

В.А. ЛИТВИНОВ, С.Я. МАЙСТРЕНКО, И.Н. ОКСАНИЧ, В.И. ХОДАК

НЕКОТОРЫЕ МЕТОДЫ УСКОРЕНИЯ ОБРАБОТКИ СЛОВАРЯ ДОПУСТИМЫХ СЛОВ ПРИ АВТОМАТИЧЕСКОЙ ИДЕНТИФИКАЦИИ И ИСПРАВЛЕНИИ ОШИБОК ПОЛЬЗОВАТЕЛЯ

Abstract. Features, algorithms and the potential effectiveness of methods of dictionary processing of the permitted words during automatic identification and correction of user errors are examined. The methods are based on well-organized checking of words and arbitrary search of variations of a distorted word; sequential ("direct") search and distortion of dictionary words. Quantitative estimations and practical recommendations on the applicability of the methods are provided.

Key words: fuzzy search, errors of user, automatic errors correction.

Анотація. Розглядаються особливості, алгоритми й потенційна результативність методів обробки словника припустимих слів при автоматичній ідентифікації та виправленні помилок користувача. Методи засновані на упорядкованому переборі й довільному пошуку варіацій перекрученого слова; послідовному переборі й перекрученні слів словника. Приводяться кількісні оцінки і практичні рекомендації по застосуванню методів.

Ключові слова: нечіткий пошук, помилки користувача, автоматичне виправлення помилок.

Аннотация. Рассматриваются особенности, алгоритмы и потенциальная результативность методов обработки словаря допустимых слов при автоматической идентификации и исправлении ошибок пользователя. Методы основаны на упорядоченном переборе и произвольном поиске вариаций искаженного слова; последовательном переборе и искажении слов словаря. Приводятся количественные оценки и практические рекомендации по применимости методов.

Ключевые слова: нечеткий поиск, ошибки пользователя, автоматическое исправление ошибок.

1. Введение

Процесс идентификации ошибочного слова и автоматического исправления ошибок пользователя при вводе данных [1], нечетком поиске образца [2] и т.п. возможен на основе генерации множества всевозможных вариаций ошибочного слова («обратных» искажений в классах корректируемых ошибок) и поиске вариаций в словаре допустимых слов (СДС) [1].

При этом время обработки СДС растет пропорционально количеству сгенерированных вариаций, в свою очередь, зависящему от алфавита q , количества символов в слове n и видов корректируемых ошибок (транскрипции, транспозиции, пропуски символов и т.п.).

Потенциальным способом ускорения обработки СДС в этом случае являются учет вероятностей появления тех или иных ошибок пользователя и построение соответствующей последовательности перебираемых вариаций с целью уменьшения среднего объема перебора.

Наряду с этим возможно и принципиальное изменение схемы обработки с произвольной (поочередном поиске каждой отдельной вариации) на последовательную (поочередном «прямом» искажении каждого отдельного слова словаря). Можно ожидать, что при малых объемах СДС последовательная схема будет работать быстрее, чем произвольная. В статье рассматриваются особенности, алгоритмы и потенциальная результативность обеих возможностей ускорения обработки СДС.

2. Упорядоченный перебор вариаций искаженного слова

Естественным интуитивным решением представляются генерация и перебор вариаций в порядке убывания вероятностей p_j появления ошибок пользователя. Неэффективность такого решения

определяется существенной неравномерностью распределения количества вариаций V_j , которые требуется перебрать для ошибки класса E_j . Так, например, для идентификации однократной ошибки E_1 (вероятность $p_1 = 0,5557$ [1]) необходимо перебрать вариаций более, чем на порядок больше по сравнению с идентификацией вставки символа E_2 ($p_2 = 0,1567$) или транспозиции соседних символов E_4 ($p_4 = 0,0664$). Поэтому адекватная стратегия упорядоченного перебора должна базироваться на учете вероятностей ожидаемой результативности проверки одной вариации π_j , а не множества вариаций мощностью V_j . Такой подход попутно проясняет целесообразность рассмотрения и включения в ансамбль корректируемых ошибок и более редких ошибок, чем это принято в [1], при условии, что их идентификация требует перебора небольшого количества вариаций.

Обозначим через $\bar{v}(y)$ среднее количество переборов вариаций в порядке y до нахождения в словаре совпадения вариации с допустимым словом при условии, что произошла именно и только ошибка из ансамбля корректирующих ошибок.

