Задача эффективной кластеризации текстовой выборки в зависимости от различной параметризации этой выборки

Э. А. Головастова[I], Д.Н. Красотин[II]

[I] Московский государственный университет им. М.В. Ломоносова, г. Москва, Россия
[II]ЗАО “Московский научно-исследовательский телевизионный институт”, г. Москва, Россия

Аннотация. Данное исследование посвящено проблеме необходимости проведения быстрой и качественной автоматизированной кластеризации больших объемов текстовых выборок в условиях постоянно разрастающегося объёма информации, в том числе получаемых из сети Интернет. В статье рассмотрены различные способы параметризации текстовой выборки и различные алгоритмы кластеризации. Качество работы методов оценивалось по скорости их выполнения, значению коэффициента Силуэт (формальному показателю качества кластеризации) и полноты финального отображения кластеров. В статье приведены результаты работы методов кластеризации, проведен их анализ и сравнение.

Ключевые слова: Кластеризация, текстовая выборка, параметризация выборки, tf-idf-мера, ключевые слова, эффективный метод.

DOI 10.14357/20718632190406

Введение

В современном мире большая часть текстовой информации хранится, преимущественно, в сети Интернет. Поэтому появляется необходимость в быстрой автоматизированной обработке большого количества информационных ресурсов и объема данных. В частности, одной из важнейших задач автоматизированной обработки текстовой информации является группировка новостей по сюжетам или кластеризация новостного потока. Спектр применения кластерного анализа выборки очень велик: его используют в филологии, социологии, психологии, маркетинге и других дисциплинах. Существует достаточно много алгоритмов решения данной задачи [1-3]. Например, довольно популярны методы, где предлагается реализовать модель кластеризации новостного потока со сдвигаемым временным окном, в котором проводится предварительная кластеризация новостей, поступивших за некоторый период. Число сюжетов при этом заранее неизвестно и потенциально неограниченно. Такие модели могут учитывать изменения распределения данных с течением времени, поэтому они могут быть использованы для объяснения эволюции рассматриваемого процесса. Для применения какого-либо алгоритма кластеризации необходимо задать текстовую выборку некоторым числовым набором значений её признаков. Наиболее распространенным представлением является матрица с элементами, равными величине статистической меры, рассчитывающейся на основе частоты употребления слова [4]. Но, к примеру, для кластеризации новостного потока со сдвигаемым временным окном, нужно сохранять информацию о ранее созданных кластерах и текстах, поступивших задолго до рассматриваемого промежутка. Матрицы векторизированных объектов

зачастую громоздки, поэтому при хранении потребуют много памяти. Также векторизация новых сообщений предполагает пересчет некоторых параметров, зависящих от всех предыдущих текстов, что является вычислительно затратной операцией. Как решение этой проблемы предлагается хранить только ключевые слова кластеров и текстов.

В данной работе будет рассмотрена задача кластеризации текстовой выборки фиксированной длины. Цель данного исследования — поиск наиболее эффективного алгоритма для поставленной задачи в зависимости от различной параметризации текстов и метода кластеризации по значениям показателей быстродействия, точности и полноты финального отображения кластеров на различных объемах текстовых выборок, а также сравнение качества работы выбранных алгоритмов. Время выполнения измеряется не только для непосредственно кластеризации, но и для всех стадий алгоритмов. При этом проверялся результат применения дополнительных этапов. А именно: уменьшение размерности матрицы векторов этой выборки и фильтрация при итоговом отображении кластеров.

При визуализации полученных кластеров некоторые тексты игнорируются в пользу соответствия новостей теме кластера, так как необходимо, чтобы внутри него все новости были одной тематики. При этом мы следим за числом отброшенных элементов. Малое количество текстов при финальном отображении свидетельствует о том, что много информации из выборки потеряно. Такой факт будет считаться отрицательной характеристикой алгоритма. Точность соответствия новостей теме кластера оценивалась как по формальному показателю, так и по непосредственному просмотру их содержимого.

1. Обработка выборки, оценка качества, использованные методы

В работе использовалась неразмеченная выборка из нескольких тысяч элементов, каждый из которых является русскоязычным новостным текстом, к которому прибавлено предложение — его заголовок. Алгоритмы решения поставленной задачи реализованы на языке Python 3 [5].

Заголовки зачастую отражают основное содержание текста новости, поэтому мы увеличиваем меру соответствующих слов. Для улучшения качества кластеризации тексты предварительно стеммингуются [6]. В процессе числового представления текста исключаются как часто встречающиеся слова (как-то: союзы, предлоги и др.), так и редко встречающиеся (например, опечатки), в целях уменьшения влияния слов, не определяющих текст, на конечный результат параметризации.

Формально показатель точности кластеризации определяется с помощью силуэтного анализа [7], который использует саму модель кластеризации, и поэтому применяется, если истинные метки элементов выборки неизвестны.

Коэффициент силуэта рассчитывается для каждого кластера и зависит от двух показателей: a — среднего расстояния между текстом и всеми другими точками кластера, в котором он находится, b — среднего расстояния между текстом и всеми точками во всех других кластерах, которым этот текст не принадлежит. Коэффициент силуэта s для одного кластера задается как:

$$s = \frac{b - a}{\max(a, b)}.$$

Здесь и далее под расстоянием понимается евклидово расстояние:

$$dist(\vec{p}, \vec{q}) = \sqrt{(p_1 - q_1)^2 + ... + (p_n - q_n)^2}, \quad (1)$$

где $\vec{p} = (p_1, ..., p_n), \vec{q} = (q_1, ..., q_n)$ — некоторые n-мерные вектора.

