Наука и Образование
МГТУ им. Н.Э. Баумана
Сетевое научное издание
ISSN 1994-0408

Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2016. № 10. С. 117–131.

DOI: **10.7463/1016.0848929**

Представлена в редакцию: 04.09.2016
Исправлена: 18.09.2016

УДК 004.021

Выявление и выделение шаблонов в массиве коротких сообщений

Вирцева Н. С.[1], **Вишняков И. Э.**[1,*]

*vishnyakov@bmstu.ru

[1]МГТУ им. Н.Э. Баумана, Москва, Россия

Статья посвящена проблеме выявления и анализа автоматически генерируемых сообщений в произвольном множестве коротких текстовых сообщений. Рассмотрены такие этапы решения задачи, как предобработка сообщений, группировка похожих сообщений и определение структуры шаблона, а также отнесение произвольных сообщений к одному из шаблонов. Предложен метод выделения шаблонов, основанный на использовании алгоритма построения FP-дерева с последующим уточнением результатов путём кластеризации. Данный метод демонстрирует приемлемое время выполнения при высоких показателях точности и полноты. В рамках задачи классификации произвольного сообщения продемонстрирована эффективность использования комбинации двух (метрического и основанного на выравнивании последовательностей) способов оценки близости шаблона с входным сообщением. Интерес для дальнейших исследований представляет задача автоматического поиска оптимальных параметров алгоритмов в зависимости от массива входных сообщений.

Ключевые слова: короткие сообщения, выделение шаблонов, классификация сообщений, выравнивание последовательностей, FP-дерево, кластеризация

Введение

В настоящее время множество коммуникационных приложений предоставляют возможность обмена короткими сообщениями. Значительная доля сообщений генерируется автоматически различными сервисами. Содержимое таких рассылок может содержать полезную информацию, но может оказаться и спамом со ссылками, ведущими на вредоносные сайты. Часто подобные сообщения генерируются по одному шаблону с заменой некоторых частей, таких как сама ссылка. Все такие сообщения, соответственно, будут иметь одинаковый набор слов, за исключением некоторой переменной части. В рамках работы рассмотрена проблема выделения автоматически генерируемых сообщений из общего объема коротких текстов, и разработаны алгоритмы, позволяющие выделять шаблоны и классифицировать сообщения.

Задача поиска шаблонов может быть разбита на три этапа: выделение в множестве сообщений групп похожих текстов, выделение шаблона по группе похожих сообщений и

отнесение произвольного сообщения к одному из выделенных шаблонов. Под шаблоном понимается последовательность символов, часть которых фиксирована, а часть может принимать различные значения. Таким образом, определим шаблон как непустую последовательность $a_0 b_0 a_1 b_1 \dots a_{n-1} b_{n-1} a_n$, где $a_i, i = \overline{0,n}$ — постоянные части, $b_j, j = \overline{0,n-1}$ — переменные части шаблона. При этом $a_i, i = \overline{1,n-1}$ и $b_j, j = \overline{0,n-1}$ — непустые последовательности символов, а a_0 и a_n — последовательности символов, возможно пустые.

Будем относить сообщение к одному из шаблонов в том случае, когда в сообщении полностью присутствует фиксированная часть шаблона, а на позициях переменной части стоят произвольные символы. Фактически это означает, что произвольные сообщения потенциально могут быть отнесены к одному шаблону, если количество слов в них одинаковое или почти одинаковое, и есть совпадающие части, которые находятся на близких позициях и порядок которых совпадает. В большинстве задач классификации текстов наличие тех или иных слов в классифицируемом объекте заранее неизвестно, и потому алгоритм заведомо не может быть точным. В случае шаблонов, наоборот, известен набор и порядок слов, которые должны присутствовать в сообщении. Это определяет требования высокой точности классификации, при этом также необходимо учитывать производительность алгоритма классификации для обеспечения возможности обработки потока сообщений.

1. Методы сравнения сообщений

Для того чтобы реализовать классификацию сообщений в первую очередь необходим метод сравнения самих сообщений. Одним из подходов является сравнение последовательностей с помощью алгоритмов выравнивания, применяемых в биоинформатике. На вход таким алгоритмам подаются две последовательности символов. На выходе получается число, называемое счетом и соответствующее степени схожести последовательностей, а также представление последовательностей друг под другом так, чтобы были видны схожие участки. Подобные методы могут быть использованы для сравнения сообщений и выявления шаблонов, так как в данной задаче конечной целью является оценка их схожести и выявление общих и переменных частей сообщения. Одним из методов локального выравнивания последовательностей является алгоритм Смита – Ватермана [1].

