УДК 81'322.2 DOI 10.25559/SITITO.2017.3.460

Давыдова Ю.В.

Орловский государственный университет имени И.С. Тургенева, г. Орел, Россия

МОДЕЛЬ ОШИБОК ДЛЯ НЕЧЕТКОГО ТЕКСТОВОГО ПОИСКА В ЗАДАЧЕ МОНИТОРИНГА ВИРТУАЛЬНЫХ СОЦИАЛЬНЫХ СЕТЕЙ ДЛЯ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННО-ПСИХОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ЛИЧНОСТИ

Аннотация

Мониторинг виртуальных социальных сетей с целью выявления угроз информационно-психологической безопасности личности является актуальной задачей. Под угрозами понимается пропаганда суицида, экстремизма, терроризма, наркотиков. В процессе мониторинга осуществляется поиск по ключевым словам в текстах сообщений пользователей, который существенно осложняется наличием большого количества ошибок. В данной статье предлагается использовать модель ошибок, интегрированную в нечеткий текстовый поиск, для повышения эффективности мониторинга. Приведена оценка параметров статистической компоненты модели ошибок на основе анализа корпуса сообщений.

Ключевые слова

Мониторинг виртуальных социальных сетей; нечеткий текстовый поиск; динамическое программирование; обработка ошибок; модель зашумленного канала, модель ошибок на основе правил.

Davydova Yu.V.

Orel State University, Orel, Russia

ERROR MODEL FOR FUZZY TEXT SEARCH IN THE TASK OF ONLINE SOCIAL NETWORKS MONITORING FOR PROVIDING INFORMATION-PSYCHOLOGICAL SECURITY OF A PERSON

Abstract

Monitoring of online social networks with the aim of detecting threats to information-psychological security of a person is an urgent task. Threats mean suicide propaganda, propaganda of extremism and terrorism, drug propaganda. Keyword searching implemented in the monitoring process is complicated by users' misspellings. In this article error model integrated with fuzzy text search is proposed to make monitoring more effective. Parameters of statistical component of error model were estimated using collection of posts and are given in the present work.

Keywords

Online social networks monitoring; fuzzy text search; dynamic programming; misspellings processing; noisy channel model; rule-based error model.

Введение

Мониторинг виртуальных социальных сетей (ВСС) для различных целей в последнее время становится все более востребованным, аудитория ВСС постоянно растет. По данным Brand Analytics [1] за май 2017 года только в сети «ВКонтакте» количество активных авторов составило 25,7 млн. человек, количество опубликованных сообщений – 310 млн. В контексте данного исследования мониторинг является одним из инструментов обеспечения информационно-психологической безопасности личности, он предназначен для выявления материалов террористической и экстремистской направленности, пропаганды суицида, пропаганды и незаконного распространения наркотических средств и психотропных веществ в ВСС, а также материалов, потенциально относящихся к вышеперечисленным тематикам [2]. Высокая скорость распространения информации, возможность обращения к

аудитории напрямую, низкая степень цензуры делают ВСС хорошим инструментом в организации противоправной деятельности, сложность выявления которой заключается не только в больших объемах данных, которые необходимо своевременно отслеживать, но и в эффективном анализе публикуемых пользователями сообщений.

Существует множество работ, посвященных различным аспектам анализа контента и поведения пользователей ВСС. На рынке представлены как платные, так и бесплатные продукты различного функционала для мониторинга и анализа ВСС. В [3] приведен обзор информационных систем анализа сетей исходя из целей и методов анализа, функций систем и ряда других параметров. В работе выделены уровни анализа ВСС – мониторинг, анализ, прогнозирование процессов в ВСС и управление сетями. Акцент делается на необходимости развития последних двух уровней и улучшения методов анализа ВСС, которые в целом сводятся к различным статистическим методам и методам теории графов (отдельно выделяется анализ тональности текста). Следует отметить, что исследований, посвященных выявлению противоправных действией пользователей довольно мало. В работе [4] рассматриваются вопросы информационного противоборства, при этом исследуются поведение и влияние пользователей, распространение мнений с использованием теории графов, теоретико-игровых моделей, имитационного моделирования и др., т.е. объект исследования – социальные сети как совокупность пользователей (агентов) и связей между ними. В [5], помимо предложенного авторами метода автоматического аннотирования сообщений пользователей ВСС и выявления ключевых слов, приводится обзор работ, посвященных используемым методам в задачах выявления террористической и экстремисткой активности. По результатам обзора делается вывод, что чаще всего используются методы классификации и кластеризации, сетевой анализ.

Автором данной работы также было выявлено, что в исследованиях, посвященных анализу контента в ВСС, чаще всего описывается анализ более высокого уровня (например, классификация сообщений), предполагающий дальнейшую обработку выделенных из текста слов, но в открытом доступе отсутствуют публикации, посвященные эффективному поиску по ключевым словам в процессе мониторинга. Однако этот этап является важным, в том или ином виде он присутствует в большинстве информационных систем мониторинга и анализа ВСС, поскольку он является первичным фильтром для отбора сообщений для дальнейших исследований. Выявление слов в сообщениях может быть некорректным из-за различного рода ошибок, что в свою очередь в той или иной степени влияет на результаты последующего анализа. Коммуникации в ВСС характеризуются использованием неформального стиля, употреблением сленга [6], большим количеством орфографических ошибок и опечаток по сравнению, например, с новостными лентами. При обработке текста чаще всего используется стемминг, а поиск может организовываться только по хештегам, например, как в [7]. Однако при использовании хештегов пользователи стремятся к унифицированному написанию, что снижает количество ошибок и облегчает процедуру поиска. Стемминг предназначен для нормализации текста, но он не решает проблему обработки некорректного написания слов. Поиск ключевых слов среди неструктурированного текста является более сложной задачей. Для снижения чувствительности поиска к ошибкам в текстах сообщений и расширения поисковой выдачи предлагается использовать нечеткий поиск с интегрированной в него моделью ошибок.