Для группы кластеров коэффициент силуэта рассчитывается как среднее значение от всех коэффициентов для каждого кластера. Здесь более высокий показатель коэффициента силуэт относится к модели, лучше всего разделяющей выборку по сюжетам. Он соответствует 1, а для худшей модели — -1. Значения вблизи 0 указывают на то, что в выборке много элементов, каждый из которых можно отнести к разным кластерам.

При фильтрации набора данных текущей выборки из нее для каждого кластера убираются элементы, удаленные от центра этого кластера на евклидово расстояние (1), превышающее некоторое пороговое значение, которое было эмпирически подобрано. В процессе

тестирования пороговое значение для фильтрации было выбрано равным 0.7. Так убираются элементы, представляющие собой, главным образом, шум. То есть те элементы, которые либо не относятся ни к одному кластеру, либо, будучи отнесенными к некоторому кластеру, возможно, ему не принадлежат.

Также немаловажным является фактор скорости реализации, поэтому мы также проводим замер времени выполнения ключевых операций.

Тексты предварительно представляются tf-idf-матрицей. Для каждого слова t текста *d*:

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

здесь: $idf\ (t)=log\{(1+N)/(1+df(t))\}+1$ — инверсия частоты, с которой слово t встречается в текстах выборки; $df(t)$ — число текстов из выборки, где встречается слово *t;* $tf(t,d)$ — частотность слова *t* в тексте *d;* N — общее число текстов.

Максимальный размер словаря векторизатора — 15 000 слов.

Далее было сделано предположение, что уменьшение размерности векторов сократит время выполнения кластеризации без значительной потери качества. Мы проверим это предположение, сравнив качество работы алгоритмов с уменьшение размерности и без.

Для уменьшения размерности до значения в 100 раз меньшего исходного используется метод TruncatedSVD (далее TrSVD), реализованный в библиотеке scikit-learn [8], основывающийся на сингулярном разложении матрицы tf-idf-векторов текстов [9].

Так как новостные сообщения являются описаниями событий реального мира, поэтому основную информацию содержит его текстовое наполнение. Текст новости определяется своими ключевыми словами, которые есть особо важные, краткие, общепонятные слова в тексте, набор которых может дать для читателя высокоуровневое описание его содержания, обеспечив при этом компактное представление и хранение смысла текста на запоминающем устройстве. Учитывая это, в качестве альтернативного представления зададим тексты их ключевыми словами.

Для проведения непосредственно самой кластеризации были выбраны для сравнения два метода: K-Means и Dbscan, реализацию которых берем из библиотеки sklearn.cluster.

K-Means — один из наиболее популярных методов кластерного анализа, разбивающий множество элементов векторного пространства на заранее известное число кластеров. Основная цель алгоритма — минимизация среднеквадратичного отклонения на точках каждого кластера. На каждой итерации вычисляется центр каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются вновь на группы по принципу близости к новому центру по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения кластеров. В целях ускорения выполнения и снижения затрат памяти мы использовали модификацию алгоритма K-Means — MiniBatchKMeans. MiniBatchKMeans использует подмножества входных данных, выбирающиеся случайным образом в каждой итерации, — mini-batches — для сокращения времени вычислений, оптимизируя ту же целевую функцию, что и K-Means. Алгоритм итерирует два основных этапа, аналогично K-Means. На первом из выборки выбираются случайным образом данные, формирующие mini-batches. Затем они назначаются ближайшему центру. На втором этапе центры обновляются [10].

В качестве альтернативы рассмотрим Dbscan**Ошибка! Закладка не определена.** — плотностной алгоритм пространственной кластеризации с присутствием шума, не требующий задания числа кластеров [11-13]. Идея, лежащая в основе алгоритма, заключается в том, что внутри каждого кластера наблюдается определенная плотность объектов выборки, которая заметно выше, чем плотность вне кластера, а также плотность в областях с шумом ниже плотности любого кластера.

В тестируемой программе изначально кластеризация проводилась методом Dbscan, затем запускался метод K-Means с числом кластеров, равным числу точек, находящихся в областях наибольшей плотности данной выборки, если это число не превышало 500. Иначе число кластеров для K-Means полагалось равным 500.

51	Грузинская певица Нино Катамадзе отказалась от концертор в России из-за протестов в Тбилиси
40	Молочнные продукты без растительных жиров с 1 июля визуально выделяются в магазинах РФ
32	В северной Корее встречу Трампа и Ким Чен Ына назвали "исторической"
7	Трамп заявил, что на саммите "Большой двадцатки" встречался "с президентами диктаторами"

Рис. 1. Визуализация содержимого кластеров

На оба варианта параметризации текстов были написаны свои программы, обеспечивающие также визуализацию найденных кластеров (Рис. 1, Рис. 12). В случае параметризации выборки с помощью tf-idf-матрицы для каждого кластера определяются десять главных слов, его характеризирующих. Также определяется заголовок новости, который лучше других отражает тему сюжета. Содержимое кластера фильтруется по удаленности от центра, и при визуализации все кластера упорядочиваются по размеру (Рис. 1).