В задачах биоинформатики вставка или удаление символа может не влиять значительно на структуру и свойства белка. Напротив, в случае выделения шаблонов добавление или удаление слова может являться признаком того, что сообщения относятся к разным шаблонам. С этой точки зрения, более подходящей является модификация одного из алгоритмов выравнивания, используемая для сравнения исходных текстов программ. При сравнении кодов программ вставка или удаление токена в исходном коде может полностью изменить смысл программы, поэтому в статье [2] учитывается необходимость поиска непрерывных отрезков программ. В данном алгоритме в качестве символов алфавита выступают токены программы, полученные в результате ее лексического анализа. В случае

выявления шаблонов в качестве символов алфавита могут выступать, например, слова сообщения.

С другой стороны, при сравнении исходных кодов изменение порядка некоторых блоков программ может не влиять на работу всей программы, поэтому алгоритм не рассчитан на то, что порядок должен быть фиксированным, как в случае сопоставления сообщений и шаблонов. При поиске шаблона, которому соответствует сообщение, изменение порядка слов недопустимо. Поэтому исходный алгоритм модифицирован путём изменения процедуры поиска последовательности индексов совпадающих частей сообщения по построенной в процессе выравнивания таблице. Для учета порядка сохраняется индекс i строки таблицы, элемент которой был последним учтен в построении последовательности совпадающих частей. На каждом следующем шаге, в отличие от исходного алгоритма, производится поиск максимального элемента не во всем столбце, а только от нулевого до -ого элемента столбца. Таким образом, не может быть осуществлено перехода от более раннего индекса строки к более позднему.

Помимо алгоритмов выравнивания, сравнение сообщений может производиться за счет введения метрики на сравниваемых объектах. В рамках данного подхода результатом сравнения является расстояние между сообщениями, и чем меньше расстояние между двумя объектами, тем более они похожи. Для сравнения сообщений при выделении шаблонов большое значение имеет не только наличие конкретного слова, но и его позиция в тексте. В связи с этим предложена модификация приведённой в [3] формулы вычисления расстояния между текстами в системах поиска некорректных заимствований таким образом, чтобы разница в позициях каждого слова в шаблоне и сообщении учитывалась как вес слова:

$$D = \frac{1}{1+\ln(1+|f_d - f_q|)} \sum_t \left(\frac{\frac{N}{f_t}}{1+|f_{q,t} - f_{d,t}|} \cdot \frac{1}{1+|pos_{t,q} - pos_{t,d}|} \right) \quad (1)$$

где D – расстояние между сообщениями;

N – общее количество шаблонов;

f_t – количество шаблонов, содержащих слово t;

f_d – количество слов в шаблоне d;

f_q – количество слов в сообщении q;

$f_{q,t}$ – количество вхождений слова t в сообщение q;

$f_{d,t}$ – количество вхождений слова t в шаблон d;

$pos_{t,q}$ – позиция слова t в сообщении q;

$pos_{t,d}$ – позиция слова t в шаблоне d.

Необходимо отметить, что, в отличие от алгоритмов выравнивания, основанный на вычислении расстояния способ сравнения сообщений не даёт информации об общих и отличающихся частях сравниваемых объектов.

2. Предобработка сообщений

Для повышения эффективности работы алгоритмов обучения и классификации могут использоваться различные способы подготовки данных. Некоторые методы предобработки приведены в [4] и [5]. Для выделения шаблонов на этапе предобработки сообщение разбивается на отдельные слова. Альтернативным способом является разбиение сообщения на -граммы. Но очевидно, что в шаблонах фиксированные части заведомо состоят из отдельных неизменяемых слов или их наборов, поэтому способ с разбиением на -граммы представляется менее эффективным и не рассматривается.

После разбиения переменные части, состоящие из одного или нескольких слов, могут быть заменены на слова специального вида. Такая предобработка необходима, если для определённого класса сообщений некоторое слово из-за большого количества своих вхождений оказывается отнесено к постоянной части шаблона, а не к переменной. Например, наличие слова «улица» в переменной части сообщения, содержащей адрес, может привести к формированию отдельного избыточного шаблона, описывающего только некоторый подкласс сообщений определённого типа. Однако, данный шаг предобработки может мешать корректному выделению шаблонов, если слова-кандидаты для замены не могут быть определены достаточно точно. Поэтому замена переменных частей сообщения специальными словами целесообразна не во всех ситуациях и требует настройки.

Следующим шагом предобработки является вычисление значений хэш-функций для всех слов. Полученная последовательность значений хэш-функции является результатом предобработки одного сообщения. После предобработки всех сообщений строится инвертированный индекс для быстрого поиска сообщений, содержащих то или иное слово. Таким образом, алгоритмы сравнения будут применяться не к самому сообщению, а к его представлению в виде более короткой последовательности слов или символов.

