УДК 004.912

Алгоритм кластеризации новостного потока текстовых сообщений

The algorithm of clustering of the news stream of text messages

1. **Гапанюк Ю.Е. (Gapanyuk Yu.E.), доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, gapyu@bmstu.ru**
2. **Чернобровкин С.В. (Chernobrovkin S.V.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, sergey.chernobrovkin@inbox.ru**
3. **Латкин И.И. (Latkin I.I.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, igor.latkin@outlook.com**
4. **Леонтьев А.В. (Leontiev A.V.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, alekseyl@list.ru**
5. **Ожегов Григорий Андреевич (Ozhegov G.A.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, grigory@ozhegov.name**
6. **Опрышко Александр Владимирович (Opryshko A.V.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, alexopryshko@yandex.ru**
7. **Мялкин Максим Павлович (Myalkin M.P.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, maxmyalkin@gmail.com**

1. Введение

С развитием новостных агентств и средств массовой информации в интернете все чаще перед редакторами изданий встает задача нахождения информации для написания статей. Помимо агентств масштаба государства есть множество региональных СМИ. В последнее время информационные поводы для новостей могут появляться также на страницах социальных сетей политиков, ученых, деятелей культуры.

Первичной задачей обработки новостного потока текстовых сообщений является кластеризация сообщений, то есть выделение групп, элементы которых будут схожи друг с другом на основе какой-либо метрики.

Особенностью кластеризация новостного потока является то, что это задача онлайн-кластеризации, так как новости появляются в потоке постоянно, и их тематика заранее неизвестна.

В соответствии с [1] традиционно выделяют два основных подхода к кластеризации: иерархическая и итеративная кластеризация.

При иерархической кластеризации исходное множество элементов выстраивается в древовидную структуру. При использовании иерархических дивизимных (нисходящих) методов исходное множество элементов считается начальным кластером, который в процессе работы алгоритма делится на меньшие кластеры. При использовании иерархических агломеративных (восходящих) методов каждый элемент исходного множества считается отдельным кластером, в процессе работы алгоритма элементы объединяются в большие кластеры. Условие останова деления или объединения зависит от используемой метрики.

Очевидно, что алгоритмы иерархической кластеризации изначально не подходят для онлайн-кластеризации, так как алгоритмы восходящего или нисходящего построения дерева кластеров предполагают, что исходное множество элементов известно заранее и не может изменяться в процессе работы алгоритма.

Алгоритмы итеративной кластеризации осуществляют итеративное деление исходного множество элементов на кластеры. При этом новые кластеры формируются до тех пор, пока не будет выполнено правило остановки. Теоретически, алгоритмы итеративной кластеризации в большей степени подходят для решения поставленной задачи, так как исходное множество элементов может пополняться в процессе работы алгоритма. Но на практике ограничения существующих алгоритмов могут мешать их использованию для кластеризации новостей. Одним из наиболее известных алгоритмов итеративной кластеризации является алгоритм к-средних (k-means). Число кластеров k является

гиперпараметром алгоритма (задается заранее перед началом работы алгоритма), что делает непригодным оригинальный алгоритм для кластеризации новостей. Другой проблемой алгоритма k-means является его склонность к переобучению в зависимости от начальных условий, начальные данные могут очень сильно влиять на конфигурацию полученных кластеров, что становится особенно заметно при кластеризации больших потоков данных.

В настоящее время алгоритм k-means пытаются адаптировать для онлайн-кластеризации [2, 3]. Но предложенные модификации в основном направлены на борьбу с переобучением, число кластеров по-прежнему остается гиперпараметром, что делает непригодным модифицированные алгоритмы для кластеризации новостного потока.

В данной статье предлагается простой алгоритм кластеризации новостного потока текстовых сообщений, который можно рассматривать как модификацию алгоритма k-means, и исследуются результаты его работы.

2. Структура предлагаемой системы на основе метаграфового подхода

С точки зрения подхода гибридных интеллектуальных информационных систем (ГИИС), предложенного в [4], задача кластеризации относится к модулю подсознания (МП) ГИИС. Структура предлагаемой системы, представленная на рис. 1, является частным случаем структуры системы сбора и анализа данных Интернет-источников на основе метаграфового подхода, рассмотренной в [5].