Также присутствует возможность объединения кластеров с помощью метрики Жаккара (коэффициентом сходства между двумя множествами A и B) по их ключевым словам, то есть возможность нахождения похожих сюжетов.

Метрика Жаккара рассчитывается следующим образом [14]:

$$J = |A \cap B| / |A \cup B|, \quad (3)$$

где A,B — некоторые множества, | A | — число элементов во множестве A. Из формулы (3) видно, что значение метрики Жаккара изменяется от 0 до 1 и достигает максимума при совпадении двух множеств.

2. Результаты для параметризации текстов tf-idf-матрицей

Для данного способа определения выборки алгоритм кластеризации состоял из следующих этапов: рассчитывалась tf-idf-матрица текстов, затем (по выбору) проводилось уменьшение размерности векторов, далее матрица выборки поочерёдно посылалась на оба метода кластеризации и полученные кластера визуализировались (по выбору) с предварительной фильтрацией. При тестировании алгоритма были получены следующие результаты (Табл. 1, Табл. 2)

В Табл. 1 коэффициент силуэта посчитан для нефильтрованных данных. То есть данные сильно зашумлены, и поэтому в этом случае сложно что-либо сказать о качестве кластеризации по формальному признаку.

Исходя из результатов Табл. 1 отметим, что применение TrSVD значительно снижает число найденных кластеров для алгоритма Dbscan, что является скорее негативным фактором, так как вполне возможно приведет к потери информации при финальном отображении сюжетов. Далее, применение TrSVD увеличивает скорость выполнения K-Means, но имеет противоположный эффект для алгоритма Dbscan: Dbscan работает намного быстрее на больших размерностях векторов текстов для определенных размеров выборки. Из результатов, представленных в Табл. 1, следует, что Dbscan дает значительное ускорение при объёмах выборки в пределах от 1000 до 15 000 текстов. При числе текстов более 20 000 он становится медленнее K-Means (Рис. 2, Рис. 3).

Отметим, что K-Means сходится медленнее, то есть требует больше итераций для сходимости, при применении TrSVD на размерах выборки менее 1500 текстов.

Также, естественно, применение TrSVD увеличит суммарное время выполнения всех этапов предварительной кластеризации. Как видно из Рис. 4 и Рис. 5, вся предварительная кластеризация при использовании алгоритма Dbscan проходит быстрее без применения TrSVD, и это время сравнимо на некоторых размерностях векторов текстов с результатами K-Means с предварительным уменьшением размерности с помощью TrSVD.

Табл. 1. Результаты работы алгоритмов при параметризации выборки tf-idf-матрицей без применения фильтрации

Без фильтрации										
					DBSCAN			K-Means		
TrSVD	Размер выборки	Размер вектора признаков	Время векторизации (в с.)	Время TrSVD (в с.)	Время кластеризации (в с.)	Силуэт	Число кластеров	Время кластеризации (в с.)	Силуэт	Число кластеров
Да	800	8980	15.22	0.33	0.21	0.01	28	1.81	0.06	3
Да	1000	10428	18.15	0.49	0.31	0.00	27	2.44	0.06	13
Да	5000	15000	83.19	2.10	8.99	-0.01	29	9.45	0.04	282
Да	10000	15000	166.39	3.34	53.78	-0.01	19	14.79	0.03	500
Да	15000	15000	260.27	4.66	138.11	-0.02	15	11.51	0.03	500
Да	20000	15000	336.72	5.94	216.38	-0.02	10	12.32	0.03	500
Да	22000	15000	368.61	6.43	364.18	0.00	9	11.45	0.02	500
Нет	800	8980	13.66	Нет	0.14	0.01	1	0.09	0.01	3
Нет	1000	10428	17.93	Нет	0.15	0.01	3	0.40	0.01	13
Нет	5000	15000	81.07	Нет	1.41	-0.01	58	35.10	0.02	282
Нет	10000	15000	168.80	Нет	5.17	-0.02	189	86.83	0.01	500
Нет	15000	15000	263.09	Нет	12.04	-0.01	325	95.90	0.02	500
Нет	20000	15000	340.33	Нет	214.37	-0.02	446	113.03	0.02	500
Нет	22000	15000	371.13	Нет	447.26	-0.01	507	105.38	0.01	500

Табл. 2. Результаты работы алгоритмов при параметризации выборки tf-idf-матрицей с применением фильтрации

