УДК 519.688

Разработка программного обеспечения для выявления источников общественно-опасной информации в социальной сети "ВКонтакте"

А. А. Березин, Э. Ф. Гайнутдинов, А. С. Максимов, Е. Ю. Никитина
Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
eldar_gaynutdinov@mail.ru, neyu@psu.ru; +79519220712, +79028305928

Рассмотрена социальная сеть "ВКонтакте" на наличие социально опасной информации. Осуществлена оценка требований и возможностей по поиску данной информации. Произведена разработка программного комплекса, позволяющего собирать информацию из открытых источников, проводить поиск заданных ключевых слов в полученной информации и осуществлять оценку данной информации на релевантность заданной теме.

Ключевые слова: *открытые источники; ВКонтакте; опасная информация; поиск информации; получение информации; большой объем информации; социальные сети; точный поиск; неточный поиск; ключевые слова; программа; программный комплекс; ранжирование; релевантность.*

DOI: 10.17072/1993-0550-2018-3-104-110

Введение

С появлением и распространением сети Интернет, человек может получать и распространять большие объемы информации практически мгновенно. Данная информация может быть как закрытой, так и открытой, т.е. доступной всем, у кого есть доступ в Интернет. Однако не все данные, находящиеся в открытом доступе, являются безопасными. Сама информация может представлять опасность, угрозу, в особенности для несовершеннолетних. Именно об информации такого рода и будет дальнейшее исследование.

В первую очередь, необходимо определиться с источниками данных. Одним из самых доступных вариантов распространения новостей являются социальные сети.

Крупнейшая социальная сеть России – "ВКонтакте" – насчитывает более 82 млн активных пользователей в месяц, а версия этой сети для персональных компьютеров занимает 5-е место в мире среди самых посещаемых сайтов. Более 50 % ее аудитории – дети и молодые люди до 24 лет [1]. Большая часть информации, расположенной в этой сети, является открытой (данные на "стенах" пользователей и в открытых группах). Исключения составляют только личные чаты и закрытые группы. Информация, находящая в открытом доступе, практически не проходит процесс модерации в связи с огромными потоками данных: он проводится только после жалоб пользователей. А потому считать безопасной всю открытую информацию, находящуюся в социальной сети, нельзя. В качестве исходных данных для разрабатываемого программного обеспечения используются записи пользователей сети "ВКонтакте": новости, комментарии, описания к медиа-контенту, обсуждения в группах.

В настоящее время в социальной сети "ВКонтакте" присутствует информация, которая не только запрещена законами Россий-

ской Федерации, но и несет прямую или косвенную опасность гражданам: националистические призывы, лозунги в поддержку террористических организаций. К примеру, не так давно возникло целое движение "Синий кит" – группы лиц требовали от подростков выполнять ряд заданий, последним из которых было самоубийство. Жертвами этой "игры" по меньшей мере, стало 130 детей, совершивших суицид [2]. Поскольку доля подростков, общественное сознание которых полностью еще не сформировано, использующих социальную сеть, очень высока, то воздействовать на них и манипулировать ими с помощью инструментов социальной сети не представляет трудностей. Это свидетельствует о том, что необходимо своевременно выявлять опасную информацию и предотвращать ее распространение.

В настоящее время существующими аналогами описываемой работы можно назвать автоматизированную систему "Демон Лапласа" и проект "Герда. Старшая сестра в Интернете" Программный комплекс "Демон Лапласа" – это продукт, позволяющий производить мониторинг данных в социальных сетях. Его серьезный недостаток состоит в том, что этот продукт является платным, а также для его функционирования необходимо подключение к серверу.

Второй программный комплекс – "Герда. Старшая сестра в Интернете" – это пермский узкоспециализированный продукт, позволяет проверить, состоит ли пользователь сети "ВКонтакте" в опасных группах ("группы смерти"). Однако данное программное обеспечение не обнаруживает группы в социальной сети, если пользователь скрыл их настройками приватности.