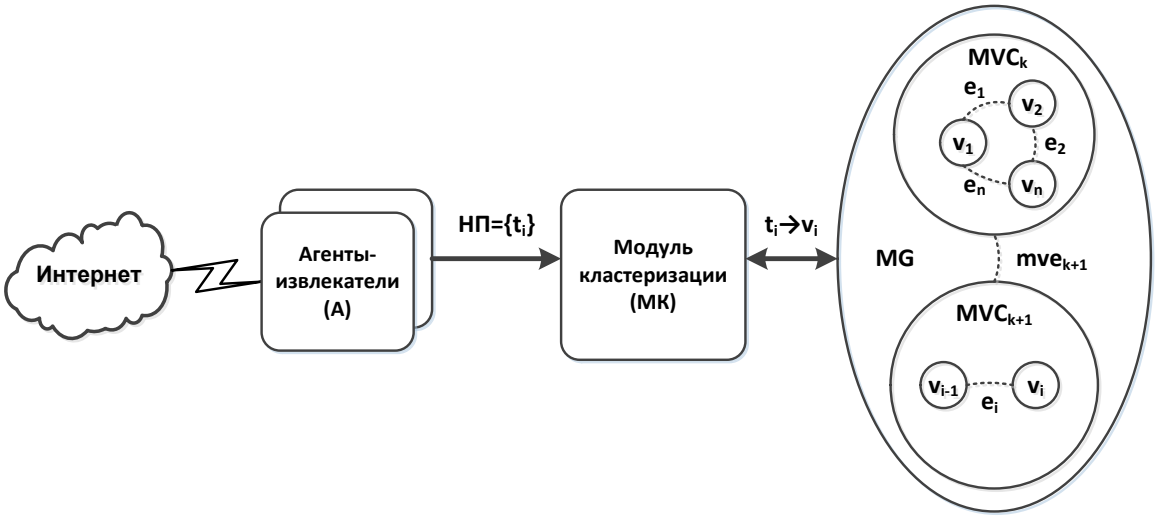

Рис. 1. Структура системы кластеризации новостного потока текстовых сообщений на основе метаграфового подхода

Сбор информации осуществляется с помощью агентов-извлекателей (А), которые осуществляют извлечение информации с сайтов, из новостных лент, социальных сетей и других источников. Агенты-извлекатели формируют новостной поток (НП) текстовых сообщений t_i, $НП = \{t_i\}$, который поступает на вход модуля кластеризации (МК).

Модуль кластеризации, в соответствии с [6], может быть реализован в виде метаграфового агента, который формирует выходной метаграф *MG*. Текстовые сообщения t_i преобразуются в вершины метаграфа v_i, которые могут быть аннотированы дополнительными атрибутами (текст исходного сообщения, вспомогательные метрики кластеризации). Выходной метаграф *MG* содержит произвольное количество метавершин-кластеров $MVC = \{MVC_k\}$. Каждая метавершина-кластер содержит ранее кластеризованные вершины-сообщения v_1, v_2,… v_n, относящиеся к данному кластеру. В зависимости от применяемого алгоритма кластеризации, вершины-сообщения могут быть соединены вспомогательными ребрами e_1, e_2,… e_n, которые используются в алгоритме. При необходимости метаграфовый агент может добавить новую метавершину-кластер MVC_{k+1} и поместить в нее новую вершину-сообщение v_i. При этом для новой вершины v_i могут быть добавлены вспомогательные ребра, связывающие ее с другими вершинами кластера (на рисунке показано ребро e_i). Для новой метавершины-кластера MVC_{k+1} также могут быть добавлены вспомогательные ребра, связывающие ее с другими кластерами (на рисунке показано ребро mve_{k+1}).

Система правил метаграфового агента модуля кластеризации содержит два правила. При приходе нового текстового сообщения t_i оно преобразуется в вершину v_i. Далее в

зависимости от используемого алгоритма кластеризации и текущего состояния метаграфа MG выполняется одно из следующих правил:

1. $(MG, v_i) \to A\lg(MG, v_i) = MVC_k; MVC_k + v_i; MVC_k + \{e_i\}$ – с использованием алгоритма кластеризации *Alg* для новой вершины-сообщения v_i из существующего набора метавершин-кластеров выбирается наиболее подходящая метавершина-кластер MVC_k. В найденный кластер добавляется новая вершина-сообщение v_i и множество вспомогательных внутрикластерных ребер $\{e_i\}$.
2. $(MG, v_i) \to A\lg(MG, v_i) = \varnothing; MG + MVC_{k+1}; MG + \{mve_{k+1}\}; MVC_{k+1} + v_i; MVC_{k+1} + \{e_i\}$ – если подходящий кластер не найден, то в метаграф добавляется новая метавершина-кластер MVC_{k+1} и множество вспомогательных межкластерных ребер $\{mve_{k+1}\}$. В новый кластер добавляется новая вершина-сообщение v_i и множество вспомогательных внутрикластерных ребер $\{e_i\}$.

