МЕТОДЫ И АЛГОРИТМЫ НЕЧЁТКОЙ ПАРОЛЬНОЙ АУТЕНТИФИКАЦИИ

Дубинецкий В. В.
Отдел студенческой науки и магистратуры, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: vitali_dubinetski@tut.by

В данной статье исследована модификация методов нечёткого поиска и алгоритмов автодополнения в сочетании с парольной аутентификацией. Сделана попытка выбора параметров алгоритма на основе частоты появления ошибок разных типов при наборе текста разными группами людей. Оценена безопасность данных, защищенных с помощью данного алгоритма

Введение

С развитием науки и увеличением объема информации растет необходимость в удобном и быстром поиске. В первых информационных системах поиск был основан на полном совпадении введенного пользователем слова или строки с оригиналом (как слово «Пассажирский» в словосочетании «станция Минск-Пассажирский»). Затем появился поиск, основанный на вхождении поисковой строки в исходный текст (как «Пассажир» или «саж» в предыдущем примере). В этом случае получили широкое распространение алгоритмы, такие как алгоритм Укконена, в которых на вхождение проверяется только начало слова (по запросу «Мин» слово «Минск-Пассажирский» будет найдено, а по запросу «жир» – нет). Эти алгоритмы применяются в системах с автодополнением текста. Следующим этапом в развитии поисковых систем стало широкое распространение программных систем и средств, поиск информации в которых основан не на простом совпадении введенного пользователем слова или строки с оригиналом или его частью, а на их схожести. Ошибки и опечатки в строке уже не приводили к отсутствию найденных результатов. В таких системах вычисляется редакционное расстояние (расстояние Левенштейна [1] или его модификация [2] и др.) между поисковой строкой и строкой оригинала или просто устанавливается факт их схожести (без подсчета расстояния). Широкое применение получили метод динамического программирования Вагнера и Фишера, метод N-грамм, алгоритм Bitap и др.

Рассмотрим глубже понятие редакционного расстояния.

Расстояние Левенштейна (редакционное расстояние или дистанция редактирования) – это минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую [1].

В расстоянии Дамерау-Левенштейна (модификации расстояния Левенштейна) к операциям вставки, удаления и замены добавляется операция транспозиции (перестановки двух соседних символов) [2].

Из определений следует, что все операции равнозначны и их вес (цена) при подсчете расстояния одинаков и равен единице. Однако в общих алгоритмах нахождения редакционного расстояния они могут быть произвольными.

I. Модификация редакционного расстояния

Проанализировав основные ошибки при вводе текста, было решено расширить список операций при подсчете редакционного расстояния:

- операция вставки одного символа, ее цена w_I;
- операция удаления одного символа, ее цена w_D;
- операция замены одного символа на другой, ее цена w_R;
- операция перестановки двух соседних символов местами, ее цена w_T;
- пустая операция, или операция совпадения (символы обеих строк совпадают), ее цена w_E (обычно равна нулю и добавлена для универсальности алгоритма);
- операция локального сдвига одного символа на соседний по устройству ввода (клавиатуре), ее цена w_{LS};
- операция группового сдвига (замены последовательности из более чем N символов на соседние по устройству ввода (клавиатуре) в одном направлении), ее цена w_{GS} , цена сдвига каждого символа группы w_{GSS}, (обычно равна нулю и добавлена для универсальности алгоритма);
- операция смены языка ввода всего текста, ее цена w_{CL};
- операция смены языка ввода одного символа, ее цена w_{CLS}, начисляется один раз для группы подряд идущих символов;
- операция смены регистра ввода всего текста, ее цена w_{CC};

– операция смены регистра ввода одного символа, ее цена w_{CCS}, начисляется один раз для группы подряд идущих символов;
– операция удаления продублированного символа , ее цена w_{DD}.

II. Парольная аутентификация и нечёткий поиск

Для проверки особенностей алгоритма была сгенерирована случайная выборка из 30000 слов, и от каждого из восьми слов длиной от одного до восьми символов было рассчитано расстояние до каждого слова из выборки. Параметры алгоритма: $w_I = 0.8$, $w_D = 0.9$, $w_R = 1$, $w_T = 0.1$, $w_E = 0$, $w_{LS} = 0.5$, $N = 2$, $w_{GS} = 0.2$, $w_{GSS} = 0$, $w_{CL} = 0.1$, $w_{CLS} = 0.7$, $w_{CC} = 0.1$, $w_{CCS} = 0.7$, $w_{DD} = 0.1$. Результаты приведены на рис. 1-2.