Программный комплекс, создаваемый в рамках данной работы, реализует поиск в социальной сети "ВКонтакте" по различным фильтрам (регион, ключевые слова, город и т.д.), при этом получаемые результаты поиска не будут ограничены (по сравнению со встроенным поиском сети "ВКонтакте"). Кроме того, наш программный продукт будет бесплатным для правоохранительных органов.

В силу вышеуказанного является актуальной задача осуществления поиска общественно опасной информации в открытых источниках информации, в частности, в социальной сети "ВКонтакте".

1. Описание программного комплекса

Нами предлагается реализация программного комплекса для поиска информации в социальной сети "ВКонтакте", а также определения в автоматическом режиме, является ли найденная в сети информация социально-опасной.

Перед непосредственной реализацией программного комплекса необходимо ответить на ряд вопросов: как получить информацию, как определить, что информация опасна и что делать с опасной информацией.

На последний вопрос можно ответить сразу – все решения и действия по поводу незаконной информации могут принимать только уполномоченные на это органы Российской Федерации, поэтому необходимо лишь передавать им найденную информацию.

Прежде чем отнести информацию к общественно опасной, необходимо определить критерии "опасности". Один из вариантов решения – определение по набору ключевых слов. Именно такой способ определения социально-опасной информации и был применен в разрабатываемом программном комплексе. Для определения был составлен набор ключевых слов, который использовался для анализа информации в социальной сети "ВКонтакте". Программный комплекс на основании присутствия или отсутствия в тексте указанных ключевых слов делает вывод о том, насколько опасен текст.

Реализация получения информации может варьироваться в зависимости от источника информации, но сохранять общий принцип его построения. Основная цель части программного обеспечения, отвечающей за получение информации из открытых источников – выполнение действий быстро и эффективно. Для достижения максимальных показателей необходимо использовать встроенные методы получения информации из источников (API), если таковые имеются. Если таких методов нет, то необходимо получать и выделять необходимую информацию из HTTP-запросов.

Можно выделить три обособленных модуля программного комплекса:

1. модуль сбора информации – отвечает за получение информации из открытых источников и передачу ее для дальнейшей обработки;

2. модуль поиска ключевых слов – отвечает за поиск ключевых слов в большом объеме информации;

3. модуль ранжирования документов – отвечает за определение того, является ли информация опасной.

2. Сбор информации

Для получения информации из социальной сети "ВКонтакте" существуют как веб-интерфейс, так и встроенные методы взаимодействия. Второй вариант позволяет получать информацию по запросу, без необходимости ее дополнительной обработки.

В социальной сети "ВКонтакте" всю информацию можно разделить на две части: принадлежит группе или принадлежит пользователю. Обработка запросов на получение информации из сети "ВКонтакте" не зависит от того, кому принадлежит информация, поэтому автоматизация ее получения заключается в переборе групп и пользователей сети.

Ввиду большого количества пользователей и групп полный перебор практически невозможен, вследствие чего необходимо сократить выборку, например, собирать данные для определенного города. Устанавливая различные ограничения для выборок, в конечном итоге можно получить весь объем информации, хранящейся в социальной сети.

Также приходится учитывать, что существуют ограничения со стороны источников информации на доступ к этой информации. В сети "ВКонтакте" есть ограничение на количество производимых запросов в секунду и на количество запросов одного типа в сутки [3].

3. Поиск с помощью ключевых слов

Из загруженных данных происходит выборка информации на основе поиска ключевых слов, которые составляются экспертами. Используются два способа поиска: точное и неточное совпадения. В случае неточного совпадения допускается отличие в нескольких символах: случаи, когда допущена ошибка, или слово употребляется в другом числе или падеже.

3.1. Точный поиск

Для нахождения точного совпадения ключевых слов со словами полученного из социальной сети текста, в данной работе используется алгоритм Ахо-Корасика [4].

Суть алгоритма заключена в использовании структуры данных – бора и построения по нему конечного детерминированного автомата. Бор (префиксное дерево) – структура данных, позволяющая хранить ассоциативный массив, ключами которого являются строки.