Полученная на этапе кластеризации метаграфовая структура является основой для следующих этапов обработки новостных текстовых сообщений и может быть преобразована другими метаграфовыми агентами для решения дальнейших задач.

3. Описание алгоритма кластеризации

Предложенная структура позволяет использовать различные алгоритмы кластеризации *Alg* для решения поставленной задачи. В данной статье используется алгоритм на основе порогового расстояния, который состоит из двух шагов.

Шаг 1. $Dist_{\min} = \min\left(Dist\left(MVC_k, v_i\right)\right), \forall MVC_k \in MVC$

Для каждого кластера MVC_k, входящего во множество кластеров *MVC* на основе метрики *Dist* вычисляется расстояние между кластером и новой вершиной-сообщением v_i. Из вычисленных расстояний выбирается минимальное расстояние $Dist_{\min}$.

Шаг 2. $MVC_{RES} = \begin{cases} Dist_{\min} < Dist_{\text{threshold}} \rightarrow MVC_k \\ Dist_{\min} \geq Dist_{\text{threshold}} \rightarrow \varnothing \end{cases}$

Выходной кластер MVC_{RES} определяется на основе порогового расстояния $Dist_{\text{threshold}}$. Если найденное минимальное расстояние меньше порогового, то в качестве найденного кластера выбирается кластер MVC_k, соответствующий найденному минимальному расстоянию. Если найденное минимальное расстояние больше порогового, то возвращается признак того что кластер не найден, что приводит к созданию нового кластера.

Пороговое расстояние $Dist_{\text{threshold}}$ и метрика *Dist* являются гиперпараметрами алгоритма.

В качестве *Dist* использовался ряд метрик, результаты экспериментов с которыми приведены в следующих разделах. Но перед рассмотрением результатов экспериментов необходимо кратко рассмотреть используемые метрики оценки качества кластеризации.

4. Используемые метрики оценки качества кластеризации

Для оценки качества будем использовать метрики F-score, Purity и NMI.

4.1. Оценка качества кластеризации с использованием метрики F-score

В соответствии с [7] F-score или f-score (F-метрика) определяется следующим образом:

$$F = 2 * \frac{precision * recall}{precision + recall}, precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN},$$

где *precision* – точность алгоритма; *recall* – полнота алгоритма; *TP* (True Positives) – количество корректно распознанных алгоритмом положительных примеров; *FP* (False Positives) – количество примеров, некорректно распознанных алгоритмом как положительные; *FN* (False Negatives) – количество примеров, некорректно распознанных алгоритмом как отрицательные.

Рассмотренную формулу часто называют F_1-мерой, она является частным случаем F_β-меры: $F_\beta = \left(1+\beta^2\right) * \frac{precision * recall}{\left(\beta^2 * precision\right) + recall}$. Значение параметра β обычно указывается через дефис, например f-0,5 или f-2.

В результате работы алгоритма кластеризации получается множество кластеров с отнесенными к ним статьями. Так как количество получаемых кластеров заранее неизвестно, невозможно определить соответствие полученного кластера и кластера из исходного множества, чтобы рассчитать необходимые для F-score данные.

Поэтому было принято решение разбить все исходные данные на множество всевозможных пар текстов, относящихся к одному кластеру. Таким же образом разбивается и полученное множество результатов. В итоге получается два множества пар, где каждая пара представляется двумя текстами, относящимися к одному и тому же кластеру. Обозначим пары, полученные из исходного множества за G, а пары из результирующего множества за C.