В соответствии с определением

$$\bar{v}(y) = \sum_{x=1}^V \pi_x(y)x, \quad (1)$$

где V – суммарное количество сгенерированных вариаций;

$\pi_x(y)$ – удельная вероятность успешности одной вариации (в смысле совпадения с СДС).

Строго оптимальное решение о порядке y перебора вариаций предлагает следующая простая теорема 368 [3].

Пусть (a) и (b) есть две конечные системы (равновеликие множества) чисел.

Теорема. Если (a) и (b) заданы с точностью до перестановки, то $\sum ab$ принимает наибольшее значение, когда (a) и (b) обе монотонно убывают или обе монотонно возрастают, и наименьшее значение, когда одна из них монотонно возрастает, а другая – монотонно убывает.

Поскольку x в (1) монотонно возрастает, для минимизации \bar{v} порядок y должен располагать $\pi_x(y)$ строго в порядке убывания (невозрастания).

Группируя слагаемые (1) с одинаковыми значениями $\pi_x(y)$, получим

$$\bar{v}(y) = \sum_y \pi_y \left[V_y \sum_{k=1}^{y-1} V_k + \frac{V_y(V_y + 1)}{2} \right], \quad (2)$$

где $\pi_y \equiv \pi_x(y) = \frac{P_y}{V_y}$,

V_y – количество вариаций для ошибок класса E_j в порядке y .

В табл. 1 приведены исходные и расчетные данные для расширенного (по сравнению с [1]) ансамбля корректируемых ошибок из классов, приведенных в [4]: E_1 – однократные транскрипции, E_2 – добавление символа, E_3 – пропуск символа, E_4 – транспозиция смежных символов, E_5' – неспецифическая двукратная транскрипция, E_6 – двукратная транскрипция смежных идентичных символов, E_7 – транспозиция символов через одну позицию, E_8 – двукратная транскрипция идентичных символов, расположенных через одну позицию.

Примечание. Для множества E_5' ошибок класса E_5' справедливо $E_5' \cup E_6 \cup E_7 \cup E_8 = E_5$, где E_5 – множество произвольных двукратных ошибок [1] (искажений двух произвольных символов) за вычетом транспозиций смежных символов.

В табл. 1 приняты следующие дополнительные обозначения:

q – алфавит символов слова;

n – длина ошибочного слова (количество символов).

Таблица 1. Расчетные данные корректируемых ошибок

E_i	E_1	E_2	E_3	E_4	E_5'	E_6	E_7	E_8
p_i	0,5557	0,1567	0,1204	0,0664	0,0176	0,0081	0,0037	0,0028
$q=32; n=8$								
V_i	248	8	288	7	26833	31	6	31
$\pi_i \cdot 10^4$	22,4	195,9	4,2	94,9	0,007	2,6	6,2	0,9
y_0	1	2	3	4	5	6	7	8
y_1	3	1	5	2	8	6	4	7
y_2	6	8	4	7	1	3	5	2
$\bar{v}(y_0)$	901,3							
$\bar{v}(y_1)$	411,3							
$\bar{v}(y_2)$	27025,2							
$q=10; n=12$								
V_i	108	12	130	11	5271	9	10	9
$\pi_i \cdot 10^4$	51,5	130,6	9,3	60,4	0,033	9	3,7	3,1
y_0	1	2	3	4	5	6	7	8
y_1	3	1	4	2	8	5	6	7
y_2	6	8	4	7	1	3	5	2
$\bar{v}(y_0)$	236,1							
$\bar{v}(y_1)$	133,6							
$\bar{v}(y_2)$	5423,3							

3. Последовательный алгоритм обработки словаря

Примем следующие обозначения:

$A_j = (a_1, \dots, a_i, \dots, a_m)$ – очередное эталонное слово СДС ($j = 1, \dots, N$);

$B = (b_1, \dots, b_k, \dots, b_n)$ – ошибочное слово, для которого в СДС ищется «ближайшее» слово, отличающееся от B каким-либо прямым искажением, принадлежащим к множеству корректируемых ошибок.