С фильтрацией												
					DBSCAN				K-Means			
TrSVD	Размер выборки	Размер вектора признаков	Время векторизации (в с.)	Время TrSVD (в с.)	Время кластеризации (в с.)	Силуэт	Число отброшенных текстов	Число кластеров	Время кластеризации (в с.)	Силуэт	Число отброшенных текстов	Число кластеров
Да	800	8478	13.83	0.29	0.21	0.54	779	0	1.76	0.68	730	1
Да	1000	9518	16.88	0.40	0.32	0.53	967	1	2.49	0.65	903	4
Да	5000	15000	92.04	2.10	9.71	0.74	4922	10	8.25	0.68	4789	29
Да	10000	15000	171.72	4.96	52.15	0.80	9948	5	10.43	0.67	9780	23
Да	15000	15000	257.44	4.69	137.85	0.92	14853	5	10.55	0.76	14625	18
Да	20000	15000	336.02	5.98	217.63	0.96	19925	0	11.59	0.75	19524	18
Да	22000	15000	372.56	6.45	367.24	0.96	21925	0	12.40	0.74	21522	18
Нет	800	8980	13.80	Нет	0.17		794					
Нет	1000	10428	17.50	Нет	0.17	0.50	983	2	0.34	0.49	986	1
Нет	5000	15000	80.28	Нет	1.10	0.53	4661	46	33.18	0.42	4550	53
Нет	10000	15000	164.52	Нет	4.15	0.46	8874	157	77.20	0.33	9088	105
Нет	15000	15000	258.18	Нет	9.36	0.46	12805	272	107.26	0.41	13774	86
Нет	20000	15000	335.72	Нет	16.75	0.43	16911	376	95.33	0.46	18543	112
Нет	22000	15000	369.45	Нет	88.84	0.41	18498	421	101.54	0.43	20551	89

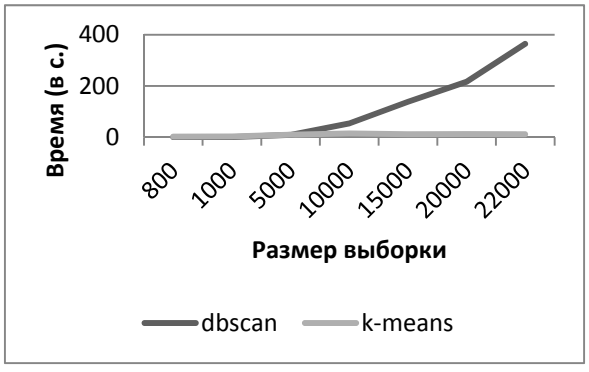

Рис. 2. Время выполнения кластеризации при применении TrSVD

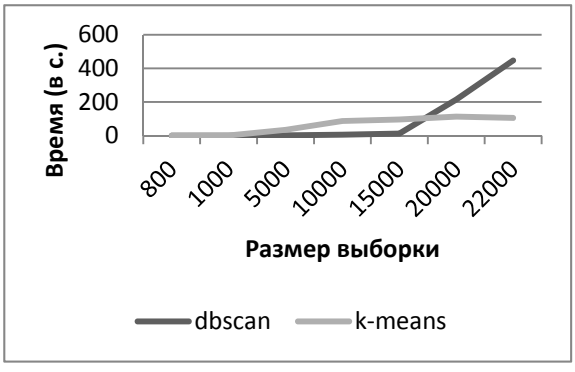

Рис. 3. Время выполнения кластеризации без применения TrSVD

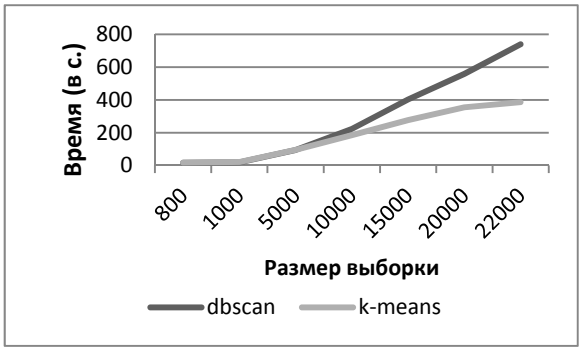

Рис. 4. Суммарное время
(tf-idf-векторизация, TrSVD, кластеризация)

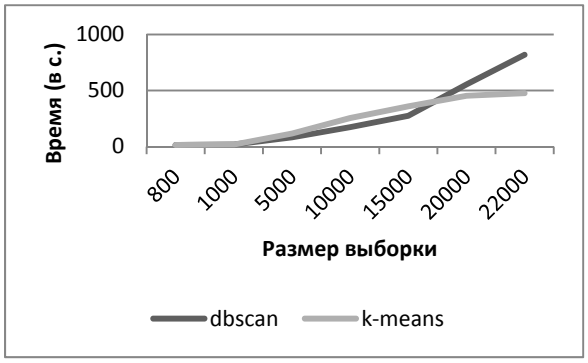

Рис. 5. Суммарное время
(tf-idf-векторизация, кластеризация)

В Табл. 2 коэффициент силуэта посчитан для отфильтрованных данных. По его значению можно говорить о хорошем разделении новостей по сюжетам при использовании обоих методов.

Применение фильтрации приводит к уменьшению выдаваемой в конце информации. А именно, теряется некоторое количество текстов и, соответственно, кластеров. Отметим (Рис. 6 - Рис. 9), что применение фильтрации значительно уменьшает число кластеров для K-Means, для Dbscan наблюдается меньшая разница.

Укажем также, что с применением TrSVD отбрасывается больше текстов чем без его применения для обоих методов. Без использования TrSVD алгоритм Dbscan теряет при фильтрации меньше элементов исходной выборки (Рис. 10, Рис. 11), и это число уменьшается с ростом её объёма (Рис. 11).

Пользуясь данными Табл. 1 и Табл. 2 заметим, что зачастую выполнение tf-idf-векторизации занимает больше всего времени, и, естественно, по (2), растет с увеличением объёма выборки.

В итоге, можно подчеркнуть, что результаты работы Dbscan на тестах приемлемы: он показал хорошее качество кластеризации и высокую скорость на больших размерах векторов текстов в некоторых границах объёма выборки.

