0,74
		полнота	0,08	0,11	0,79
	Образование	точность	0,22	0,69	0,86
		полнота	0,02	0,10	0,76
	Политика	точность	0,45	0,86	0,85
		полнота	0,03	0,10	0,86
Средние	IT инновации	точность	0,85	0,83	0,79
		полнота	0,11	0,10	0,78
	Образование	точность	0,37	0,82	0,88
		полнота	0,03	0,12	0,80
	Политика	точность	0,48	0,96	0,88
		полнота	0,04	0,11	0,83
Большие	IT инновации	точность	0,86	0,86	0,83
		полнота	0,11	0,11	0,80
	Образование	точность	0,33	0,80	0,64
		полнота	0,03	0,09	0,52
	Политика	точность	0,31	0,92	0,97
		полнота	0,02	0,10	0,48

Исходя из полученных результатов можно сказать, что алгоритмы K-means и Affinity Propagation показали хорошие результаты на текстах всех размерностей.

Алгоритм DBScan показал плохой результат, и это можно объяснить тем, что полученные изначально вектора не совсем пригодны для кластеризации данным алгоритмом, ведь плотность скопления данных во всех местах примерно одинаковая, что делает практически невозможным применение данного алгоритма.

Для K-means же получилась очень благоприятная ситуация, ведь наши изначальные данные в некоторой степени напоминали круги, а данный алгоритм формирует сферические кластеры.

Заключение

В результате данного исследования можно сделать выводы, что алгоритмы кластеризации K-means и Affinity Propagation показали хорошие результаты по точности кластеризации текстовых данных.

Средняя точность кластеризации алгоритмом Affinity Propagation — 85 %, в то время как у алгоритма K-means 81.5 %, алгоритм же DBScan в среднем кластеризует текстовые данные с точностью 52 %.

Но если рассмотреть алгоритмы в плане полноты кластеризации, то в этом аспекте алгоритм K-means превзошел алгоритмы DBScan и Affinity Propagation в несколько раз. Средняя полнота кластеризации алгоритмом K-means — 78 %.

Исходя из этих данных можно сделать вывод, что для кластеризации текстовой информации лучше всего использовать алгоритм K-means или Affinity Propagation, но все же стоит отметить, что если разработать алгоритм, который будет объединять схожие кластеры в один, то результат полноты у Affinity Propagation будет не меньше, чем у K-means.

ЛИТЕРАТУРА

1. Go Machine Learning Projects / Oreilly, by Xuanyi Chew, Released November 2018, Publisher(s): Packt Publishing, ISBN: 9781788993401
2. Stiphen Chowdhury, Na Helian, Renato Cordeiro de Amorim Feature weighting in DBSCAN using reverse nearest neighbours // Pattern Recognition, January 2023
3. Lin Xiang, Application of an Improved TF-IDF Method in Literary Text Classification // Advances in Multimedia, May 2022
4. Ashwini K.S., Shantala C.P., Tony Jan, Impact of Text Representation Techniques on Clustering Models // Fabuary 22.02.2022
5. Hasugian P.M. et al. Best cluster optimization with combination of K-means algorithm and elbow method towards rice production status determination // International Journal of Artificial Intelligence Research. — 2021. — Т. 5. — №. 1. — С. 102–110.
6. Jaafar B.A., Gaata M.T., Jasim M.N. Home appliances recommendation system based on weather information using combined modified k-means and elbow algorithms //Indonesian Journal of Electrical Engineering and Computer Science. — 2020. — Т. 19. — №. 3. — С. 1635–1642.
7. Majid Hamed Ahmed, Sabrina Trun, Nor S Sani, Nazila Omar, Short Text Clustering Algorithms, Application and Challenges: A Survey. // Applied Sciences December 2022
8. Fitri Andriyani, Yan Puspitarani, Performance Comparison of K-Means and DBScan Algorithms for Text ClusteringProduct Reviews. // SinkrOn, July 2022
9. Natural Language Processing in Action video edition [Электронный ресурс]. — URL: https://www.oreilly.com/library/view/natural-language-processing/9781617294631VE/ (дата обращения 20.04.2024).
10. Birant D., Kut A. ST-DBSCAN: An algorithm for clustering spatial–temporal data //Data & knowledge engineering. — 2007. — Т. 60. — №. 1. — С. 208–221.
11. Khan K. et al. DBSCAN: Past, present, and future //The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014). — IEEE, 2014. — С. 232–238.
12. Wang K. et al. Adaptive affinity propagation clustering //arXiv preprint arXiv:0805.1096. — 2008.
13. Shang F. et al. Fast affinity propagation clustering: A multilevel approach //Pattern recognition. — 2012. — Т. 45. — №. 1. — С. 474–486.
14. Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values //Data mining and knowledge discovery. — 1998. — Т. 2. — №. 3. — С. 283–304.

© Хайров Марат Русланович (khayrov_mr@pfur.ru); Сабирова Динара Илхомовна (khayrov_mr@pfur.ru);
Новикова Дарья Сергеевна (khayrov_mr@pfur.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»