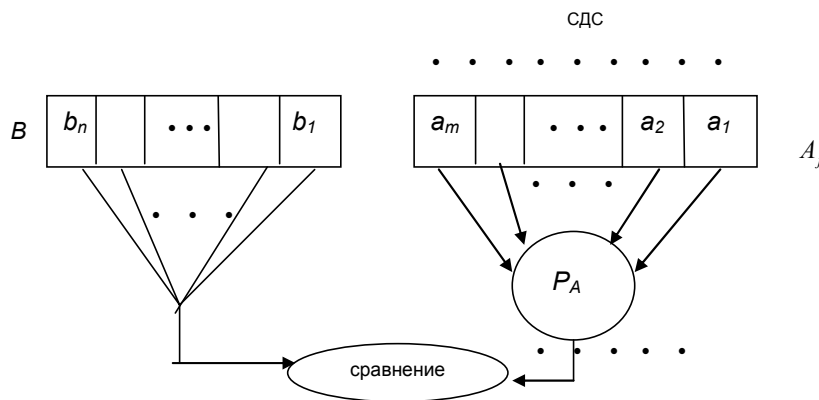


Рис. 1. Укрупненная схема последовательного процесса искажения A_j и идентификации ошибки

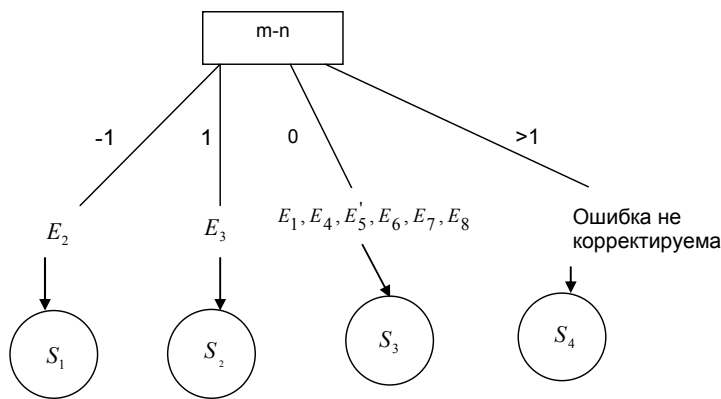


Рис. 2. Первый этап синтаксического анализа

Укрупненная схема процесса поочередного искажения слов A_j показана на рис. 1. Через $P(A)$ обозначен оператор прямого искажения слова A_j . Факт предполагаемой идентификации ошибки определяется результатом сравнения $P_A(A_j) = B$.

Алгоритм анализа слов

A_j, B

Первым этапом синтаксического анализа слов A_j, B является вычисление значения $(m - n)$, позволяющее совершенно определенно отнести потенциальную ошибку к одной из четырех категорий возможных ошибок в B относительно A_j :

добавление символа (E_2), пропуск символа (E_3), транскрипции или транспозиции разных видов (E_1, E_4, E_5, E_6, E_7), какая-то некорректируемая ошибка (рис. 2). Через $S_1 - S_4$ обозначены процедуры, составляющие второй этап анализа.

Условия выбора процедур определяются следующим правилом выбора (в понятиях и терминах PASCAL):

- CASE (m-n) of
- 1: процедура S_1 ;
 - 1: процедура S_2 ;
 - 0: процедура S_3 ;
 - >1: процедура S_4 .

Прежде чем перейти к описанию следующего этапа анализа, отметим следующее. Для того, чтобы идентифицировать ошибку E_2 (добавление символа), следовало бы в соответствии со схемой рис. 1 генерировать вставки символов со значениями, соответствующими алфавитным номерам $0 \div (q - 1)$, поочередно в позиции $a_{m+1} \div a_m, a_m \div a_{m-1}, \dots, a_2 \div a_1$ слова A_j . Такая процедура требует генерации $(m - 1)q$ вариантов искажений слова A_j и сравнения этих

вариантов с B для проверки совпадения. Очевидно, что этого же результата можно добиться, варьируя слово B путем поочередного удаления символов a_n, a_{n-1}, \dots, a_1 . В этом случае требуются генерация и анализ всего n вариантов ($n \ll m - 1$) q .

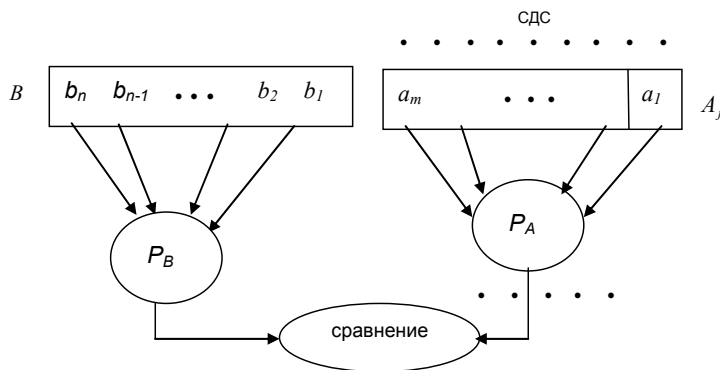


Рис. 3. Комбинированная схема искажения A_j, B

Таким образом, целесообразно несколько усложнить схему рис. 1 и принять комбинированную схему анализа (рис. 3), включающую в целесообразных случаях как прямые искажения A_j (оператор P_A), так и обратные искажения B (оператор P_B). В этом случае факт