1 Особенности задачи нечеткого поиска в текстах сообщений пользователей ВСС и методы ее решения

Формально постановка классической задачи нечеткого поиска [8] заключается в следующем: пусть дана строковая последовательность $P = p_1, p_2, ..., p_n$ длиной n, $|P| = n$, являющая образцом (или паттерном). Пусть имеется текст $T = t_1, t_2, ..., t_m$ длиной m, $|T| = m$, причем $n \le m$. $d(P, T_{i..j})$ – функция расстояния между образцом P и подстрокой $T_{i..j}$. Пусть $k \ge 0$ – количество допустимых ошибок. Задача формулируется следующим образом: найти все подстроки $T_{i..j}$, такие что $d(P, T_{i..j}) \le k$. Данная задача является частью общей задачи выявления пользователей ВСС, осуществляющих противоправные и потенциально противоправные действия в области рассматриваемых угроз [9].

Под функцией расстояния понимается редакционное расстояние (или расстояние редактирования) [10, 11] – минимальное количество операций редактирования, необходимых для преобразования одной строки в другую. Операции редактирования включают в

себя следующие преобразования:

- вставка символа в строку;
- удаление символа из строки;
- транспозиция символов, подстрок (в задачах обработки текста чаще всего используется операция транспозиции соседних символов);
- замена символа строки на другой символ.

Существует несколько расстояний редактирования в зависимости от типа разрешенных операций и возможных ограничений. Например, расстояние Хэмминга применяется для строк равной длины и допускает только операции замены символа. Поскольку объект исследования – тексты сообщений пользователей ВСС, то предлагается использовать в качестве функции расстояния расстояние Дамерау-Левенштейна, как наиболее подходящее для обработки текста на естественном языке. Данное расстояние разрешает все четыре типа операций редактирования. Согласно Дамерау [12] 80% всех ошибок представляют собой единичную ошибку, принадлежащую к одному из четырех классов ошибок, связанных с вышеперечисленными операциями.

Алгоритмы нечеткого поиска подразделяются на два класса: поиск онлайн и поиск оффлайн [8]. Поиск оффлайн предполагает предварительную индексацию текста и дальнейший поиск по построенному индексу. Онлайн алгоритмы нечеткого поиска не предполагают предварительной индексации. Как правило, индексация проводится для текстов большого размера, сопоставимого с небольшим корпусом, размер же сообщений в ВСС позволяет применять онлайн поиск. Для онлайн алгоритмов нечеткого поиска можно выделить четыре основные группы в зависимости от лежащих в их основе методов:

- алгоритмы на основе динамического программирования;
- алгоритмы с использованием конечных автоматов;
- алгоритмы на основе параллельных вычислений;
- алгоритмы с фильтрацией.

По умолчанию многие алгоритмы ориентированы на использование расстояния Левенштейна [10, 11] с единичным весом для операций. Данное расстояние разрешает операции вставки, удаления и замены символа. Таким образом, следует учитывать возможности адаптирования того или иного алгоритма к расстоянию Дамерау-Левенштейна. Одинаковый единичный вес для операций редактирования позволяет определять количество, но не качество ошибок в слове. С точки зрения обработки естественного языка ошибки в некоторых буквах имеют больший вес, поскольку являются часто встречающимися. Таким образом, предлагается применять взвешенное расстояние Дамерау-Левенштейна.

Исходя из выбранной функции расстояния, можно сделать вывод о неприменимости алгоритмов с использованием конечных автоматов и алгоритмов на основе параллельных вычислений для решения поставленной задачи. Детерминированный конечный автомат (ДКА) для распознавания слова с k ошибками сложен для реализации даже расстояния Левенштейна. Сначала строится недетерминированный конечный автомат (НКА), который может быть построен путем создания $k+1$ копий ДКА для точного распознавания паттерна и последующего соединения состояний копий переходами, означающими разрешенные операции редактирования [13]. Как правило, детерминизация НКА не производится, поскольку число состояний возрастает экспоненциально, вместо этого реализуют имитацию ДКА через распараллеливание операций. Кроме того, алгоритмы на автоматах предполагают использование дискретных весов для операций редактирования, при появлении ошибки в процессе распознавания в зависимости от значения веса происходит перемещение на один или несколько уровней автомата. Алгоритмы на основе параллельных вычислений предназначены либо для имитации ДКА, либо для распараллеливания вычислений алгоритмов на основе динамического программирования, кроме того, они ориентированы на использование расстояния Левенштейна с единичным весом операций.

Идея алгоритмов динамического программирования заключается в вычислении элементов матрицы D размера $(n+1)\times(m+1)$, где n – длина паттерна, а m – длина текста, на основе реккурентных соотношений [14].

$$D_{i,j}=\begin{cases}0, i=0, j=0,\\ i, j=0,\\ 0, i=0,\\ D_{i-1,j-1}, x_i=y_j\\ \min\left(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\right)+1, x_i\neq y_i\end{cases} \quad (1)$$

Как правило, при этом используются единичные веса и три допустимые операции редактирования – вставка, удаление и замена буквы. Каждое значение в последней строке матрицы, не превосходящее заданное число k, указывает на позицию в тексте, в которой

заканчивается подстрока, имеющая не больше k отличий от образца. Временная сложность алгоритма – $O(nm)$, чем больше размер текста, тем более затратным является алгоритм. Однако метод динамического программирования является самым гибким для адаптирования его к расстоянию Дамерау-Левенштейна.

Основная идея алгоритмов на основе фильтрации заключается в том, что совпадения в тексте встречаются реже, чем несовпадения, поэтому целесообразно отфильтровать те фрагменты текста, которые точно не содержат совпадений. Нечеткий текстовый поиск в данном случае состоит из двух этапов – фаза сканирования (или фильтрация) и фаза верификации. Один из подходов реализации фазы сканирования заключается в следующем: паттерн делится на $k+1$ отрезков, исходя из предположения, что как минимум один из отрезков может быть найден в тексте без ошибок [15]. Подстроки паттерна ищутся в тексте с помощью алгоритма Ахо-Корасика. Для каждого совпадения реализуется этап верификации, для которого теоретически подходит любой алгоритм, но чаще всего используется динамическое программирование. Данный этап позволяет выяснить, действительно ли была найдена подстрока текста, находящаяся на расстоянии k от образца. Временная сложность фазы сканирования составляет $O(m)$ (однако есть сильная зависимость от количества ошибок), несмотря на то, что верификация осуществляется методом динамического программирования, в целом алгоритм является одним из самых быстрых.