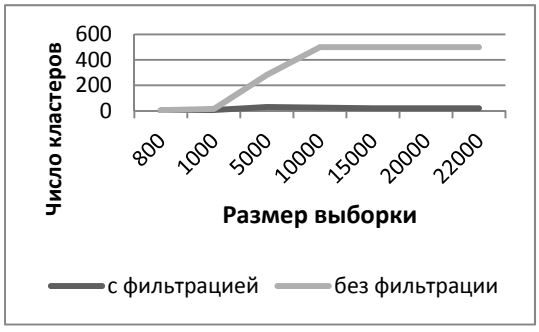

Рис. 6. Число кластеров K-Means с применением TrSVD

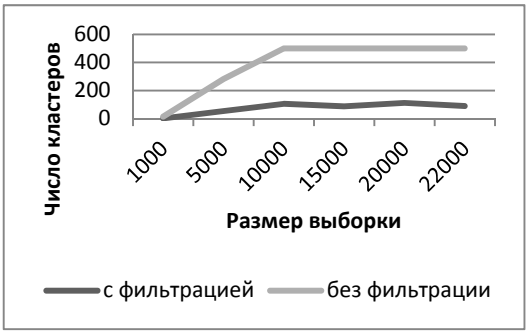

Рис. 7. Число кластеров K-Means без применения TrSVD

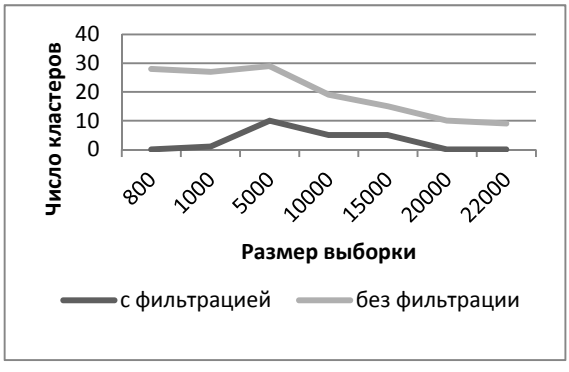

Рис. 8. Число кластеров Dbscan с применением TrSVD

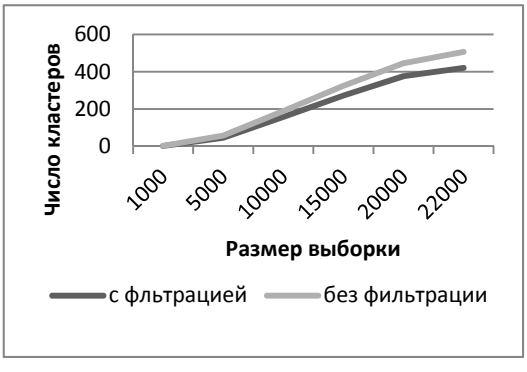

Рис. 9. Число кластеров Dbscan без применения TrSVD

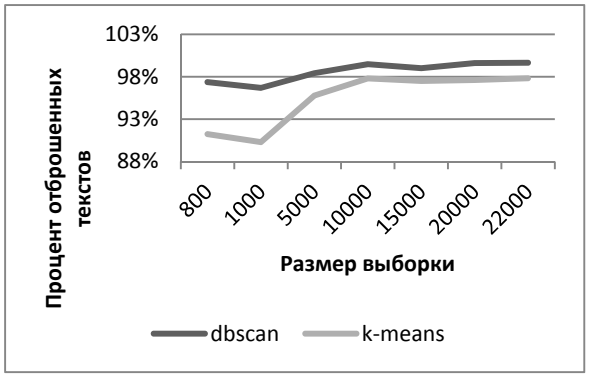

Рис. 10. Процент отброшенных тестов по отношению к начальному размеру выборки с применением TrSVD

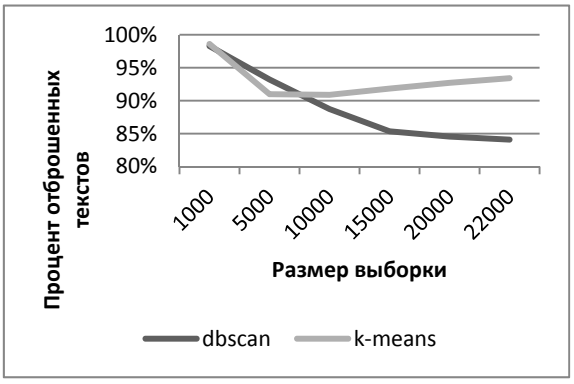

Рис. 11. Процент отброшенных текстов по отношению к начальному размеру выборки без применения TrSVD

3. Результаты для параметризации текстов по ключевым словам

На основе полученных результатов перейдем к реализации задачи кластеризации при определении текстовой выборки ключевыми словами. Ключевые слова текста новости определяются основе tf-idf-представления выборки: наиболее важные слова для текущего текста будут иметь наибольшее значение tf-idf-меры (2). Упорядочивая слова текста по убыванию значения этой меры, выбираем первые 15. Теперь же новость задается своими 15 ключевыми словами. Отметим, что все они различны. Далее, для кластеризации необходимо числовое представление текста новости. В данном случае, на основе всех ключевых слов новостей выборки создается общий словарь размером в 15 000 слов, и вектор новости — это вектор числа вхождений её ключевых слов в общий словарь. В результате получаем, что матрица выборки — это матрица со значениями 0 или 1. Затем, в целях ускорения работы Dbscan, мы предварительно рассчитаем матрицу расстояний между векторами новостей. Для упрощения вычислений и, следовательно, уменьшения времени выполнения, мы используем метрику Жаккара (3). Коэффициент силуэта считается для фильтрованных данных.