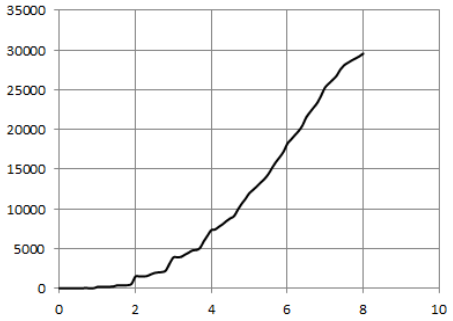

Рис. 1 – Зависимость количества успешных авторизаций от абсолютного редакционного расстояния

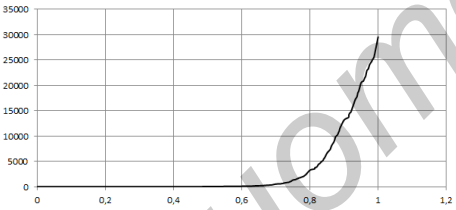

Рис. 2 – Зависимость количества успешных авторизаций от относительного редакционного расстояния

Относительное редакционное расстояние – это отношение редакционного (абсолютного) расстояния к длине пароля. Для определения сходства двух слов удобнее использовать именно его, т.к. одно и то же абсолютное расстояние для паролей различной длины имеет различный смысл (редакционное расстояние, равное пяти, для слов длиной шесть символов говорит больше об их различии, а для слов длиной 20 – об их сходстве). Графики показывают, сколько различных слов будет распознано алгоритмом как правильный пароль при заданном пороговом абсолютном или относительном расстоянии. Например, если одинаковыми считаются все слова с относительным расстоянием менее 0.6, то успешных авторизаций будет 97 (точка {0.6; 97} графика на рис. 2). Это означает, что при оптимальном алгоритме подбора пароля взломщиком время взлома сократится в 97 раз по сравнению со строгой парольной аутентификацией. Чтобы этого не произошло, пароль при нечёткой аутентификации должен иметь сложность в 97 раз выше, чем при строгой. Например, при использовании в пароле только английских строчных букв, длина пароля должна быть на $\sqrt[26]{97} \approx 1.19 < 2$ символа больше, чем при строгой аутентификации.

Отсюда следует, что при выборе порогового значения редакционного расстояния необходимо учитывать требования к паролю (минимальная длина, обязательное наличие цифр, прописных букв и спецсимволов и др.), и наоборот, при выборе требований к паролю учитывать пороговое расстояние.

Использовать неточный поиск можно во многих случаях, но особенно удобно при вводе пароля, когда вводимый текст нельзя проверить на отсутствие ошибок и опечаток.

Несмотря на достоинства, применение редакционного расстояния для проверки пароля накладывает ряд ограничений:

– Для сравнения введенной строки с паролем, метод хранения последнего должен позволять восстанавливать его значение. Из-за этого нельзя хранить в системе только результат применения к паролю односторонних функций (хеш-значение пароля) –самый распространенный метод, обеспечивающий высокий уровень безопасности.
– Невозможно использовать данный метод в системах с высокими требованиями к защите (где нарушение безопасности может повлечь финансовый ущерб или людские потери).
– Требования к паролю (минимальная длина, наличие цифр, прописных букв и спецсимволов и др.) должны быть более жесткими, чем в обычной системе. Доступ в систему предоставляется, даже если пароль введен не точно, т.е. вместо ввода пароля можно ввести любое слово из группы слов, которые близки (в смысле редакционного расстояния) к паролю. Простая атака полным перебором для одинаковых паролей займет в разы меньше времени, чем при строгой парольной аутентификации.

Например, с учетом ограничений областью применения может служить менеджер паролей на компьютере пользователя.

1. Расстояние Левенштейна [Электронный ресурс]. – 2013. – Режим доступа: http://ru.wikipedia.org/wiki/Расстояние_Левенштейна Дата доступа: 05.01.2013.
2. Damerau-Levenshtein distance [Electronic resource]. – 2013. – Mode of access: http://en.wikipedia.org/wiki/Damerau-Levenshtein_distance Date of access: 01.04.2013.