Таким образом, выбор метода динамического программирования в качестве центрального алгоритма для задачи нечеткого поиска по ключевым словам в текстах сообщений пользователей ВСС является обоснованным. Взвешенное расстояние Дамерау-Левенштейна позволяет учитывать наиболее частотные одиночные ошибки при нечетком текстовом поиске, например, пропуск буквы «н» после «н» (сдвоенные согласные). Однако более сложные ошибки или некоторые языковые игры, при которых намеренно искажаются слова, необходимо обрабатывать другими способами, поскольку в таких случаях расстояние Дамерау-Левенштейна будет большим, и нужное слово, возможно, не попадет в поисковую выдачу. Примером может служить слово «пофтыкав» (слово взято из корпуса сообщений пользователей ВСС). Интуитивно понятно, что «пофтыкав» гораздо ближе к «повтыкав», чем «коза» и «роза», хотя количество ошибок в первом случае больше, чем во втором. Рассмотрим подходы к обработке ошибок из области автоматической коррекции ошибок.

2 Методы коррекции ошибок

Существует множество подходов к исправлению ошибок, в целом можно выделить три группы (некоторые техники могут комбинировать подходы):

- методы на основе правил;
- методы на основе генерации ключей;
- статистические методы.

По признаку учета контекста можно выделить методы коррекции изолированного термина и методы коррекции с учетом контекста.

Как правило, большинство техник исправления опечаток состоят из двух этапов – определение списка кандидатов на коррекцию ошибочно написанного слова и выбор лучшего кандидата путем ранжирования различными методами.

Примером подхода на основе правил являются работы [16, 17]. В первой работе метод основан на строгих морфологических правилах языка (рассматривается баскский язык): морфологический анализатор разбивает искомое слово на морфемы, которые уже по отдельности проверяются на наличие ошибок. Исправлению подвергаются только те морфемы, которые содержат ошибку. На заключительном этапе морфемы собираются в слово. В [17] правила (шаблоны грамматических ошибок) выявляются на основе анализа корпуса текстов. В зависимости от частоты встречаемости каждому шаблону присваивается вес. Процесс коррекции заключается в следующем: если искомое слово не найдено в словаре, но существуют кандидаты на исправление с высокой степенью близости, и сегмент слова, содержащего ошибку, подчиняется разработанным правилам, то детектируется ошибка. После применения правил к рассматриваемому слову, оценивается вес изменений, и кандидаты на коррекцию ранжируются с использованием правила Байеса. Системы на основе правил не получили широкого распространения, поскольку разработка правил, учитывающие все особенности правописания языка является очень трудоемкой. Однако элементы такого подхода используются для разработки фонетических моделей ошибок, которые будут рассмотрены ниже.

Методы на основе генерации ключей предполагают кодирование слов по определенным правилам, например, удаление из слова малозначащих букв, замена двух согласных одной (в случае удвоения согласных). Такие правила должны быть устойчивыми к

большинству ошибок. Примером данного подхода является работа [18]. Поиск кандидатов на коррекцию осуществляется путем сравнения ключей, сгенерированных для слов из словаря, с ключом, сгенерированным для ошибочно написанного слова. К отобранным кандидатам применяются все возможные операции вставки, удаления, замены и транспозиции в попытке получить первоначальное слово с ошибкой. К методам на основе генерации ключей можно отнести фонетические алгоритмы, например, Soundex. Хотя они не являются методами коррекции ошибок в полном смысле, а только определяют фонетическую близость между словами, данная категория алгоритмов используется в виде составляющей компоненты в системах проверки правописания. Основой фонетических алгоритмов является разбиение букв на группы, исходя из степени схожести их звучания (например, оригинальный вариант Soundex) или разбиение по сочетаниям букв, исходя из фонетических правил языка (например, Metaphone). Несмотря на то, что фонетические алгоритмы учитывают особенности произношения, что является важным для поиска в текстах неофициального стиля, не следует использовать их без дополнительных алгоритмов, поскольку велико число ложноположительных совпадений. Методы на основе генерации ключей (за исключением фонетических алгоритмов) также не получили широкого распространения.

Среди статистических подходов наиболее важным является коррекция ошибок на основе модели зашумленного канала [19]. Данный метод используется как одна из составляющих коррекции ошибок в современных системах (например, поисковых системах Яндекс [20], Rambler [21]) и научных исследованиях [22]. Эта модель, заимствованная из области распознавания речи, предполагает, что в результате прохождения зашумленного канала исходное слово может искажаться, превращаясь в слово, написанное с ошибками. В контексте письменной речи зашумленный канал – это человек, печатающий текст. Требуется восстановить исходное слово, для этого используется формула Байеса. Пусть x – ошибочно написанное слово. Из множества W возможных кандидатов на коррекцию необходимо выбрать лучший вариант, такой, что

$$\hat{w} = \arg\max_{w \in W} P(w \mid x) = \arg\max_{w \in W} \frac{P(x \mid w)P(w)}{P(x)}. \quad (2)$$

Таким образом, для определения лучшего слова необходимо максимизировать значение $P(x \mid w)P(w)$, где $P(x \mid w)$ – модель зашумленного канала (или модель ошибок), представляющая собой вероятность того, что слово w может быть искажено в x, а $P(w)$ – языковая модель (словарная модель вероятности). $P(x \mid w) = \prod_{i=1}^{n} p(x_i \mid w_i)$, где $|w| = n$, а пары x_i, w_i являются сегментами выравнивания между двумя словами, т.е. определяется какой символ одного слова является соответствием (отображением) символа другого слова. При этом w_i может соответствовать пустому символу ε, что является ошибкой удаления, и наоборот x_i может соответствовать ε, что является ошибкой вставки. Для исправления ошибок сначала выбираются кандидаты на коррекцию, находящиеся на расстоянии один или два от ошибочно написанного слова. Следует отметить, что на этапе выбора кандидатов не используется взвешенное расстояние. Затем кандидаты ранжируются, исходя из полученных значений по вышеописанной формуле. Условные вероятности для модели ошибок определяются либо на основе корпуса пар вида ошибка-исправление, либо путем автоматической обработки корпуса с последующим избавлением от шума (как правило, на основе языковой модели). Ключевыми улучшениями этого метода является использование модели ошибок не на основе букв, как в [19], а на основе подстрок [23], а также интеграция фонетической модели [24]. Рассмотрим последний метод подробнее.