Согласно алгоритма строится конечный автомат (рис. 1), которому в качестве исходных данных передается строка поиска. Автомат получает по очереди все символы строки и переходит по соответствующим ребрам. Если автомат пришел в конечное положение, соответствующее слово из списка присутствует в строке поиска.

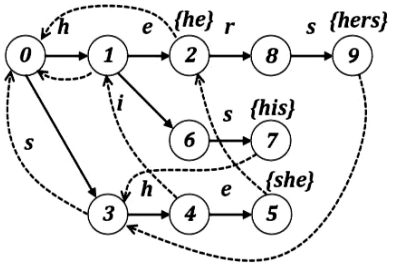

Рис. 1. *Конечный автомат, распознающий слова* he, she, his, hers

3.2. Неточный поиск

Алгоритмы неточного поиска характеризуются метрикой — функцией расстояния между двумя словами, позволяющей оценить степень их сходства в данном контексте. Расстояние Левенштейна (также редакционное расстояние или дистанция редактирования) между двумя строками в теории информации и компьютерной лингвистике – это минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую. Для расчета расстояния Левенштейна между двумя строками используется метод Вагнера–Фишера [5].

Для неточного поиска используется метод N-грамм. N-граммой называется последовательность из N элементов – букв, звуков, слогов или слов. Последовательность из двух последовательных элементов часто называют *биграмма*, последовательность из трех элементов называется *триграмма.* Например, для слова "крокодил" триграммами являются следующие буквенные сочетания: "кро", "рок", "око", "код", "оди", "дил".

Алгоритм основывается на следующем принципе: если слово А совпадает со словом Б с учетом нескольких ошибок, то с большой долей вероятности у них будет хотя бы одна общая подстрока длины N [6]. В данном случае подстроки длины N и являются N-граммами.

Работу данного алгоритма можно разбить на 2 стадии:

1) индексация ключевых слов;

2) поиск слова в списке.

Во время первой стадии каждое слово из набора ключевых слов разбивается на N-граммы, а затем это слово попадает в списки для каждой из этих N-грамм. Так слово "крокодил" попадет в список слов для каждой своей триграммы (рис. 2).

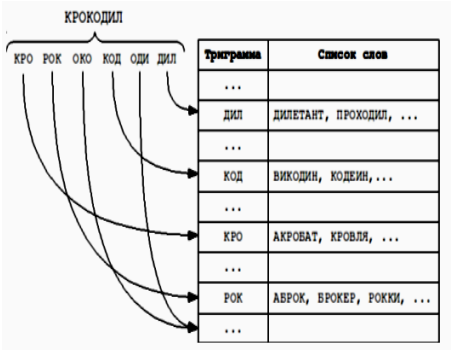

Рис. 2. *Индексация слова "Крокодил"*

На второй стадии искомое слово также разбивается на триграммы. Если триграмма содержится в таблице, полученной на первой стадии, то между искомым и каждым ключевым словом, содержащим эту триграмму, вычисляется расстояние Левенштейна, которое характеризует количество ошибок. В результате принимаются слова, расстояние Левенштейна которых минимально. В данной работе оно не превышает 2, т.е. подразумевается, что в слове могло быть допущено не более 2 ошибок.

4. Ранжирование документов

После того, как выполнен предыдущий этап поиска (выборка по ключевым словам), необходимо ранжировать полученные результаты, то есть определить релевантность документов: насколько они удовлетворяют критериям запроса. Так, например, при поиске двух ключевых слов "митинг" и "революция" документ, содержащий одно слово "митинг" будет иметь релевантность ниже, чем документ, содержащий два таких слова; документ, содержащий слова "митинг" и "революция" будет иметь самую высокую релевантность среди трех приведенных документов.

В качестве методов ранжирования рассматриваются такие, как TF-IDF, Okapi BM25 (модифицированный алгоритм TF-IDF), метод на основе косинусного расстояния (как мера похожести двух векторов).