Тогда требуемые параметры можно вычислить следующим образом: $TP = \left|G \cap C\right|$, $FP = \left|C \setminus G\right|$, $FN = \left|G \setminus C\right|$. Очевидно, что правильно определенными примерами необходимо считать те пары, которые в исходном множестве были в одном кластере и в полученном множестве также оказались в одном кластере, то есть являются пересечением исходного и результирующего множеств. Некорректно определенными можно считать те пары текстов, которые в результате работы алгоритма попали в один кластер, хотя в исходном множестве были в разных кластерах. Таким образом, FP можно определить как разность результирующего и исходного множеств. Некорректно определенные отрицательные примеры FN – это те тексты, которые в исходном множестве принадлежат одному и тому же кластеру, однако в результате работы алгоритма оказались в разных кластерах. Таким образом, FN можно определить как разность исходного и результирующего множеств.

4.2. Оценка качества кластеризации с использованием метрики Purity

В соответствии с [7] Purity или чистота кластера – простая метрика качества кластеризации. Для ее подсчета каждый результирующий кластер помечается меткой того кластера, который встречается чаще остальных для документов этого кластера. Затем считается точность присвоения документов кластерам и делится на N – общее количество документов. Метрика purity определяется следующим образом: $purity(\omega, c) = \frac{1}{N}\sum_k \max_j \left|\omega_k \cap c_j\right|$, где ω – множество результирующих кластеров; c – множество исходных кластеров.

Метрика purity тем выше, чем больше в одном кластере оказалось документов, принадлежащих к одному кластеру в исходном множестве.

4.3. Оценка качества кластеризации с использованием метрики NMI

В соответствии с [7] NMI – normalized mutual information или нормализованная взаимная информация – мера взаимной зависимости двух случайных величин. Данная мера показывает количество информации полученной об одной случайной величине через другую случайную величину.

Высокое значение purity не всегда гарантирует высокое качество работы алгоритма кластеризации. Дело в том, что если количество кластеров K равно общему количеству документов N, то значение purity будет равно единице. NMI устраняет эту проблему, благодаря нормализации. Метрика NMI определяется следующим образом: $NMI(\omega, c) = \frac{I(\omega, c)}{\left(H(\omega) + H(c)\right)/2}$, где $I(\omega, c)$ – взаимная информация исходных и результирующих кластеров; $H(\omega)$ – энтропия результирующих кластеров; $H(c)$ – энтропия исходных кластеров.

Числитель NMI – взаимная информация, обладает тем же недостатком, что и метрика purity – растет с увеличением количества кластеров. Чтобы обойти этот недостаток используется нормализация. Энтропия растет с увеличением числа кластеров и достигает

своего максимума при K = N. Это позволяет использовать NMI для сравнения алгоритмов кластеризации при разном количестве кластеров. Вид знаменателя NMI подобран таким образом, чтобы значения NMI принадлежали интервалу от 0 до 1.

5. Результаты экспериментов с различными метриками кластеризации

В данном разделе рассмотрим результаты экспериментов с рядом метрик, которые использовались в качестве расстояния *Dist* в предложенном ранее алгоритме.

В качестве технологического стека для проведения экспериментов был использован язык Python с пакетами для анализа текстов PyMorphy (морфология русского языка) и NLTK. Для задач обработки данных и оценки качества кластеризации использовались библиотеки scikit-learn и numpy.

В качестве эталонных данных для проведения экспериментов были взяты новостные статьи с сервиса Яндекс.Новости. Кластером считались все новости, объединенные Яндексом в один инфоповод.

5.1. Коэффициент Жаккара

Коэффициент Жаккара в соответствии с [7] определяется следующим образом: $D = \frac{|A \cap B|}{|A \cup B|}$, где D – расстояние между множествами A и B.

В случае применения коэффициента к текстам в качестве множеств A и B могут выступать множество слов первого и второго текстов. Данная метрика является понятной и легко реализуема на практике. Основным недостатком этой метрики является то, что она никак не учитывает частоту появления слов в текстах. Результаты экспериментов представлены на рис. 2. На всех графиках, содержащих результаты экспериментов, на оси X показано пороговое расстояние $Dist_{\text{threshold}}$, а на оси Y значение меры качества кластеризации (F-score, Purity, NMI) в диапазоне [0;1].