Для формирования фонетической модели определяются вероятности различных вариантов произношения той или иной буквы и формулируются правила произношения с учетом контекста, т.е. расположенных рядом букв. Кандидаты на коррекцию слова переводятся в код согласно фонетической модели и затем сравниваются с кодом ошибочно написанного слова. Модель ошибок состоит из двух компонентов – модель ошибок на основе подстрок (как в [23]) и фонетическая модель ошибок, при этом модели являются независимыми друг от друга. Полученные значения от каждой компоненты модели ошибок суммируются и передаются в формулу Байеса для ранжирования.

Следует отметить, что учет контекста во многих методах осуществляется за счет использования биграмм и триграмм в качестве языковой модели.

В задаче нечеткого поиска в текстах сообщений пользователей ВСС нет потребности в исправлении ошибок, но необходим дополнительный инструмент, с помощью которого степень уверенности в релевантности

найденных результатов повышалась бы или понижалась. Наибольший интерес представляет модель ошибок из модели зашумленного канала и подход на основе правил.

3 Предлагаемая модель ошибок и интеграция ее в нечеткий поиск

Предлагается разбивать текст на подстроки, соответствующие словам, ошибки сегментации обрабатываются отдельно. Целью данной статьи является описание модели ошибок и ее интеграция в нечеткий поиск, поэтому рассмотрим только те этапы поискового алгоритма, которые непосредственно связаны с моделью ошибок (т.е. этап верификации). Предлагается осуществлять нечеткий текстовый поиск не на уровне текста (1), а на уровне выделенных подстрок, т.е. выявлять редакционные расстояния между словами сообщения и словарем ключевых слов.

Пусть даны две строки x и y длиной n и m соответственно, матрица D размером $(n+1)\times(m+1)$, тогда расстояние между строками $d(x,y)=D(n,m)$, при этом $i=0..n$, $j=0..m$. Модифицируем формулу (1), введем следующие обозначения: пусть $\delta(x_i,y_j)$ – стоимость (вес) замены x_i на y_j, причем, если $x_i=y_j$, то $\delta(x_i,y_j)=0$. $\delta(x_i,\varepsilon)$ – стоимость удаления x_i, $\delta(\varepsilon,y_j)$ – стоимость вставки y_j в строку x. $\delta_t(x_{i-1}x_i,x_ix_{i-1})$ – стоимость транспозиции двух смежных букв (замена $x_{i-1}x_i$ на x_ix_{i-1}), при условии, что $x_{i-1}x_i=y_jy_{j-1}$. Изменим начальные условия при $i=0$ и $j=0$, поскольку осуществляется поиск не паттерна в тексте, а вычисляется расстояние редактирования между словами. Тогда формула (1) примет вид:

$$D_{i,j}=\begin{cases}0, i=0, j=0,\\ D_{i-1,j}+\delta(x_i,\varepsilon), j=0,\\ D_{i,j-1}+\delta(\varepsilon,y_j), i=0,\\ \min(D_{i-1,j}+\delta(x_i,\varepsilon),\\ D_{i,j-1}+\delta(\varepsilon,y_j), D_{i-1,j-1}+\delta(x_i,y_j)),\\ \min(D_{i-1,j}+\delta(x_i,\varepsilon), D_{i,j-1}+\delta(\varepsilon,y_j), D_{i-1,j-1}+\\ \delta(x_i,y_j), D_{i-2,j-2}+\delta_t(x_{i-1}x_i,x_ix_{i-1}))\end{cases} \quad (3)$$

По своей сути модель ошибок из модели зашумленного канала (2) представляет собой совокупность весов, которые можно использовать в нечетком текстовом поиске при вычислении расстояния Дамерау-Левенштейна. Пусть $A=\{а,б,...,я\}$ – алфавит языка, при этом будем учитывать букву «ё» как «е», ь, ъ $\notin A$.

$Q_{ins}=\{p_i(wz\mid w)\mid i=\overline{1..1024}, w\in A, z\in A\}$ – множество вероятностей вставки z после w.

$Q_{del}=\{p_i(w\mid wz)\mid i=\overline{1..1024}, w\in A, z\in A\}$ – множество вероятностей удаления z после w.

$Q_{sub}=\{p_i(w\mid z)\mid i=\overline{1..1024}, w\in A, z\in A\}$ – множество вероятностей замены z на w.

$Q_{trans}=\{p_i(zw\mid wz)\mid i=\overline{1..1024}, w\in A, z\in A\}$ – множество вероятностей перестановки wz. Тогда статистическая модель ошибок примет вид:

$$Q=Q_{ins}\cup Q_{del}\cup Q_{sub}\cup Q_{trans} \quad (4)$$

Условные вероятности оцениваются на основе обучающего текстового корпуса, во избежание нулевых вероятностей в связи с неполнотой любых обучающих данных традиционно используется сглаживание, в данном случае сглаживание Лапласа:

$$p(wz\mid w)=\frac{Ins(w,z)+1}{count(w)+1024}, \quad (5)$$

где $Ins(w,z)$ – элемент матрицы Ins, значение которого определяет количество ошибочных вставок z после w, $count(w)$ – количество раз, когда w встретилось в тексте.

$$p(w\mid wz)=\frac{Del(w,z)+1}{count(wz)+1024}, \quad (6)$$

где $Del(w,z)$ – элемент матрицы Del, значение которого определяет количество ошибочных удалений z после w, $count(wz)$ – количество раз, когда wz совместно встретились в тексте.

$$p(w\mid z)=\frac{Sub(w,z)+1}{count(z)+1024}, \quad (7)$$

где $Sub(w,z)$ – элемент матрицы Sub, значение которого определяет количество замен z на w, $count(z)$ – количество раз, когда z встретилось в тексте.

$$p(zw\mid wz)=\frac{Trans(w,z)+1}{count(wz)+1024}, \quad (8)$$

где $Trans(w,z)$ – элемент матрицы $Trans$, значение которого определяет количество перестановок wz как zw, $count(wz)$ – количество раз, когда wz совместно встретились в тексте.