4.1. Формула TF-IDF

Формула TF-IDF оценки значимости определенного слова в документе относительно всей выборки.

Согласно TF-IDF, вес слова пропорционален количеству употреблений в текущем документе и обратно пропорционален частоте употребления слова в других документах выборки. Если слово используется в этом документе чаще, чем в других, то оно имеет большую значимость для него [7].

TF (англ. term frequency) – это частотность слова, показывающая, насколько часто оно употребляется в документе. В длинном тексте вхождений термина может быть значительно больше, чем в коротком, хотя это вовсе не означает, что термин точнее отвечает требованиям посетителей социальной сети. Чтобы уравнять шансы длинных и коротких документов, используется отношение количества употребления слова к общему количеству слов в документе (4.1.1)

$$TF(t_i,d)=\frac{n(t_i,d)}{\sum_j (t_j,d)}, \qquad (4.1.1)$$

где $n(t_i,d)$ – число вхождений слова t_i в документ d, $\sum_j (t_j,d)$ – общее количество всех слов документа d.

IDF (англ. inverse document frequency) – обратная частота употребления слова в документе. Данный показатель вводится в формулу оценки значимости для снижения веса часто употребляемых слов в выборке. После расчета TF получается только частотность, но слова сами по себе еще равнозначны. Часто употребляемые предлоги и союзы не могут быть сопоставимы по важности с другими словами, так как не несут смысловой нагрузки. Множитель IDF позволяет перераспределить вес значимости слов. Редкие слова получают более высокую значимость, а часто употребляемые – низкую (4.1.2).

$$IDF(t_i,b)=\log\frac{\sum_j(d_j,b)}{\sum_k(d(t_i)_k,b)}, \quad (4.1.2)$$

где $\sum_j(d_j,b)$ – общее количество документов d в выборке b, $\sum_k(d(t_i)_k,b)$ – общее количество документов d, в которых встречается слово t_i. Основание логарифма может быть любым, так как IDF является относительной мерой.

Итоговая формула выглядит следующим образом:

$$TFIDF(t_i,d,b)=TF(t_i,d)*IDF(t_i,b). \quad (4.1.3)$$

Релевантность по запросу *Q* (здесь и далее запрос $Q=q_1,q_2,\ldots,q_i,\ldots,q_m$ – список ключевых слов) равна сумме релевантностей по всем словам запроса:

$$score(d,Q)=\sum_{i=1}^{m}TF(t_i,d)*IDF(t_i,b). \quad (4.1.4)$$

4.2. Формула OKAPI BM25

При увеличении вхождений ключевого слова в документ пропорционально увеличивается значение TF. Таким образом, добавление вхождений ключевых слов в документ значительно повышает его релевантность. Функция BM25 должна была устранить этот недостаток.

Функция BM25 (англ. "best match"), часто ее называют также Okapi BM25, по названию поисковой системы Okapi, где она была использована впервые, – это поисковая функция на неупорядоченном множестве документов, которые она оценивает на основе встречаемости слов запроса в каждом документе [8].

В функцию BM25 внедрены свободные коэффициенты, которые могут принимать различные значения. При увеличении вхождений слов в документ релевантность асимптотически стремится к определенному коэффициентами a, k_1 значению, в то время как в классической формуле TF стремится к бесконечности.

Оценка релевантности документа d по запросу Q, содержащего слова по формуле BM25:

$$BM25(d,Q)=\\ =\sum_{i=1}^{m}IDF(q_i,b)*\\ *\frac{TF(q_i,d)*(k_1+1)}{TF(q_i,d)+k_1*\left(1-a+a*\frac{\sum_k(t_k,d)}{AVGLEN(d_j,b)}\right)}, \quad (4.2.1)$$