3. Алгоритмы выявления шаблонов сообщений

Один из самых эффективных способов выявления шаблонов основан на поиске частых наборов с помощью FP-дерева (Frequent-Pattern Tree) [6]. На вход подается множество элементов, по которым формируются признаки. При построении дерева для массива сообщений признаками являются слова без учета позиций. Полученные признаки сортируются по весу, в данном случае — по количеству вхождений в сообщения. Создаётся корень дерева r, не относящийся ни к какому признаку. Далее для каждого элемента x_i входной выборки задается текущая вершина v, изначально равная r. Затем для всех признаков, содержащихся в x_i и упорядоченных по убыванию веса, запускается цикл. Если у вершины v нет дочерней вершины u с текущим признаком f, то создается новая вершина u с этим признаком и поддержкой (весом узла), равной нулю. Иначе выбирается существующая дочерняя вершина u с признаком f. Поддержка вершины u увеличивается на $1/M$, где M — это размер входной выборки, после чего выполняется переход на сле-

дующую итерацию, то есть к следующему признаку текущего элемента выборки. В результате получается FP-дерево, содержащее все частые наборы входной выборки.

Рассматриваемый алгоритм также целесообразно дополнить таким образом, чтобы обеспечить хранение в каждой вершине дерева списка $list_u$ индексов сообщений, соответствующих частому набору из признаков от данной вершины до корня дерева. Такой способ менее эффективен с точки зрения использования памяти, но значительно выигрывает по времени выполнения, так как исключает необходимость поиска сообщений с заданным набором признаков после определения всех частых наборов. Модифицированный алгоритм построения FP-дерева приведён в Листинге 1.

Листинг 1 – Модификация алгоритма построения FP-дерева.

1: упорядочить признаки $f \in F: v(f) \geq \delta$ по убыванию $v(f)$;
построение FP-дерева по выборке X^M
2: для всех $x_i \in X^M$:
3: $v := r$;
4: для всех $f \in F$ таких, что $f(x_i) \neq 0$
5: если нет дочерней вершины $u \in S_v: f_u = f$ то
6: создать новую вершину u; $S_v := S_v \cup \{u\}$;
$f_u := f; c_u := 0; S_u := \emptyset; list_u := \emptyset$
7: $c_u := c_u + \frac{1}{L}; v := u;$
8: $list_u = list_u + [i]$

Алгоритм поиска частых наборов позволяет найти все возможные частые комбинации признаков, однако при выделении шаблонов интерес представляют только самые длинные комбинации. Это связано с тем, что отнесение нескольких сообщений к одному шаблону возможно в том случае, когда в указанных сообщениях совпадает максимальное число слов, причем эти слова должны встречаться достаточно часто. Таким образом, из двух частых наборов, один из которых является подмножеством другого, необходимо оставить только самый длинный. Поэтому поиск частых наборов заменен на поиск признаков в непрерывных цепочках узлов, начинающихся с корня дерева и удовлетворяющих следующему условию: вес последнего узла в найденном поддереве должен быть больше или равен наименьшему допустимому весу, а все его дочерние вершины должны иметь вес, меньший допустимого значения.

Для выделения шаблонов также подходят алгоритмы кластеризации. В качестве альтернативы алгоритму с FP-деревом рассмотрен алгоритм Ланса-Вильямса для иерархической кластеризации сообщений [7]. Расстояние R между кластерами задается формулой Уорда [8]:

$$R = \frac{|W||S|}{|W| + |S|}\rho^2\left(\sum_{w \in W}\frac{w}{|W|}, \sum_{s \in S}\frac{s}{|S|}\right),$$

где W и S – кластеры;

w и s – это элементы кластеров W и S, соответственно;

$\rho(a, b)$ – расстояние между векторами a и b.

На вход алгоритм принимает векторы признаков (слов с их позициями в сообщении), соответствующих элементам входной выборки. Для того, чтобы получить из набора сообщений векторы признаков, выполняется предобработка сообщений, в результате чего каждое сообщение представляется последовательностью признаков. Затем для каждого признака выполняется подсчет количества вхождений его во все входные сообщения. Составляется массив всех признаков, присутствующих в сообщениях, который сортируется по убыванию количества вхождений. После этого отбрасываются те признаки, количество вхождений которых оказывается меньше некоторого заранее заданного значения. При этом для любого признака, являющегося постоянной частью шаблона, пороговое значение не должно превосходить количества сообщений, содержащих данный признак. Слишком малым пороговое значение задавать также не следует, так как будет исключено слишком мало несущественных признаков, что отрицательно повлияет на время выполнения. Результатом данных операций является последовательность $(f_1, \dots, f_n)$ самых частых признаков, отсортированная по частоте. Затем для каждого сообщения составляется вектор $(x_1, \dots, x_n)$, где $x_i = 0$, если признак f_i отсутствует в сообщении, иначе $x_i = 1$.