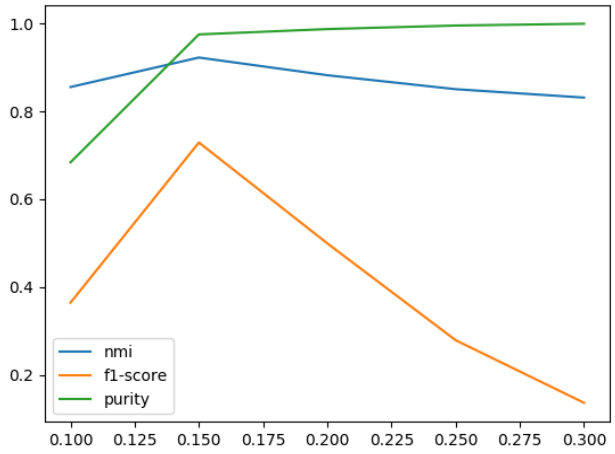

Рис. 2а

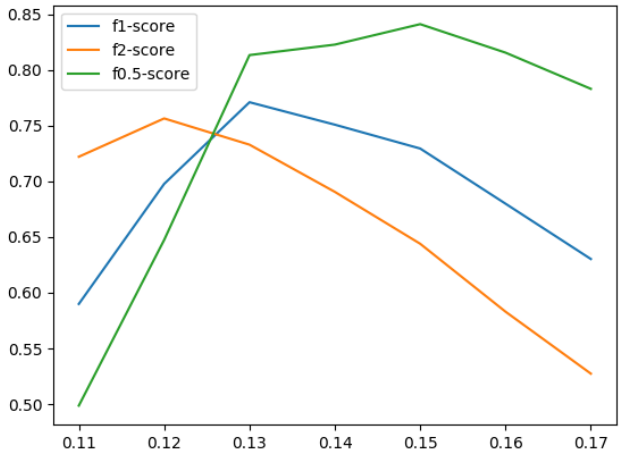

Рис. 2б

Рис. 2. Оценка качества алгоритма с использованием коэффициента Жаккара, 2а – на всем исходном множестве; 2б – в окрестностях значения 0,15

В соответствии с рис. 2а наилучшие результаты алгоритм показывает в районе порога при значении 0,15. Метрики качества NMI и Purity оказываются не слишком репрезентативны для оценки такой кластеризации. Это может быть обусловлено, в том числе, тем, что образуется множество единичных кластеров (в них находится только одна статья). F-мера оказывается более репрезентативной метрикой, поэтому проведем еще один эксперимент, но с двумя модификациями f-меры: f-0.5 и f-2. В соответствии с рис. 2б будем рассматривать окрестности порога 0,15.

Лучшие значения оказались в точке 0,13. Следовательно, этот порог является наилучшим для наших экспериментальных данных при использовании коэффициента Жаккара. Значение показателя f1 составляет примерно 0,77.

5.2. Мера TF-IDF

Мера TF-IDF в соответствии с [7] является широко распространенной метрикой схожести текстов и расшифровывается как Term Frequency – Inversed Document Frequency. Такая мера построена на частоте появления терма (слова или литерала) в тексте и во всем множестве текстов. Формула состоит из трех частей:

1. $TF(t,d) = \frac{n_t}{\sum_k n_k}$, где t – терм; d – документ; n_t – количество вхождений терма t в документ; $\sum_k n_k$ – количество термов в документе.

2. $IDF(t,D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|}$, где $|D|$ – общее количество документов в корпусе D; $|\{d_i \in D \mid t \in d_i\}|$ – количество документов в корпусе D, в которых встречается терм t.

3. $TF - IDF(t,d,D) = TF(t,d) * IDF(t,D)$. Итоговая мера *TF-IDF* является произведением *TF* и *IDF*. Каждое слово в тексте характеризуется значением *TF-IDF*, а сам текст – вектором из таких чисел.

Для сравнения двух векторов значений *TF-IDF*, как правило, используют косинусную меру: $\cos(\alpha) = \frac{A * B}{|A| * |B|}$, где A и B – вектора значений *TF-IDF*.

Плюсом рассмотренного подхода является учет частоты появления термов при формировании вектора текста, однако для подсчета *TF-IDF* необходимо хранить $\{d_i \in D \mid t \in d_i\}$ для всех термов, что требует дополнительных ресурсов памяти. К тому же необходимо постоянно обновлять эти данные.

В начале работы алгоритма *TF-IDF*-вектор одного и того же текста может отличаться от *TF-IDF*-вектора этого же текста через некоторое время работы, так как при добавлении новых слов в словарь веса термов могут изменяться.

Результаты экспериментов с мерой TF-IDF представлены на рис. 3.