Введем фонетическую модель ошибок. Кодировать каждую букву в фонему с учетом контекста представляется нецелесообразным, данный подход является очень трудоемким и избыточным для задачи нечеткого поиска. Предлагается обрабатывать наиболее типичные фонетические ошибки, например, оглушение согласных в слабой позиции (например, д-т в слове «код»), озвончение согласных (с-з в слове «просьба»), гласные после шипящих («шепот»-«шопот»). При этом переход непосредственно к

фонемам не осуществляется, например, буква «я» в слове «яблоко» не преобразуется в фонему «йа». Осуществляется замена последовательности букв, как правило, длиной от единицы до четырех, на последовательность букв, схожую по звучанию. Таким образом, фонетическая модель представляет собой множество правил $R=\{r_1,r_2,...,r_n\}$, где n – количество правил. Каждое правило можно представить в следующем виде: $r_i=ab\to f, a\in A^*, b\in A, f\in A^+$.

Предлагается ввести дополнительное множество для учета клавиатурной близости букв, при этом для рассматриваемой буквы учитываются только смежные буквы в этом же ряду:

$$K=\{k_1,k_2,...,k_{32}\},$$

$$k_i=\langle s,\langle a,d\rangle\rangle, s\in A, a\in A, d\in A\vee d=\varepsilon .$$

Возможное равенство $d=\varepsilon$ позволяет формировать наборы для букв, расположенных на краях клавиатуры.

Таким образом, модель ошибок представляется в виде совокупности трех компонентов. Буквенная модель ошибок Q (4) применяется на первой итерации нечеткого поиска методом динамического программирования, при этом условные вероятности используются в качестве весов для вычисления расстояния Дамерау-Левенштейна (3) следующим образом:

$$\delta(x_i,y_j)=-\log p(y_j\mid x_i),\ \delta(x_i,\varepsilon)=-\log p(x_{i-1}\mid x_{i-1}x_i),$$

$$\delta(\varepsilon,y_j)=-\log p(x_iy_j\mid x_i),$$

$\delta_t(x_{i-1}x_i,x_ix_{i-1})=-\log p(x_ix_{i-1}\mid x_{i-1}x_i)$. Если при этом был достигнут минимальный порог сходства $k_{\min}$, то к рассматриваемым словам применяются правила из фонетической модели ошибок и осуществляется вторая итерация поиска с целью повышения порога сходства до требуемого k. Кроме того, выполняется выравнивание между словами, выявленные ошибки проверяются на принадлежность клавиатурным заменам для соответствующих букв. Если в результате второй итерации поиска достигается требуемый уровень k, то слово считается найденным.

4 Оценка параметров статистической компоненты модели ошибок

Для выявления распределения ошибок по типам операций редактирования и оценки параметров (5-8) компоненты Q (4) модели ошибок было проведено исследование корпуса текстовых сообщений в размере 45000 слов, являющегося подвыборкой Генерального интернет-корпуса русского языка [25], содержащего тексты из различных социальных медиа, в том числе Вконтакте. Ошибки в данной подвыборке были исправлены асессорами для проведения соревнования по автоматическому исправлению опечаток SpellRuEval. Таким образом, обучающие данные представляют собой два корпуса – один зашумленный, т. е. содержащий ошибки, другой – исправленный. Для каждой пары слов из корпусов было выполнено выравнивание с использованием расстояния Дамерау-Левенштейна с одинаковым весом для всех операций и букв. Использование одинаковых весов обусловлено текущей задачей – выявление ошибок, их типов, количества и позиций в словах для формирования матриц ошибок (5-8).

Пример матрицы ошибок представлен в таблице 1. Из таблицы 1 видно, что количество ошибочных замен «о» на «а» (262) более, чем в 2 раза превышает количество замен «а» на «о» (115), поэтому замена «о» на «а» при расчете расстояния Дамерау-Левенштейна будет стоить в 2 раза дешевле, чем «а» на «о». Матрицы ошибок включают в себя сведения о некоторых фонетических правилах и клавиатурной близости, например, частотная замена «ч» на «ш», «к» на «е». Однако введение отдельной фонетической компоненты ошибок необходимо, она позволит обрабатывать более сложные ошибки такие, как «тся» – «ца». Учет подобных замен в матрице ошибок возможен, но является очень трудоемкой и ресурсоемкой задачей, для решения которой потребуется корпус размером на несколько порядков больше. Для наиболее частотных ошибок, например, «в общем» – «вообщем» – «вопщим» предлагается сформировать отдельный список слов. Выделение клавиатурных замен в явном виде позволит повысить или понизить значение функции расстояния между словами при нечетком поиске.

Таблица 1 – Матрица ошибок $Sub(w,z)$ для операции замены

	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
а		6		6	4	8		2	6					7	262	2	1	2	6					1				3	1			22
б					2									1					2									1				
в	2			8	3					1	2	1	2	2		1		4		1	1										1	1

г							1				13		1	4	1				2												1
д	1	2	1	1		1	2					5	1	3		1			3				1						1			1
е	20		1		1			1	202	3	4	1		3	45	1	1		4	3		1	1	1				1	1	18	1	27
ж				1	1			2	1			1										1			1	3		1		1		5
з			2			2							1					16				1			1				7			1
и	8		1	1	1	151	1			8	3	7	3	5	6		2	1	5	2				2				8	8			5
й					1	12			11				1	1				1				2										2
к		1	1	6	1	3	2		3			1		3	3	2			2	1		1						1				2
л		1			1	1	1		1				1	6	4	1		1	4					1								
м			1					1	4	3		1			1	3	2	1	5			1						1				1
н	5	3	3		2	1		1	2	1	1	2	2					2	8										1	2		
о	115		2			22		1	9		3	3		2		2	2	3	5	4				1				2	5		1	11
п	3	6				1					2	2	1				1	2	3												
р	2	1	1					18			1		1		6	2			1				1							1	
с		1		2	1	6		18	2			2	2	1	1	1				1		1	1	1	1						1
т	7	1	3		23	3			7		5	2	2	7	2	2	1	1		1		1		3		1			1		
у	1					6			4	1				1	4		1		1									4			3	
ф			27												1																
х				2	2	1		1		1	4				1			1						2			1				
ц					4			1										37	10				7	5				1	2		
ч								2			1		1	1				5					1			11		1		1		
ш						2	1		2					1	1			2	4			1		40		35						
щ							4	1				1	1	1	1			1					1	23	9							
ъ																						1							1			
ы	1						1		25					2	13				1	1											2
ь					1	4	1		5	1				1	4			1	4			1					9					1
э			1			8	1				1																2	2				
ю											1								2	1								1		2		2
я	8				1	11			4			1			4				1	1							2	4		2		