предполагаемой идентификации ошибки определяется результатом сравнения $P_A(A_j) = P_B(B)$, а процедуры $S_1 \div S_4$ определяются соответствующими $P_A(A_j), P_B(B)$.

Процедура S_1 (предполагаемая ошибка – добавление символа).

Оператор P_A – пустой.

Оператор P_B :

Поочередное удаление символов b_k ($k = 1, n$) и сравнение оставшейся части:

$$B'(k) = (b_1, \dots, b_{k-1}, b_{k+1}, \dots, b_n) \text{ с } A_j.$$

Если для какого-то значения $B'(k)$ существует такое $k = l$, что $B'(l) \equiv A_j$, то A_j есть искомое эталонное слово, в котором произошла ошибка E_2 – добавление символа b_l (в общем случае, конечно, – одно из искомых слов). Однозначность идентификации ошибки E_2 определяется следующим очевидным утверждением: не существует двух или более различных значений $B'(k)$, совпадающих с одним и тем же значением A_j .

Иначе – ошибка не идентифицирована. Переход к A_{j+1} .

Процедура S_2 (предполагаемая ошибка – пропуск символа).

Оператор P_B – пустой.

Оператор P_A :

Поочередное удаление символов a_r ($r = 1, m$) и сравнение оставшейся части:

$$A'(r) = (a_1, \dots, a_{r-1}, a_{r+1}, \dots, a_m) \text{ с } B.$$

Если для какого-то значения $A'(r)$ существует такое $r = s$, что $A'(s) \equiv B$, то A_j есть искомое эталонное слово, в котором произошла ошибка E_3 – пропуск символа a_s . Однозначность идентификации ошибки определяется утверждением, аналогичным сделанному выше в описании оператора P_B процедуры S_1 : не существует двух или более различных значений $A'(r)$, совпадающих с одним и тем же значением B .

Иначе – ошибка не идентифицирована. Переход к A_{j+1} .

Процедура S_3 (предполагаемая ошибка – однократная или двукратная транскрипция или транспозиция).

Операторы P_A, P_B :

Поочередное сравнение значений символов a_i, b_i (здесь $m = n$, $i \equiv k$) и определение количества не совпавших символов α .

Case α of

1: идентификация A_j как эталонного слова, в котором произошла однократная транскрипция E_1 ; переход к A_{j+1} ;

2: идентификация A_j как эталонного слова, в котором произошло искажение двух символов (одна из ошибок E_4, E_5, E_6, E_7); переход к A_{j+1} ;

>2: идентификация ошибки как некорректируемой; переход $A_j \rightarrow A_{j+1}$.

Примечание. Если установлено, что в слове B искажены в точности 2 символа ($\alpha = 2$), то A_j может быть идентифицировано как эталонное слово, соответствующее B , независимо от того, какая именно произошла ошибка: E_4, E_5', E_6, E_7, E_8 .

Процедура S_4 .

Операторы P_B, P_A – пустые.

Идентификация ошибки как некорректируемой; переход $A_j \rightarrow A_{j+1}$.

Аналитическое определение сравнительного быстродействия произвольного и последовательного алгоритмов представляет значительные трудности, связанные, в первую очередь, с различным характером выполняемых операций при произвольном поиске и последовательном переборе и, соответственно, с необходимостью учета большого количества разнородных величин.

В связи с этим сравнение алгоритмов проведено с помощью имитационной модели, запрограммированной в системе Delphi.

Произвольный алгоритм обработки словаря базировался на дихотомическом поиске очередной вариации в СДС, последовательный – на описанной выше схеме последовательной обработки.

Скорость работы обоих алгоритмов определялась на пакете из 100 ошибочных слов, искаженных ошибками $E_1 \div E_8$ в пропорциях, соответствующих принятым вероятностям появления ошибок (55% – E_1 , 15% – E_2 и т.д.).