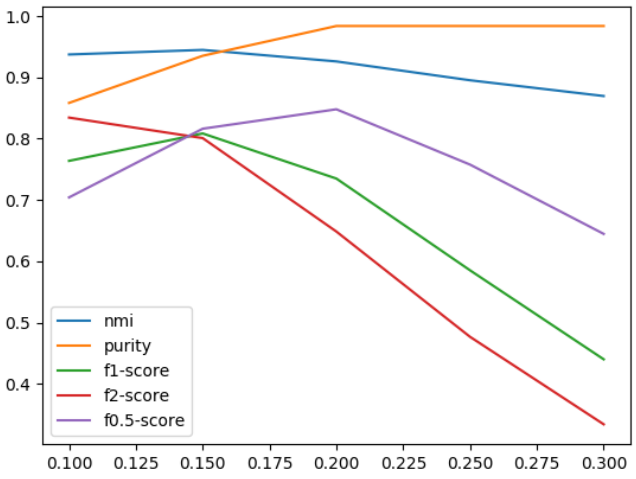

Рис. 3а

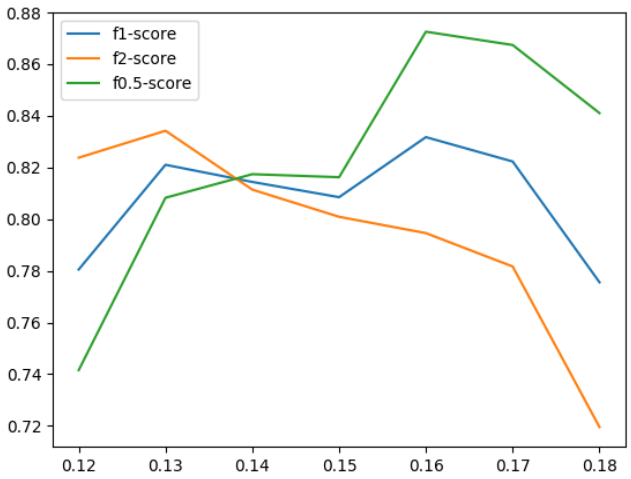

Рис. 3б

Рис. 3. Оценка качества алгоритма с использованием меры TF-IDF, 3а – на всем исходном множестве; 3б – в окрестностях значения 0,15

Уберем из рассмотрения нерепрезентативные метрики NMI и Purity. В соответствии с рис. 3а наилучший результат f-мера показывает, так же, как и в предыдущем случае в окрестностях точки 0,15. Значение f1-меры равно 0,81.

В соответствии с рис. 3а наилучшие результаты алгоритм показывает при пороге 0,16. f1-score = 0,84.

Таким образом, результаты оценки качества кластеризации показывают превосходство использования меры TF-IDF над использованием коэффициента Жаккара.

6. Влияние предобработки текстовых сообщений на качество работы алгоритма

В рассмотренных выше алгоритмах в качестве термов используются обыкновенные слова с учетом падежа/склонения/числа и т.д., что может оказывать влияние на результат. Рассмотрим влияние предобработки текстовых сообщений на результат работы алгоритма. Были проведены эксперименты с двумя вариантами предобработки: приведение слов к нормальной форме и использование биграмм из слов. Поскольку предыдущие эксперименты показали превосходство меры TF-IDF над коэффициентом Жаккара, то в дальнейших экспериментах будет использоваться TF-IDF. Результаты экспериментов представлены на рис. 4.