Такой алгоритм векторизации и расчета матрицы расстояний выбран из предположения его быстродействия, так как реализация Dbscan позволяет в качестве параметра передать ему заранее вычисленную матрицу расстояний.

При визуализации представляется фильтрованное содержимое найденных кластеров, а именно, ссылки на источник теста и 15 ключевых слов этого текста. При этом все кластера упорядочиваются по размеру. Также для каждого кластера определяются десять главных слов, характеризирующих его (Рис. 12).

порошенк,положен,воен,петр,украин,продл,продлен,оборон,снбо,президент	
	Порошенко сказал, почему не добивался продления военного положения
	порошенк, добива, положен, продлен, воен, украинск, петр, корабл, верховн, выбор, продл, попрос, буксир, никопол, бердянск
	Некоторые меры военного положения могут быть продлены
	порошенк, положен, воен, продлен, петр, продл, мер, критикова, поддержат, предложен, пограничник, прош, снбо, некотор, александровн
	Порошенко объяснил причины отмены военного положения на Украине
	порошенк, положен, объясн, воен, продлева, отм, украин, надобн, поч, суверенитет, защит, стран, целостн, демократ, ига.г
	Порошенко заявил, что готов продлить отдельные запреты военного положения
	порошенк, снбо, гот, положен, продлен, петр, воен, запрет, продл, мер, критикова, поддержат, заяв, предложен, аиф-украин
	Ляшко призвал Порошенко отчитаться об итогах военного положения на Украине
	ляшк, отчита, порошенк, положен, воен, украин, петр, призва, аиф-моск, радикальн, бесконечн, итог, завершa, войск, утвержда
	Пушков объяснил отмену военного положения на Украине
	пушк, отмен, положен, воен, порошенк, украин, объясн, откажет, даб, привыт, выбор, обострен, сенатор, наращива, сорва
	Преступность упала, гривна выросла что дало Украине военное положение
	порошенк, положен, воен, грив, макрофинансов, тимошенк, украин, инвест, огрос, зеленск, выбор, транш, азовск, украинск, рейтинг
собянин,москв,событ,мэр,главн,серг,назва,градоначальник,электробус,2018	
	Собянин назвал главные события 2018 года для москвичей и себя лично
	собянин, градоначальник, стройк, москвин, моск, мэр, событ, параллел, пиаесс, лиин, экзамен, завершат, облик, главн, назва
	Собянин отметил ключевые события года в жизни Москвы
	собянин, событ, моск, ключев, отрасл, серг, электробус, подиток, произошл, трамва, полюб, душевн, троллейбус, главн, перешл
	Собянин назвал главные события уходящего года в жизни Москвы
	собянин, реновац, метрополит, столичн, мэр, рекорд, моск, назва, событ, достижен, уходя, турникет, электробус, миф, главн
	Собянин озвучил главные достижения столицы в 2018 году
	собянин, достижен, озвуч, моск, тренировочн, мэр, столиц, назва, долголет, серг, заряд, душевн, посвита, рекордн, главн

Рис. 12. Визуализация содержимого кластеров

При тестировании алгоритма были получены следующие результаты (Табл. 3).

Из Табл. 3 следует, что самыми затратными по времени выполнения являются tf-idf-векторизация и вычисление матрицы расстояний между векторами текстов.

Отметим, что суммарное время выполнения всех операций для параметризации выборки по ключевым словам занимает значительно больше времени, чем для tf-idf-параметризации (Рис. 13). Также здесь наблюдается более низкое значение коэффициента силуэт (Рис. 14).

Однако для случая определения выборки по ключевым словам при фильтрации остаётся больше текстов, то есть в таком алгоритме кластера получаются более сконцентрированными около своих центров. Также при таком способе параметризации число кластеров получается больше (Рис. 15, Рис. 16).

Заключение

В данной работе были рассмотрены методы кластеризации, некоторым образом задействовавшие ключевые слова текста: при tf-idf-параметризации выборки по ним в конце объединялись похожие кластера, в последнем подходе именно ключевые слова задавали текст новости.

При сравнении методов Dbscan и K-Means для tf-idf-матрицы в качестве набора признаков оба показали сопоставимое качество кластеризации по показателю точности, независимо применяется TrSVD или нет. Однако Dbscan показал высокие показатели скорости выполнения на больших размерах векторов текстов в некоторых границах объёма выборки (Табл. 2).

Отметим, что число новостей и число кластеров, оставляемых для конечного отображения, меньшее при применении TrSVD, чем без его применения для обоих методов.