Как и в случае использования FP-дерева, результатом применения алгоритма кластеризации являются группы похожих сообщений, по которым необходимо выделить структуру шаблона. Можно предположить, что алгоритм кластеризации будет работать значительно медленнее алгоритма на основе FP-дерева. Напротив, алгоритм на основе FP-дерева может оказаться менее точным, так как он не учитывает расстояние между сообщениями, которые попадают в одну группу. В связи с этим рассмотрен способ выделения шаблонов с помощью комбинации указанных алгоритмов. В результате применения алгоритма построения FP-дерева будут получены группы сообщений. Каждая из таких групп, в зависимости от параметров алгоритма, может содержать случайные сообщения (на самом деле не относящиеся ни к какому из шаблонов), а также сообщения, относящиеся к нескольким похожим шаблонам. Последующая кластеризация сообщений в полученных группах позволяет уточнить результат. Количество кластеров при этом может варьироваться на основании выбранного критерия завершения кластеризации [9]. Во-первых, кластеризация может быть прервана при достижении определенного уровня различия между кластерами. Во-вторых, критерием останова может служить значительное отличие расстояния между объединяемыми кластерами на двух последовательных итерациях алгоритма. Наконец, самым простым вариантом может быть задание фиксированного количество кластеров.

Для исключения случайных сообщений дополнительно введён параметр, задающий минимальное количество сообщений в кластере, позволяющее считать, что все входящие

в него сообщения относятся к некоторому шаблону. Таким образом, все полученные кластеры с недостаточным количеством сообщений удаляются.

4. Выделение структуры шаблона по группе похожих сообщений

Для формирования структуры шаблона по группе похожих сообщений можно использовать прогрессивный метод (англ. progressive method) множественного выравнивания [10]. Этот метод основан на попарном выравнивании наиболее близких последовательностей символов. Для задания порядка выбора пар для выравнивания строится бинарное дерево, в котором наиболее близкие пары расположены рядом. Построение дерева сильно замедляет процесс, но в биоинформатике играет важную роль при поиске наибольшего сходства последовательностей. При выделении структуры шаблона порядок выравнивания пар сообщений не важен, поэтому в данном случае построение дерева не требуется.

Таким образом, процедура множественного выравнивания состоит из следующих шагов. Сначала выполняется выравнивание для двух сообщений, и строится первое приближение шаблона. Далее производится постепенное уточнение структуры шаблона путём его выравнивания с каждым из оставшихся сообщений. Очевидно, что при большом количестве сообщений в группе выравнивание шаблона с каждым сообщением является слишком затратным по времени. Учитывая схожесть сообщений в группе, количество рассматриваемых сообщений можно ограничить. Для произвольно выбранного сообщения производится поиск сообщений, содержащих минимальное количество общих с исходным сообщением слов. Это требование гарантирует, что различные варианты переменных частей не будут потеряны. Тем самым формируется группа сообщений меньшего размера, к которой применяется описанный метод множественного выравнивания.

5. Алгоритм классификации сообщений

Под классификацией сообщений понимается отнесение произвольного сообщения к одному из выделенных шаблонов, либо определение того, что сообщение не может быть отнесено ни к одному из них. Для подобных задач также подходят методы биоинформатики, самыми распространенными из которых являются алгоритмы FASTA и BLAST [11]. Данные методы используются для поиска последовательностей символов в базе, наиболее похожих на входную последовательность. Базы последовательностей могут достигать больших размеров, поэтому основной оптимизацией поиска является быстрое сокращение множества кандидатов до минимума. Таким образом, подобные алгоритмы можно применить для поиска шаблона, наиболее близкого к входному сообщению, сравнивая его не с каждым шаблоном, а только с наиболее похожими. После этого для каждого из кандидатов можно применить один из более ресурсоёмких и точных методов сравнения двух сообщений для выявления единственного подходящего шаблона среди оставшихся кандидатов.

Алгоритм классификации, сформированный на основе описанных идей, состоит из следующих шагов. На вход подается сообщение, которое необходимо классифицировать, список шаблонов и построенный по шаблонам инвертированный индекс. Сначала выполняется предобработка сообщения и с помощью инвертированного индекса выбираются шаблоны, имеющие хотя бы одно общее слово с входным сообщением. Следующим шагом является сортировка списка выбранных шаблонов по убыванию близости с рассматриваемым сообщением. На последнем шаге поочередно для каждого шаблона выполняется проверка на наличие в сообщении всех его постоянных частей и их порядка. Результатом выполнения алгоритма является первый шаблон, который удовлетворяет описанным требованиям.