где $TF(q_i,d)$ – частота слова q_i в документе d, рассчитывается по формуле (4.1.1); a,k_1 – свободные коэффициенты, чаще всего выбираемые $k_1=2$, $a=0{,}75$; $\sum_k(t_k,d)$ – количество слов в документе d, $AVGLEN(d_j,b)$ – средняя длина документа в выборке, рассчитывается по (4.2.2), $IDF(q_i,d)$ – обратная частота слова q_i в документе d (4.2.3).

$$AVGLEN(d_j,b)=\frac{\sum_j\sum_k(t_k,d_j)}{\sum_k(d_k,b)}, \quad (4.2.2)$$

где $\sum_j\sum_k(t_k,d_j)$ – сумма слов в каждом документе d_j выборки b (общее количество слов в выборке), $\sum_k(d_k,b)$ – количество документов в выборке.

$$IDF(q_i,b)=\log\frac{\sum_j(d_j,b)-n(q_i,b)+0{,}5}{n(q_i,b)+0{,}5}, \quad (4.2.3)$$

где $\sum_j(d_j,b)$ – количество документов в выборке, $n(q_i,b)$ – число документов в выборке b, содержащих слово q_i, 0,5 – псевдоотсчет частоты, введенный в формулу на основании данных вероятностной модели Робертсона для улучшения оценки IDF.

Как и в TF-IDF, в формуле Okapi при отсутствии слова из запроса в документе релевантность этого слова равна нулю; чем больше слов из запроса встречается в документе, а также чем больше их частота употребления, тем больше будет итоговая оценка.

IDF принимает отрицательные значения для слов, входящих более чем в половину документов. Часто встречающиеся слова сильно искажают итоговую оценку документа. При наличии двух почти одинаковых документов, отличающихся только одним часто употребляемым словом, приоритет получит документ, не содержащий его.

Данный факт далек от того, чтобы дать адекватную оценку качества документа, поэтому используется один из двух из вариантов:

1) отрицательные значения в сумме вообще игнорируются;

2) накладывается нижняя граница, при достижении которой значение IDF считается фиксированным числом.

4.3. Косинус угла между векторами текстов

Документ в векторной модели рассматривается как неупорядоченное множество термов. Термами в информационном поиске называют слова, из которых состоит текст.

Вес терма в документе можно определить различными способами – "важность" слова для идентификации данного текста (например, количество употреблений терма в документе) [9].

Все термы, которые встречаются в документах обрабатываемой коллекции, можно упорядочить. Если для некоторого документа выписать по порядку веса всех термов, включая те, которых нет в этом документе, получится вектор, который и будет представлением данного документа в векторном пространстве. Размерность этого вектора, как и размерность пространства, равна количеству различных термов во всей коллекции, и является одинаковой для всех документов.

$d_j = (t_{1j}, t_{2j}, \ldots, t_{mj})$ – векторное представление *j*-го документа, t_{ij} – вес *i*-го терма в *j*-м документе. Располагая таким представлениием для всех документов, можно находить расстояние между точками пространства и тем самым решать задачу подобия документов – чем ближе расположены точки, тем больше похожи соответствующие документы.

В случае поиска документа по запросу, запрос тоже представляется как вектор того же пространства – и можно вычислять соответствие документов запросу.

Для полного определения векторной модели необходимо указать, каким именно образом будет вычисляться вес терма в документе.

Существует несколько стандартных способов задания функции взвешивания:

• булевский вес – равен 1, если терм встречается в документе, и 0 в противном случае;

• TF (частота терма) – вес определяется как функция от количества вхождений терма в документе;

• TF-IDF (частота терма – обратная частота документа) – вес определяется как произведение функции от количества вхождений терма в документ и функции от величины, обратной количеству документов коллекции, в которых встречается этот терм.