Табл. 3. Результаты работы алгоритма при параметризации элементов выборки их ключевыми словами

Размер выборки	Размер вектора признаков	Время tf-idf-векторизации (в с.)	Время расчета матрицы частотности вхождения (в с.)	Время расчета матрицы расстояний (в с.)	Время кластеризации (в с.)	Число кластеров	Число отброшенных текстов	Силуэт	Число кластеров после фильтрации
800	8909	13.36	0.02	1.18	0.01	31	732	0.25	31
1000	10122	16.74	0.03	1.98	0.01	40	918	0.25	40
5000	15000	85.55	0.10	94.57	0.18	443	4047	0.31	404
10000	15000	169.80	0.18	397.37	0.70	996	7725	0.19	910
15000	15000	254.93	0.26	913.13	9.29	1529	11425	0.17	1390
20000	15000	339.62	0.34	1664.31	13.21	1963	15238	0.13	1820
22000	15000	369.47	0.38	2053.75	51.29	2226	16637	0.13	2034

Рис. 13. Суммарное время выполнения всех операций для параметризации выборки по ключевым словам и для tf-idf-параметризации при применении Dbscan без использования TrSVD

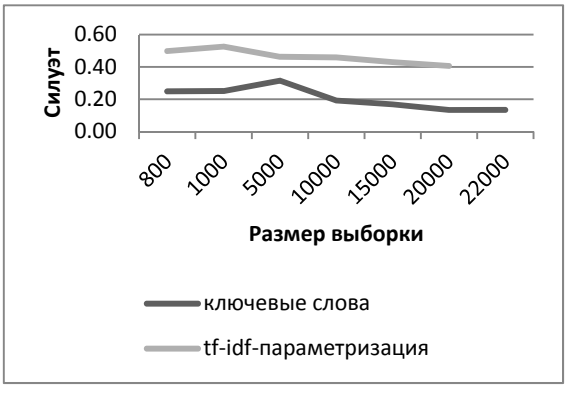

Рис. 14. Коэффициент силуэт для фильтрованных данных при параметризации выборки по ключевым словам и при tf-idf-параметризации с использованием Dbscan без TrSVD

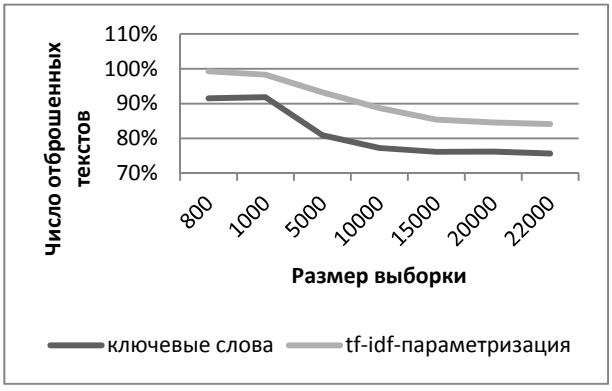

Рис. 15. Процент отброшенных текстов при фильтрации от начального размера выборки для параметризации выборки по ключевым словам и для tf-idf-параметризации при применении Dbscan без использования TrSVD

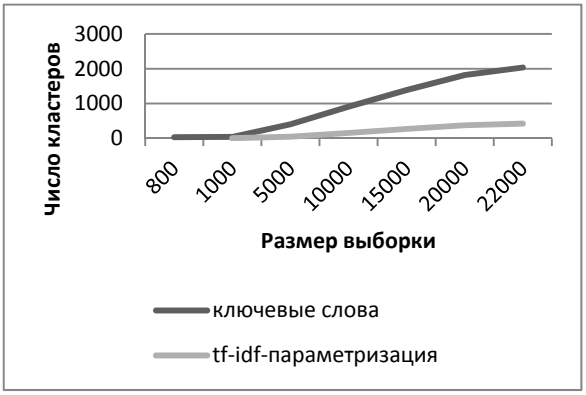

Рис. 16. Число кластеров при фильтрации для параметризации выборки по ключевым словам и для tf-idf-параметризации при применении Dbscan без использования TrSVD

Для финального алгоритма формальные показатели качества кластеризации и скорость выполнения оказались ниже, чем при tf-idf-параметризации. Особенно затратными по времени оказались tf-idf-векторизация и вычисление матрицы расстояний. Однако здесь формальные показатели качества кластеризации сохраняются довольно высокими, при просмотре содержимого кластеров ошибок метода не наблюдается. Также для этого алгоритма имеет место наиболее полное сохранение материала выборки для конечного отображения сюжетов (Табл. 2).

Таким образом, полученные в ходе исследования данные позволяют сделать вывод, что для объёма выборки менее чем в 1000 текстов эффективным является алгоритм, сочетающий K-Means и TrSVD, так как тут наблюдается высокие значения показателя скорости и коэффициента силуэта. Если объём выборки больше 1000 текстов, но меньше 20000, то более подойдет методов Dbscan, так как тут наблюдается самый высокий показатель скорости, а также высокий показатель полноты и более чем приемлемый показатель качества (Табл. 3).