Если сообщение не относится ни к одному из шаблонов, то возможны следующие варианты. Для сообщения, в котором нет ни одной постоянной части ни одного из шаблонов, результат будет получен сразу после обращения к индексу. Если шаблонов с постоянными частями, присутствующими в сообщении, мало, то и список кандидатов будет небольшим, следовательно, сравнение сообщения со всеми элементами списка незначительно повлияет на время выполнения. Возможно и ложное отнесение сообщения к одному из шаблонов, если сообщение содержит все постоянные части данного шаблона в правильном порядке. Наихудшим случаем с точки зрения производительности является наличие в сообщении постоянных частей, принадлежащих значительному количеству различных шаблонов. В этой ситуации необходимо сравнение со всеми элементами списка кандидатов, и время классификации заметно увеличится.

Очевидно, что формирование и сортировка списка шаблонов-кандидатов имеет ключевое значение как для скорости классификации, так и для её качества. Оценка близости рассматриваемого сообщения с множеством шаблонов должна производиться достаточно быстро. С другой стороны, способ оценки должен быть выбран таким образом, чтобы нужный шаблон в большинстве случаев располагался максимально близко к началу списка кандидатов, что позволит минимизировать количество дополнительных проверок на заключительном этапе алгоритма.

В работе рассмотрены три способа оценки близости шаблона с входным сообщением: по количеству совпадающих слов, по расстоянию до сообщения и по счету за выравнивание. Достоинствами первого способа являются простота и скорость, но он не учитывает порядок слов в сообщениях и шаблонах, что должно повлечь увеличение среднего количества проверяемых шаблонов при каждом сопоставлении. Метод упорядочивания шаблонов по расстоянию до сообщения является метрическим. Расстояние между входным сообщением и каждым из шаблонов-кандидатов производится по формуле (1), таким образом этот способ неявно учитывает возможный порядок постоянных частей. Наконец, для оценки близости с помощью выравнивания использована рассмотренная ранее модификация алгоритма для выявления плагиатов. Счётом за выравнивание служит максимальное значение в ячейках построенной таблицы. Выравнивание явно учитывает порядок слов в сообщениях и шаблонах, но является самым ресурсоёмким способом сравнения.

Кроме того, данный способ требует подбора параметров штрафа за удаление или вставку символа, а также оценки при совпадении или несовпадении символов. Вопрос поиска оптимальных параметров выравнивания требует отдельного исследования и в рамках данной работы не рассматривался.

Предполагается, что метрический метод и метод с выравниванием дадут лучшие результаты классификации, чем способ, основанный на поиске шаблона по числу совпадающих слов, в то время как последний должен значительно выигрывать по времени. Поэтому рассмотрена возможность использования комбинации методов: сначала с помощью оптимального по скорости алгоритма может быть сформирован предварительный список шаблонов, а затем первые n элементов этого списка могут быть переупорядочены с использованием самого точного способа оценки.

6. Тестирование алгоритмов

Входными данными для тестирования методов выделения шаблонов и классификации является массив из более чем одного миллиона сообщений, автоматически сформированный за некоторый период времени сервисом рассылки сообщений. Таким образом, все сообщения заведомо являются шаблонными, а также известны количество и структура шаблонов.

Методы выделения шаблонов

Для оценки методов выделения шаблонов использованы общепринятые критерии точности P (precision) и полноты R (recall). Применительно к рассматриваемой задаче точность определяется как отношение количества правильно выделенных шаблонов к общему количеству выделенных шаблонов, а полнота — как отношение количества правильно выделенных шаблонов к общему количеству шаблонов, которое было необходимо выделить. Так как показатели точности и полноты зачастую взаимозависимы, рассматривается также значение -меры, вычисляемое как среднее гармоническое этих показателей. Использование -меры позволяет определить параметры алгоритмов, обеспечивающие наилучшее сочетание точности и полноты. Кроме того, при оценке алгоритмов во внимание принимается время их выполнения.

Сравнение методов выделения шаблонов проводилось для n = 10000 сообщений и n = 1000000 сообщений, выбранных случайным образом из входного массива. Основным параметром, влияющим на остановку процесса разбиения сообщений на группы, соответствующие некоторым шаблонам, является минимальное количество вхождений признака в сообщения, необходимое для учета данного признака при выделении шаблонов. Таким образом, чем меньше минимальное количество вхождений признака, тем больше признаков учитывается при выделении шаблонов. Для тестирования были выбраны три значения параметра c_min, равные $n/200$, $n/500$ и $n/1000$. Результаты тестирования методов выделения шаблонов приведены в таблице 1.