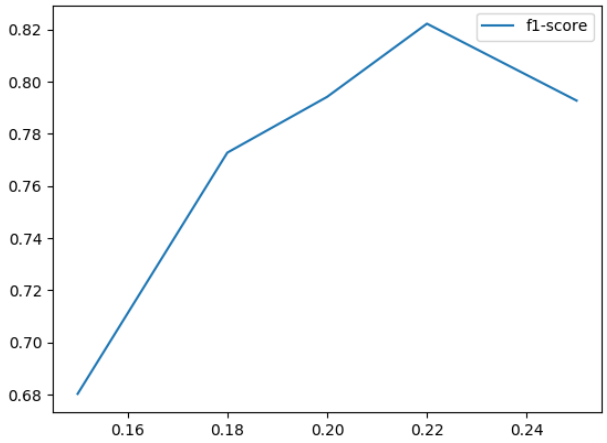

Рис. 4а

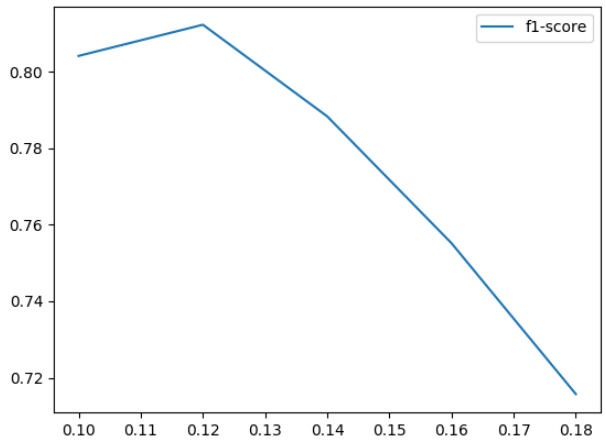

Рис. 4б

Рис. 4. Оценка качества алгоритма с использованием предобработки, 4а – приведение слов к нормальной форме; 4б – использование биграмм из слов

Приведение слов к нормальной форме предполагает приведение к мужскому роду, именительному падежу. Например, слово «победителя» будет приведено к «победитель». Также в процессе предобработки исключаются стоп-слова, ухудшающие качество работы алгоритма. Поскольку в результате предобработки слова, которые раньше считались различными, будут теперь считаться одинаковыми, то их вес будет отличаться от того веса, который был до предобработки.

При использовании биграмм из слов в качестве термов будем использовать не только слова, но и пары стоящих рядом слов. Текст будет представлен вектором, полученным из множества слов и всех подряд идущих пар слов, например, текст «мама мыла раму» будет представлен множеством { мама, мыть, рама, мама_мыть, мыть_рама }.

Как видно из графиков на рис. 4а и 4б качество работы алгоритма не изменилось, значение f1-меры по-прежнему составляет около 82%. Поэтому можно сделать вывод о том, что предобработка текстовых сообщений в виде приведения слов к нормальной форме и использования биграмм из слов не оказывает существенного влияния на качество кластеризации.

Необходимо отметить, что рассмотренные варианты предобработки часто применяются в машинном обучении, и могут при определенных условиях оказывать существенное

влияние на качество работы алгоритмов обработки текстов. Однако проведенные эксперименты показывают, что в данном случае их применение неэффективно.

Выводы

Задача кластеризации новостного потока текстовых сообщений является важным этапом обработки новостного потока текстовых сообщений.

Существующие алгоритмы иерархической и итеративной кластеризации не могут быть в неизменном виде применены для кластеризации динамического новостного потока текстовых сообщений.

С точки зрения метаграфового подхода задача кластеризации может быть представлена как задача динамического формирования метаграфа с добавлением вершин-сообщений, метавершин-кластеров и необходимых связей между ними.

Предложенный алгоритм кластеризации предполагает добавление новых кластеров на основе использования порогового расстояния и метрики для оценки принадлежности вершины к кластеру.

В качестве метрик используются коэффициент Жаккара и мера TF-IDF. Проведенные эксперименты показали, что мера TF-IDF обеспечивает лучшее качество работы алгоритма. При этом использование предобработки текстовых сообщений в виде приведения слов к нормальной форме и использование биграмм из слов не оказывает существенного влияния на качество кластеризации.

Литература

1. Чубукова И.А. Data Mining. М.: Национальный Открытый Университет «ИНТУИТ», 2016 – 471 с.

2. W. Barbakh and C. Fyfe. Online clustering algorithms. International Journal of Neural Systems, 18(3), 2008, pp. 185-194.

3. E. Liberty, R. Sriharshay and M. Sviridenko. An Algorithm for Online K-Means Clustering. Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENE), 2016, pp. 81-89. DOI: 10.1137/1.9781611974317.7

4. Черненький В.М., Терехов В.И., Гапанюк Ю.Е. Структура гибридной интеллектуальной информационной системы на основе метаграфов. Нейрокомпьютеры: разработка, применение. 2016. Выпуск №9. С. 3-14.

5. Ревунков Г.И., Гапанюк Ю.Е., Нардид А.Н. Структура системы сбора и анализа данных Интернет-источников на основе метаграфового подхода. Естественные и технические науки. 2016. Выпуск № 12. С. 275-277.

6. Самохвалов Э.Н., Ревунков Г.И., Гапанюк Ю.Е. Использование метаграфов для описания семантики и прагматики информационных систем. Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2015. Выпуск №1. С. 83-99.

7. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011 – 528 с.