Литература

1. Aggarwal C. C. A framework for diagnosing changes in evolving data streams. // In Proceedings of the ACM SIGMOD International Conference on Management of Data. —2003. — P. 575–586.
2. Guha S. Mishra N.-Motwani R., O'Callaghan L. Clustering data streams. // In Proceedings of the IEEE Symposium on Foundations of Computer Science. — 2000. —P. 359–366.
3. O'Callaghan L. Mishra N.-Meyerson A. Guha S., Motwani R. Streaming data algorithms for high-quality clustering. // In Proceedings of the 18th International Conference on Data Engineering. — 2002. — P. 685–694.
4. Jones K. S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. MCB University: MCB University Press. — 2004. — Vol. 60, no. 5. — P. 493-502.
5. [Электронный ресурс]— https://www.python.org/
6. Bird S. NLTK: the natural language toolkit // In Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics. — 2006.— P. 69-72.
7. Ester M., Kriegel H.P., Sander J., XiaoweiXu A. Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining. AAAI Press. — 1996.— P. 264-323.
8. [Электронный ресурс]— https://scikit-learn.org/
9. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes in C. — Cambridge: Cambridge University Press. 1997.— 1018p.
10. K-means vs Mini Batch K-means: a comparison. / Bejar, J. http://hdl.handle.net/2117/23414 (дата обращения - 10.05.2019).
11. Martin Ester, Hans-Peter Kriegel, J&g Sander, Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise —KDD— 1996. – P. 226-231.
12. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. // В.А. Галактионов, Е.Б. Козеренко — Москва: МИЭМ.2011. — 272 с.
13. Jaccard P. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines // Bull. Soc. Vaudoise sci. Natur. V. 37. Bd. 140. — 1901. — S. 241-272.
14. Peter J. Rousseeuw Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis // Computational and Applied Mathematics. — 1987. — Vol. 20.— P.53–65. doi:10.1016/0377-0427(87)90125-7.

Головастова Элеонора Александровна. Московский государственный университет им. М.В. Ломоносова, г. Москва. Аспирант. Окончила Московский государственный университет им. М.В. Ломоносова в 2017 году. Область научных интересов: теория массового обслуживания, информационные технологии. E-mail: golovastova.elina@yandex.ru

Красотин Дмитрий Николаевич. ЗАО "Московский научно-исследовательский телевизионный институт", г. Москва. Ведущий инженер. Окончил Московский государственный университет им. М.В. Ломоносова в 2011 году. Область научных интересов: информационные технологии, физика. E-mail: dima88_kr@mail.ru

Effective Clustering of a Text Sample Depending on the Different Parameterization of this Sample

E. A. Golovastova [I], D. N. Krasotin [II]

[I] Lomonosov Moscow State University, Moscow, Russia
[II] CJSC "MNITI", Moscow, Russia

Abstract. The Internet becomes the primary means of receiving text news. As a result, there is a necessity in automated processing of large data amount. One of the most important tasks is the automated cultivation of text information. In this paper we will consider the problem of effective clustering for objects from text sample. The most common representation of the text set is the matrix, which elements are the statistical measure values calculated on the basis of the word frequency. In opposition to we suggest parametrization by the text key words. We use two methods to provide the clustering: K-means and Dbscan. This paper considers the analysis of mentioned methods and provide comparison of the clustering quality results, which depend on various text parameterization and the used algorithm.

Keywords: Clustering, text set, sample parameterization, tf-idf-measure, keywords, effective method.

DOI 10.14357/20718632190406

References

1. Aggarwal C. C. 2003. A framework for diagnosing changes in evolving data streams. In Proceedings of the ACM SIGMOD International Conference on Management of Data: 575–586.
2. Guha S. Mishra N.-Motwani R., O'Callaghan L. 2000. Clustering data streams. In Proceedings of the IEEE Symposium on Foundations of Computer Science: 359–366.
3. O'Callaghan L. Mishra N.-Meyerson A. Guha S., Motwani R. 2002. Streaming data algorithms for high-quality clustering. In Proceedings of the 18th International Conference on Data Engineering: 685–694.
4. Jones K. S. 2004. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. MCB University: MCB University Press. Vol. 60, no. 5: 493-502.
5. [Internet resource] Available at: https://www.python.org/
6. Bird S. 2006. NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics: 69-72.
7. Ester M., Kriegel H.P., Sander J., XiaoweiXu A. 1996. Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining. AAAI Press: 264-323.
8. [Internet resource] Available at: https://scikit-learn.org/
9. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. 1997. Numerical Recipes in C. Cambridge: Cambridge University Press. 1018p.
10. Bejar, J. K-means vs Mini Batch K-means: a comparison. Available at: http://hdl.handle.net/2117/23414 (accessed - 10.05.2019).
11. Martin Ester, Hans-Peter Kriegel, J&g Sander, Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD: 226-231.
12. Bolshakova E.I., Klyshinsky E.S., Lande D.V., Noskov A.A., Peskova O.V., Yagunova E.V. 2011. Avtomaticheskaia obrabotka tekstov na estestvennom iazyke i komp'iuternaia lingvistika [Automatic processing of natural language texts and computer linguistics]. Moscow: MIEM 272 p.
13. Jaccard P. 1901. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines. Bull. Soc. Vaudoise sci. Natur. V. 37. Bd. 140: S. 241-272.
14. Peter J. Rousseeuw. 1987. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics. Vol. 20.:53–65. doi:10.1016/0377-0427(87)90125-7.

Golovastova E. A. PhD student, Lomonosov Moscow State University, Leninskie Gory, Moscow, 119991, Russian Federation. E-mail: golovastova.elina@yandex.ru

Krasotin D. N. Lead Engineer, CJSC "MNITI", 7 Golyanovskaya ulitsa, litera A, str.1, Moscow, 105094, Russian Federation. E-mail: dima88_kr@mail.ru