Как и ожидалось, наилучшее сочетание точности и полноты достигается при использовании комбинации алгоритма на основе FP-дерева и кластеризации. По времени работы алгоритм на основе FP-дерева является самым быстрым. Напротив, алгоритм кластеризации оказывается самым медленным, причём при количестве сообщений $n = 1000000$ и более его использование перестает быть рациональным. По этой причине данный алгоритм исключен из сравнения, несмотря на небольшое преимущество при $n = 10000$ по точности и полноте при некоторых значениях параметра c_min. Комбинация алгоритмов на основе FP-дерева и кластеризации показывает приемлемое время работы, незначительно превышающее время работы первого алгоритма. При этом точность и полнота оказывается даже выше, чем при использовании только алгоритма кластеризации. Тестирование при разных значениях параметра c_min наглядно показывает, что правильный выбор значения данного параметра позволяет достичь оптимального сочетания точности и полноты. Также можно заметить, что относительно небольшой размер выборки может оказаться предпочтительным при выделении шаблонов. Это в конечном итоге означает, что предложенный метод выделения шаблонов может применяться для последовательной обработки небольших порций входных данных с целью анализа изменения набора шаблонов, используемых службой рассылки.

Таблица 1. Результаты тестирования методов выделения шаблонов

Алгоритм	$c\_min = n/200$				$c\_min = n/500$				$c\_min = n/1000$			
	P	**R**	**F**	**Время, с**	**P**	**R**	**F**	**Время, с**	**P**	**R**	**F**	**Время, с**
	Результаты для 10 000 сообщений											
FP-дерево	0.85	0.35	0.50	5	0.88	0.81	0.84	5	0.63	0.95	0.76	6
Кластеризация	0.94	0.59	0.73	476	0.92	0.77	0.84	517	0.86	0.69	0.77	72
Комбинация	0.97	0.68	0.80	5	0.94	0.96	0.95	7	0.98	0.96	0.97	8
	Результаты для 1 000 000 сообщений.											
FP-дерево	0.85	0.35	0.50	182	0.88	0.74	0.80	190	0.59	0.92	0.72	202
Кластеризация	–	–	–	>3600	–	–	–	>3600	–	–	–	>3600
Комбинация	0.91	0.43	0.58	261	0.93	0.96	0.95	293	0.73	0.94	0.82	359

Алгоритм классификации сообщений

Тестирование алгоритма классификации заключается в сравнении трёх способов оценки близости шаблона с входным сообщением (по количеству совпадающих слов, по расстоянию до сообщения и по счёту за выравнивание). Основными критериями сравнения являются точность (доля правильно классифицированных сообщений) и общее время классификации всех сообщений тестового набора. Алгоритм подразумевает сортировку списка шаблонов по убыванию их близости к входному сообщению с последующей ресурсоёмкой проверкой соответствия сообщения поочерёдно каждому из шаблонов-кандидатов. Поэтому важным показателем также является среднее количество сопоставляемых шаблонов до обнаружения соответствия. По результатам тестирования к сравне-

нию был добавлен способ, сочетающий предварительную сортировку списка шаблонов-кандидатов по расстоянию до сообщения, отсечение данного списка и уточнения порядка в нём по счёту за выравнивание. Тестирование алгоритма классификации, результаты которого приведены в таблице 2, осуществлялось для $k = 300$ сообщений, выбранных случайным образом из входного массива. Рассмотрение относительно небольшого количества сообщений объясняется необходимостью проверки результатов классификации вручную.

Как и ожидалось, самую высокую точность классификации обеспечивает способ оценки близости по счету за выравнивание. Другие два способа уступают ему по точности, но значительно выигрывают по времени выполнения. Быстрая работа метрического способа обусловлена тем, что в среднем при его использовании подходящим оказывается шаблон из начала списка кандидатов. Этот факт также свидетельствует о том, что данный метод наилучшим образом подходит для предварительной сортировки списка шаблонов-кандидатов перед его отсечением в комбинированном алгоритме. При оценке близости по наибольшему числу совпадений сопоставление производится более чем с половиной всех существующих шаблонов, поэтому и время выполнения увеличивается. Относительно большое количество неподходящих шаблонов в алгоритме на основе выравнивания обосновано тем, что на этапе выравнивания сообщения с шаблоном более значимым является количество и порядок совпадающих слов, но не наличие всех постоянных частей шаблона в сообщении. Таким образом, комбинация предварительной сортировки списка шаблонов-кандидатов по расстоянию до сообщения и уточнение порядка в нём по счёту за выравнивание обеспечивает сохранение высокой точности классификации при значительном сокращении времени выполнения.