Для того чтобы определить релевантность каждого документа в выборке, для каждого эквивалентного вектора необходимо рассчитать косинусное расстояние. Косинусное расстояние – это мера похожести двух векторов. Скалярное произведение векторов a и b и косинус угла θ между ними связаны следующим соотношением:

$$a \cdot b = \|a\| \, \|b\| \cos\theta. \qquad (4.3.1)$$

Имея два вектора *A* и *B*, можно получить косинусное расстояние – $\cos\theta$:

$$\cos\theta = \frac{A \cdot B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{m} A_i * B_i}{\sqrt{\sum_{i=1}^{m} (A_i)^2} * \sqrt{\sum_{i=1}^{m} (B_i)^2}}. \qquad (4.3.2)$$

В качестве весов в векторной модели будет использоваться метрика TF IDF, рассчитанная по формуле (4.1.3). Таким образом, формула (4.3.2) примет вид

$$score(d,Q) = \frac{\sum_{i=1}^{m} TFIDF(w_i,d,b) * TFIDF(w_i,Q,b)}{\sqrt{\sum_{i=1}^{m} (TFIDF(w_i,d,b))^2} * \sqrt{\sum_{i=1}^{m} (TFIDF(w_i,Q,b))^2}}. \qquad (4.3.3)$$

Заключение

В ходе проведенной работы был реализован первый прототип данного программного комплекса, который получает информацию из открытых источников в социальной сети "ВКонтакте" и на основании ключевых слов делает вывод о принадлежности данной информации к заданной теме. Тему для поиска можно варьировать путем изменения набора ключевых слов. В качестве эксперимента было проанализировано 10 ГБ исходных данных на нескольких группах г. Перми.

Развитие программного комплекса, на наш взгляд, возможно в следующих направлениях:

• увеличение скорости и эффективности поиска путем поиска словоформ, использования контекста и машинного обучения;

• более эффективное взаимодействие между модулями;

• использование различных источников информации (помимо социальной сети "ВКонтакте").

Список литературы

1. *Top* Websites in Russian Federation. URL: https://www.similarweb.com/topwebsites/russian-federation (дата обращения: 30.01.2018).
2. *Группы* смерти. URL: https://www.novayagazeta.ru/articles/2016/05/16/68604-gruppy-smerti-18 (дата обращения: 30.01.2018).
3. *Запросы* к API ВКонтакте. URL: https://vk.com/dev/api_requests (дата обращения: 29.01.2018).
4. *Alfred V. Aho, Margaret J. Corasick.* Efficient string matching: An aid to bibliographic search – Communications of the ACM, 1975. 333 с.
5. *Метод* динамического программирования Вагнера и Фишера URL: http://algolist.manual.ru/search/lcs/vagner.php (дата обращения: 01.02.2018).
6. *Esko* Ukkonen Approximate string-matching with q-grams and maximal matches – Theoretical Computer Science 92, 1992. С. 192–211.
7. *Алгоритм* Okapi BM25 – модификация формулы TF-IDF ранжирования документов. URL: http://weblinepromo.ru/blog/algoritm-okapi-bm25-modifikaciya-formuly-tf-idf-ranzhirovaniya-dokumentov (дата обращения: 01.02.2018).
8. *Okapi* BM25. URL: https://ru.wikipedia.org/wiki/Okapi_BM25 (дата обращения: 01.02.2018).
9. *Vector* Space Model для семантической классификации текстов. URL: https://habrahabr.ru/sandbox/18635/ (дата обращения: 01.02.2018).

Development of software to identify sources of socially dangerous information in the social network "VKontakte"

A. A. Berezin, E. F. Gainutdinov, A. S. Maksimov, E. Yu. Nikitina
Perm State University; 15, Bukireva st., Perm, 614990, Russia
eldar_gaynutdinov@mail.ru, neyu@psu.ru; +79519220712, +79028305928

Open sources of information were studied for the presence of socially dangerous information. The requirements and possibilities for finding this kind of information were assessed. A software package was developed that makes it possible to collect information from open sources, in particular, from the social network "VKontakte", search for specified keywords within the received information, and evaluate the relevance of this information to the given topic.

Keywords: o*pen sources; VKontakte; dangerous information; information search; receiving information; large amount of information; social networks; accurate search; inaccurate search; keywords; program; software package; ranging; relevance.*