Таблица 2 – Оценка вероятностей $p(w \mid z)*10^{-4}$ *для операции замены, буквы «а»-«о»*

	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
а	0.44	12.61	0.93	13.44	5.4	3.65	2.64	5.64	3.91	2.5	0.98	0.82	0.99	4.15	83.99
б	0.44	1.8	0.93	1.92	3.24	0.41	2.64	1.88	0.56	2.5	0.98	0.82	0.99	1.04	0.32
в	1.32	1.8	0.93	17.28	4.32	0.41	2.64	1.88	0.56	5.0	2.93	1.64	2.97	1.55	0.32
г	0.44	1.8	0.93	1.92	1.08	0.41	5.27	1.88	0.56	2.5	13.66	0.82	1.98	2.59	0.64
д	0.88	5.4	1.85	3.84	1.08	0.81	7.91	1.88	0.56	2.5	0.98	4.92	1.98	2.07	0.32
е	9.22	1.8	1.85	1.92	2.16	0.41	2.64	3.76	113.5	10.0	4.88	1.64	0.99	2.07	14.69
ж	0.44	1.8	0.93	3.84	2.16	0.41	2.64	5.64	1.12	2.5	0.98	1.64	0.99	0.52	0.32
з	0.44	1.8	2.78	1.92	1.08	1.22	2.64	1.88	0.56	2.5	0.98	0.82	1.98	0.52	0.32
и	0.44	3.6	1.85	3.84	2.16	61.57	5.27	1.88	0.56	22.51	3.9	6.56	3.96	3.11	2.24
й	0.44	1.8	0.93	1.92	2.16	5.27	2.64	1.88	6.71	2.5	0.98	0.82	1.98	1.04	0.32
к	0.44	3.6	1.85	13.44	2.16	1.62	7.91	1.88	2.24	2.5	0.98	1.64	0.99	2.07	1.28
л	0.44	3.6	0.93	1.92	2.16	0.81	5.27	1.88	1.12	2.5	0.98	0.82	1.98	3.63	1.6
м	0.44	1.8	1.85	1.92	1.08	0.41	2.64	3.76	2.8	10.0	0.98	1.64	0.99	0.52	0.64
н	2.63	7.2	3.71	1.92	3.24	0.81	2.64	3.76	1.68	5.0	1.95	2.46	2.97	0.52	0.32
о	50.92	1.8	2.78	1.92	1.08	9.32	2.64	3.76	5.59	2.5	3.9	3.28	0.99	1.55	0.32
п	1.76	12.61	0.93	1.92	1.08	0.81	2.64	1.88	0.56	2.5	2.93	2.46	1.98	0.52	0.32
р	1.32	3.6	1.85	1.92	1.08	0.41	2.64	3.76	0.56	2.5	1.95	0.82	1.98	0.52	2.24
с	0.44	3.6	0.93	5.76	2.16	2.84	2.64	35.7	1.68	2.5	0.98	2.46	2.97	1.04	0.64
т	3.51	3.6	3.71	1.92	25.92	1.62	2.64	1.88	4.47	2.5	5.86	2.46	2.97	4.15	0.96

у	0.88	1.8	0.93	1.92	1.08	0.81	2.64	1.88	2.8	5.0	1.95	0.82	0.99	1.04	1.6
ф	0.44	1.8	25.96	1.92	1.08	0.41	2.64	1.88	0.56	2.5	0.98	0.82	0.99	0.52	0.64
х	0.44	1.8	0.93	5.76	3.24	0.81	2.64	3.76	0.56	5.0	4.88	0.82	0.99	0.52	0.64
ц	0.44	1.8	0.93	1.92	5.4	0.41	2.64	3.76	0.56	2.5	0.98	0.82	0.99	0.52	0.32
ч	0.44	1.8	0.93	1.92	1.08	0.41	2.64	5.64	0.56	2.5	1.95	0.82	1.98	1.04	0.32
ш	0.44	1.8	0.93	1.92	1.08	1.22	5.27	1.88	1.68	2.5	0.98	0.82	0.99	1.04	0.64
щ	0.44	1.8	0.93	1.92	1.08	1.62	13.18	3.76	0.56	2.5	0.98	1.64	1.98	1.04	0.64
ъ	0.44	1.8	0.93	1.92	1.08	0.41	2.64	1.88	0.56	2.5	0.98	0.82	0.99	0.52	0.32
ы	0.88	1.8	0.93	1.92	1.08	0.41	5.27	1.88	14.54	2.5	0.98	0.82	0.99	1.55	4.47
ь	0.44	1.8	0.93	1.92	2.16	2.03	5.27	1.88	3.35	5.0	0.98	0.82	0.99	1.04	1.6
э	0.44	1.8	1.85	1.92	1.08	3.65	5.27	1.88	0.56	2.5	1.95	0.82	0.99	0.52	0.32
ю	0.44	1.8	0.93	1.92	1.08	0.41	2.64	1.88	0.56	2.5	1.95	0.82	0.99	0.52	0.64
я	3.95	1.8	0.93	1.92	2.16	4.86	2.64	1.88	2.8	2.5	0.98	1.64	0.99	0.52	1.6

На основе матриц ошибок и формул (5-8) была проведена оценка параметров статистической компоненты Q (4) модели ошибок. Пример оценок представлен в таблице 2.