Таблица 2. Результаты классификации 300 сообщений

Оценка близости сообщения с шаблоном	**Доля правильно классифицированных сообщений**	**Среднее количество анализируемых шаблонов, шт.**	**Общее время работы,с**
по числу совпадений	0.96	30,6	1,6
по расстоянию	0.96	2,6	0,8
по счету за выравнивание	0.99	17,9	62,5
комбинация	0.99	6,0	10,6

Заключение

В работе рассмотрена проблема выделения автоматически сгенерированных сообщений в массиве коротких текстов, как с точки зрения выделения групп похожих текстов во всём массиве сообщений и определения структуры шаблона, так и с точки зрения отнесения произвольного сообщения к одному из выделенных шаблонов. Сравнение методов, используемых для решения схожих задач в машинном обучении, биоинформатике и для выявления плагиатов в исходных кодах программ позволило предложить способ выделения шаблонов сообщений, основанный на использовании алгоритма построения FP-дерева с последующим уточнением результатов с помощью кластеризации. Результаты тестирования показывают, что данный метод эффективен как по критериям точности и полноты,

так и по времени выполнения. Для решения задачи классификации сообщений разработан алгоритм, использующий при определении близости сообщения и шаблона-кандидата комбинацию метрической оценки и метода, основанного на выравнивании шаблонов. Результаты тестирования подтверждают, что предложенный способ классификации сообщений позволяет получить наиболее точный результат за приемлемое время.

В части выделения шаблонов актуальной проблемой для дальнейшего исследования может являться выбор оптимальных параметров алгоритма, в особенности критерия завершения кластеризации, а также более детальное изучение влияния предобработки сообщений на качество результатов. Кроме того, не был затронут вопрос оптимизации процесса определения структуры шаблона по группе похожих сообщений, например, путём использования методов многомерного выравнивания при построении шаблона. Что касается предложенного алгоритма классификации сообщений, особый интерес представляют поиск оптимальных параметров выравнивания, а также выбор порогового значения при отсечении списка шаблонов-кандидатов.

Список литературы

1. Vermij, E.P. Genetic sequence alignment on a supercomputing platform. Thesis…MSc. Delft: Delft University of Technology, 2011. 87 p.
2. Дубанов А.В. Сравнение исходных текстов программ путем выравнивания последовательностей токенов // Инженерный журнал: наука и инновации. 2014. № 9(33). DOI: 10.18698/2308-6033-2014-9-1318
3. Burrows S., Tahaghoghi S.M.M., Zobel J. Efficient plagiarism detection for large code repositories // Software: Practice and Experience. 2007. Vol. 37. №. 2. Pp. 151-176. DOI: 10.1002/spe.750
4. Schleimer S., Wilkerson D. S., Aiken A. Winnowing: local algorithms for document fingerprinting // SIGMOD'03: Proc. of the 2003 ACM SIGMOD intern. conf. on management of data. N.Y.: ACM, 2003. Pp.76-85. DOI: 10.1145/872757.872770
5. Gunawardena T., Lokuhetti M., Pathirana N., Ragel R., Deegalla S. An automatic answering system with template matching for natural language questions // ICIAFs: Proc. 5th Intern. Conf. on information and automation for sustainability. Piscataway: IEEE, 2010. Pp. 353-358. DOI: 10.1109/ICIAFS.2010.5715686
6. Han J., Pei J., Yin Y. Mining Frequent Patterns without Candidate Generation // Newsletter ACM SIGMOD Record. 2000. Vol. 29. № 2. Pp.1-12. DOI: 10.1145/335191.335372
7. Gupta G., Strehl A., Ghosh J. Distance Based Clustering of Association Rules // Intelligent Engineering Systems Through Artificial Neural Networks: Proceedings of Artificial neural networks in engineering conf. (ANNIE). N.Y.: ASME Press, 1999. Pp. 759–764.
8. De Amorim R.C. Feature Relevance in Ward's Hierarchical Clustering Using the Lp Norm // Journal of Classification. 2015. Vol. 32. № 1. Pp. 46-62. DOI: 10.1007/s00357-015-9167-1

9. Manning C.D., Raghavan P., Schutze H. Hierarchical Clustering // Manning C.D., Raghavan P., Schutze H. Introduction to Information Retrieval. Camb.: Cambridge Univ. Press, 2009, pp. 377-401. DOI: 10.1017/CBO9780511809071
10. Neuwald A.F., Altschul S.F. Bayesian Top-Down Protein Sequence Alignment with Inferred Position-Specific Gap Penalties // PLoS Computational Biology. 2016. Vol. 12. №. 5. Pp. 1-21. DOI: 10.1371/journal.pcbi.1004936
11. Pearson W.R., Lipman D.J. Improved tools for biological sequence comparison // Proc. of the National Academy of Sciences. 1988. Vol. 85. № 8. Pp. 2444-2448. DOI: 10.1073/pnas.85.8.2444

Science & Education of the Bauman MSTU
Electronic journal
ISSN 1994-0408

Science and Education of the Bauman MSTU, 2016, no. 10, pp. 117–131.