Результаты имитационного моделирования отражены в табл. 2. В таблице приведены относительные значения $\sigma = \frac{t_{np}}{t_{посл}}$, где t_{np} и $t_{посл}$ – время обработки пакета ошибочных слов произвольным и последовательным алгоритмами соответственно. Закрашенные клетки таблицы означают преимущество произвольного алгоритма. Через \bar{m} обозначено среднее количество символов слов в СДС, через q – алфавит символов слов СДС.

Таблица 2. Результаты имитационного моделирования

N	$q=36, \bar{m}=8$	$q=10, \bar{m}=12$
	σ	σ
10^2	408	74
10^3	233	37
10^4	20,8	3,58
10^5	2,4	0,42
$2 \cdot 10^5$	1,05	0,20
$3 \cdot 10^5$	0,76	0,15
$6 \cdot 10^5$	0,37	0,08

Для относительно слабого компьютера P4 (2,8 ГГц, 260 мб) абсолютное значение t_{np} для $N = 2 \cdot 10^5$ соответствует 282 мс.

4. Заключение

Анализ полученных результатов дает основание для следующих выводов:

1) Упорядоченный по удельным вероятностям перебор вариаций искаженного слова заметно снижает объем перебора. Так, в идеализированном случае, когда произошла именно корректируемая ошибка (вероятность $\sum p_j = 0,9314$), значения $\bar{v}(y_1)$ для оптимальной упорядоченности равны 411,3 и 133,6, $\bar{v}(y_0)$, для интуитивной стратегии перебора в порядке убывания p_j равны 901,3 и 236,1, а наихудший порядок перебора дает значения $\bar{v}(y_2)$, равные 27025,2 и 5423,3. Этот результат может иметь практическое значение применительно к стратегии 1, 2 (алгоритмов принятия решений при анализе результатов поиска вариаций в словаре) [1] и для малых значений $r = \frac{N}{q^m}$ и V , когда ожидаемое количество случайных (т.е. ложных) совпадений вариаций со словарем мало.

2) Границы целесообразного приложения последовательной схемы обработки определяются областью больших значений V и малых значений N . Если ансамбль корректируемых ошибок полон, т.е. включает все ошибки классов $E_1 - E_5$ (как было отмечено выше, $E_5 = E'_5 \cup E_6 \cup E_7 \cup E_8$), то границы этой области иллюстрируются данными табл. 2: $N < 2 \cdot 10^5$ для $q = 36, m = 8$ и $N < \sim 5 \cdot 10^4$ для $q = 10, m = 12$. Если исключить из ансамбля ошибки класса E'_5 , вносящие наибольший вклад в суммарное количество вариаций и, соответственно, в ожидаемое количество ложных совпадений, то имеет место следующее. С одной стороны, суммарное количество вариаций уменьшается в ~ 40 раз для $q = 36, m = 8$ и в ~ 10 раз для $q = 10, m = 12$. Примерно во столько же раз уменьшается и значение t_{np} . С другой стороны, несколько возрастает значение $t_{посл}$ за счет дополнительного дифференцированного анализа ошибок E_6, E_7, E_8 и, соответственно, усложнения процедуры S_3 . В результате граница области сдвинется примерно в район значений $N < \sim (3 \div 5) \cdot 10^3$ для $q = 36, m = 8$ и $N < \sim (2 \div 3) \cdot 10^3$ для $q = 10, m = 12$. Таким образом, данные табл. 2 иллюстрируют наиболее благоприятный для применения последовательной схемы обработки случай, соответствующий анализу полного ансамбля ошибок.

СПИСОК ЛИТЕРАТУРЫ

1. Алгоритми і моделі автоматичної ідентифікації та корекції типових помилок користувача на основі природної надмірності / Г.Є. Кузьменко, В.А. Литвинов, С.Я. Майстренко [та ін.] // Математичні машини і системи. – 2004. – № 2. – С.134 – 148.
2. Ukrainian Context Optimizer/ <http://www.uco.ua/infosection 8-doc12>.
3. Харди Г.Е. Неравенства / Харди Г.Е., Литтлвуд Д.Е., Полиа Г. – М.: Государственное издательство иностранной литературы, 1948. – С. 316.
4. Литвинов В.А. Контроль достоверности и восстановление информации в человеко-машинных системах / В.А. Литвинов, В.В. Крамаренко. – Киев: Техника, 1986. – 200 с.

Стаття надійшла до редакції 06.10.2009