Заключение

Данная статья посвящена проблеме поиска по ключевым словам в процессе мониторинга виртуальных социальных сетей с целью выявления действительных и потенциальных случаев пропаганды экстремизма и терроризма, суицида, пропаганды применения наркотических средств и психотропных веществ и их незаконного распространения. Сложность информационного поиска, помимо больших объемов, обрабатываемых данных заключается в наличии большого количества ошибок и сленга в текстах сообщений пользователей. В данной работе рассматриваются методы коррекции ошибок и подходы в области нечеткого поиска. Предлагается интегрировать трехкомпонентную модель ошибок в алгоритм нечеткого поиска на основе динамического программирования. Модель ошибок представляет собой совокупность фонетических правил языка, множество клавиатурных замен и совокупность весов для операций редактирования – вставки, удаления, замены буквы и транспозиции соседних букв. В статье приведена оценка весов на основе анализа корпуса текстовых сообщений пользователей. Путем применения недискретных весов динамическое программирование позволяет занижать расстояние для слова с типичной ошибкой и повышать его для редкой ошибки.

В дальнейших исследованиях предполагается решение следующих задач:

- выявление возможно более подходящих методов сглаживания для оценки параметров статистической компоненты модели ошибок;
- оценка необходимости расширения множества клавиатурных замен и назначение весов в соответствии с клавиатурной близостью;
- разработка фазы сканирования текста, комбинирование ее с алгоритмом динамического программирования с интегрированной моделью ошибок и последующее тестирование общего алгоритма.

Литература

1. Социальные сети в России, лето 2017: цифры и тренды [электронный ресурс] // URL: http://blog.br-analytics.ru/sotsialnye-seti-v-rossii-leto-2017-tsifry-i-trendy/ (дата обращения 22.08.2017).
2. Давыдова Ю.В. К вопросу о необходимости автоматизации поиска угроз в виртуальных социальных сетях и сообществах // Актуальные проблемы науки в 21 веке: материалы 6-й Международной научно-практической конференции (Махачкала, 28 дек. 2014 г.). – Махачкала, 2014. – С. 25-26.
3. Базенков Н.И., Губанов Д.А. Обзор информационных систем анализа социальных сетей // Управление большими системами. – 2013. – Вып. 41. – С. 357-394.
4. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Физматлит, 2010. – 228 с.
5. Машечкин И.В., Петровский М.И., Поспелова И.И., Царев Д.В. Методы автоматического аннотирования и выделения ключевых слов в задачах обнаружения экстремистской информации в сети Интернет // Современные информационные технологии и ИТ-образование. – 2016. – Т. 12, № 1. – С. 188-198.
6. Савва Ю.Б., Еременко В.Т., Давыдова Ю.В. О проблеме лингвистического анализа сленга в задаче автоматизированного поиска угроз распространения наркомании в виртуальных социальных сетях // Информационные системы и технологии. – 2015. – Т. 6, № 92. – С. 68-75.
7. Sykora M.D., Jackson T.W., O'Brien A., Elayan S. National security and social media monitoring: a presentation of the EMOTIVE and related systems // Proceedings of European Intelligence and Security Informatics Conference. – 2013. – pp. 172-175.

8. Navarro G. A guided tour to approximate string matching // ACM Surveys. – 2001. – Vol. 33, No. 1. – pp. 33-88.
9. Savva Yu.B., Davydova Yu.V. Modeling of online social networks for automated monitoring system // Proceedings of the III International conference on Information Technology and Nanotechnology. Session Data Science. – 2017. URL: http://ceur-ws.org/Vol-1903/paper16.pdf
10. Manning C.D., Raghavan P., Schutze H. Introduction to information retrieval. – Cambridge: Cambridge University Press, 2008. – 496 p.
11. Ingersoll G.S., Morton T.S., Farris L.A. Taming text. How to find, organize and manipulate it. – NY: Manning Publications Co., 2013. – 320 p.
12. Damerau F.J. A technique for computer detection and correction of spelling errors // Communications of the ACM. – 1964. – Vol. 7, No. 3. – pp. 171-176.
13. Holub J. The finite automata approaches in stringology // Kybernetika. – 2012. Vol. 48, No. 3. – pp. 386-401.
14. Smyth B. Computing patterns in strings. – Pearson Addison-Wesley, 2003. – 423 p.
15. Baeza-Yates R.A., Perleberg C.H. Fast and practical approximate string matching // Information Processing Letters. – 1996. – Vol. 59. – pp. 21-27.
16. Aduriz I. et al. Spelling corrector for Basque based on morphology // Literary and Linguistic Computing. – 1997. – Vol. 12, No. 1. – pp. 31-3.
17. Yannakoudakis E., Fawthrop D. An intelligent spelling corrector // Information processing and management. – 1983. – Vol. 19, No. 2. – pp. 101-108.
18. Pollock J.J., Zamora A. Automatic spelling correction in scientific and scholarly texts // Communications of the ACM. – 1984. – Vol. 27, No. 4. – pp. 358-368.
19. Kernighan M.D., Church K.W., Gale W.A. A spelling correction program based on a noisy channel model // Proceedings of COLING. – 1990. – pp. 205-210.
20. Панина М.Ф., Байтин А.В., Галинская И.Е. Автоматическое исправление опечаток в поисковых запросах без учета контекста // Компьютерная лингвистика и интеллектуальные технологии: материалы ежегодной Международной конференции «Диалог» (Бекасово, 29 мая – 2 июня 2013 г.). – Бекасово, 2013. – С. 568-579.
21. Карпенко М.П., Протасов С.В. Некоторые методы очистки словаря запросов поиска // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Международной конференции «Диалог» (Бекасово, 25 – 29 мая 2011 г.). – Бекасово, 2011. – С. 280-291.
22. Sorokin A.A., Shavrina T.O. Automatic spelling correction for Russian social media texts // Proceedings of the International Conference “Dialog” (Moscow, June 1 – 6, 2016). – Moscow, 2016. – pp.688-701.
23. Brill E., Moore R.C. An improved error model for noisy channel spelling correction // Proceedings of the 38th Annual Meeting on the Association for Computational Linguistics. – 2000. – pp. 286-293.
24. Toutanova K., Moore R.C. Pronunciation modeling for improved spelling correction // Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. – 2002, pp. 144-151.
25. Генеральный интернет-корпус русского языка [электронный ресурс] // URL: http://www.webcorpora.ru/ (дата обращения 04.09.2017).