DOI: **10.7463/1016.0848929**

Received: 04.09.2016
Revised: 18.09.2016

Pattern Identification and Retrieval in Short Text Messages

N.S. Virtseva[1], **I.E. Vishnyakov**[1,*]

*vishnyakov@bmstu.ru

[1]Bauman Moscow State Technical University, Moscow, Russia

Keywords: short text message, pattern retrieval, short message classification, sequence alignment, FP-growth algorithm, clustering

The problem of finding patterns in a set of short text messages is relevant in the identification and analysis of automatically generated messages, including adware and malware. A pattern refers to a sequence of tokens, some of which are fixed, and the rest may take arbitrary values.

Pattern retrieval includes the steps of preprocessing the messages from the input set, identifying groups of similar messages, which probably belong to the same pattern, and defining the pattern structure according to the groups. During preprocessing the messages are split into tokens, some of them are special ones and substitute inherently variable parts of the messages (dates, addresses, hyperlinks, etc.). A method to identify the groups of similar messages is based on a modified FP-growth algorithm with further refinement of the results through clustering. This method demonstrates an acceptable performance with high level of precision and recall (F-score in some tests is close to maximum). Multiple alignments of messages, which belong to the same group, are performed to identify pattern structure (its constant and variable parts).

The classification of an arbitrary message implies its assignment to one or none of the defined patterns. A message should be assigned to the pattern if it contains all constant parts of the pattern in the specified order. The choice of patterns for matching is based on their proximity to the input message. Three ways of proximity assessment (by the number of matching words, by the distance to the message and by the alignment score) were tested to estimate their efficiency in terms of execution time and accuracy. As a result, a combination of metric and alignment-based methods was introduced to make a list of possible matches.

An automatic search for optimal parameters of the algorithms, as well as a more detailed study of the influence of message preprocessing on the quality of the results depending on the set of input messages could be of interest for further research.

References

1. Vermij, E.P. Genetic sequence alignment on a supercomputing platform. Thesis…MSc. Delft: Delft University of Technology, 2011. 87 p.

2. Dubanov A.V. *Sravneniye iskhodnykh tekstov programm putem vyravnivaniya posledovatelnostey tokenov* [The comparison of program sources using the sequence alignment of tokens]. *Inzhenernyy zhurnal: nauka i innovatsii* [Engineering Journal: Science and Innovation], 2014, no. 9(33). DOI: 10.18698/2308-6033-2014-9-1318
3. Burrows S., Tahaghoghi S.M.M., Zobel J. Efficient plagiarism detection for large code repositories. *Software:Practice and Experience*, 2007, vol. 37, no. 2, pp. 151-176. DOI: 10.1002/spe.750
4. Schleimer S., Wilkerson D. S., Aiken A. Winnowing: local algorithms for document fingerprinting. *SIGMOD'03: Proc. of the 2003 ACM SIGMOD intern. conf. on management of data.* N.Y.: ACM, 2003, pp.76-85. DOI: 10.1145/872757.872770
5. Gunawardena T., Lokuhetti M., Pathirana N., Ragel R., Deegalla S. An automatic answering system with template matching for natural language questions. *ICIAFs: Proc. 5th Intern. Conf. on information and automation for sustainability.* Piscataway: IEEE, 2010, pp. 353-358. DOI: 10.1109/ICIAFS.2010.5715686
6. Han J., Pei J., Yin Y. Mining Frequent Patterns without Candidate Generation. *Newsletter ACM SIGMOD Record*. 2000. Vol. 29. № 2. Pp.1-12. DOI: 10.1145/335191.335372
7. Gupta G., Strehl A., Ghosh J. Distance Based Clustering of Association Rules. *Intelligent Engineering Systems Through Artificial Neural Networks: Proceedings of Artificial neural networks in engineering conf. (ANNIE).* N.Y.: ASME Press, 1999. Pp. 759–764.
8. De Amorim R.C. Feature Relevance in Ward's Hierarchical Clustering Using the Lp Norm. *Journal of Classification*, 2015, vol. 32, no. 1, pp. 46-62. DOI: 10.1007/s00357-015-9167-1
9. Manning C.D., Raghavan P., Schutze H. Hierarchical Clustering. Manning C.D., Raghavan P., Schutze H. *Introduction to Information Retrieval*. Camb.: Cambridge Univ. Press, 2009, pp. 377-401. DOI: 10.1017/CBO9780511809071
10. Neuwald A.F., Altschul S.F. Bayesian Top-Down Protein Sequence Alignment with Inferred Position-Specific Gap Penalties. *PLoS Computational Biology*, 2016, vol. 12, no. 5, pp. 1-21. DOI: 10.1371/journal.pcbi.1004936
11. Pearson W.R., Lipman D.J. Improved tools for biological sequence comparison. *Proc. of the National Academy of Sciences*, 1988, vol. 85, no. 8, pp. 2444-2448. DOI: 10.1073/pnas.85.8.2444