References

1. Socyal’nye seti v Rossii, leto 2017: cyfry i trendy [электронный ресурс] // URL: http://blog.br-analytics.ru/sotsialnye-seti-v-rossii-leto-2017-tsifry-i-trendy/ (дата обращения 22.08.2017).
2. Davydova Yu.V. K voprosu o neobhodimosti avtomatizacii ugroz v virtual’nyh social’nyh setyah i soobshhestvah // Aktual’nye problemy nauki v 21 veke: materialy 6-j Mezhdunarodnoj nauchno-prakticheskoj konferencii (Mahachkala, 28 dek. 2014 g.). – Mahachkala, 2014. – S. 25-26
3. Bazenkov N.I., Gubanov D.A. Obzor informacyonnyh system analiza social’nyh setej // Upravlenie bol’shhimi sistemami. – 2013. – Vyp. 41. – S. 357-394.
4. Gubanov D.A., Novikov D.A., Chhartishvili A.G. Social’nye seti: modeli informacyonnogo vliyaniya, upravleniya i protivoborstva. – M.: Fizmatlit, 2010. – 228 c.
5. Mashechkin I.V., Petrovskij M.I., Pospelova I.I., Tsarev D.V. Metody avtomaticheskogo annotirovaniya i vydeleniya klyuchevyh slov v zadachah obnaruzheniya jekstremistskoj informacii v seti Internet // Sovremennye informacyonnye tehnologii i IT-obrazovanie. – 2016. – T. 12, № 1. – S. 188-198.
6. Savva Yu.B., Eremenko V.T., Davydova Yu.V. O probleme lingvisticheskogo analiza slenga v zadache avtomatizirovannogo poiska urgoz rasprostraneniya narkomanii v virtual’nyh social’nyh setyah // Informatcyonnye sistemy i tehnologii. – 2015. – T.6, №. 92, – S. 68-75.
7. Sykora M.D., Jackson T.W., O’Brien A., Elayan S. National security and social media monitoring: a presentation of the EMOTIVE and related systems // Proceedings of European Intelligence and Security Informatics Conference. – 2013. – pp. 172-175
8. Navarro G. A guided tour to approximate string matching // ACM Surveys. – 2001. – Vol. 33, No. 1. – pp. 33-88.
9. Savva Yu.B., Davydova Yu.V. Modeling of online social networks for automated monitoring system // Proceedings of the III International conference on Information Technology and Nanotechnology. Session Data Science. – 2017. URL: http://ceur-ws.org/Vol-1903/paper16.pdf
10. Manning C.D., Raghavan P., Schutze H. Introduction to information retrieval. – Cambridge: Cambridge University Press, 2008. – 496 p.
11. Ingersoll G.S., Morton T.S., Farris L.A. Taming text. How to find, organize and manipulate it. – NY: Manning Publications Co., 2013. – 320 p.
12. Damerau F.J. A technique for computer detection and correction of spelling errors // Communications of the ACM. – 1964. – Vol. 7, No. 3. – pp. 171-176.
13. Holub J. The finite automata approaches in stringology // Kybernetika. – 2012. Vol. 48, No. 3. – pp. 386-401.
14. Smyth B. Computing patterns in strings. – Pearson Addison-Wesley, 2003. – 423 p.
15. Baeza-Yates R.A., Perleberg C.H. Fast and practical approximate string matching // Information Processing Letters. – 1996. – Vol. 59. – pp. 21-27.
16. Aduriz I. et al. Spelling corrector for Basque based on morphology // Literary and Linguistic Computing. – 1997. – Vol. 12, No. 1. – pp. 31-3.
17. Yannakoudakis E., Fawthrop D. An intelligent spelling corrector // Information processing and management. – 1983. – Vol. 19, No. 2. – pp. 101-108.
18. Pollock J.J., Zamora A. Automatic spelling correction in scientific and scholarly texts // Communications of the ACM. – 1984. – Vol. 27,

No. 4. – pp. 358-368.
19. Kernighan M.D., Church K.W., Gale W.A. A spelling correction program based on a noisy channel model // Proceedings of COLING. – 1990. – pp. 205-210.
20. Panina M.F., Bajtin A.V., Galinskaya I.E. Avtomaticheskoe ispravlenie opechatok v poiskovyh zaprosah bez ucheta konteksta // Komp'yuternaya lingvistika i intellektual'nye tehnologii: materialy ezhegodnoj Mezhdunarodnoj konferencii "Dialog" (Bekasovo, 29 majya – 2 iyunya 2013 g.). – Bekasovo, 2013. – S. 568-579.
21. Karpenko M.P., Protasov S. V. Nekotorye metody ochistki slovarya zaprosov poiska // Komp'yuternaya lingvistika i intellektual'nye tehnologii: materialy ezhegodnoj Mezhdunarodnoj konferencii "Dialog" (Bekasovo, 25 – 29 majya 2011 g.). – Bekasovo, 2011. – S. 280-291.
22. Sorokin A.A., Shavrina T.O. Automatic spelling correction for Russian social media texts // Proceedings of the International Conference "Dialog" (Moscow, June 1 – 6, 2016). – Moscow, 2016. – pp.688-701.
23. Brill E., Moore R.C. An improved error model for noisy channel spelling correction // Proceedings of the 38th Annual Meeting on the Association for Computational Linguistics. – 2000. – pp. 286-293.
24. Toutanova K., Moore R.C. Pronunciation modeling for improved spelling correction // Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. – 2002, pp. 144-151.
25. General'nyj internet-korpus russkogo yazyka [электронный ресурс] // URL: http://www.webcorpora.ru/ (дата обращения 04.09.2017).

Поступила: 30.09.2017

Об авторе:
Давыдова Юлия Витальевна, ведущий программист, Орловский государственный университет имени И.С. Тургенева, j.davydova@ostu.ru

Note on the author:
Davydova Yulia V., Leading software developer, Orel State University, j.davydova@ostu.ru