УДК 681.142.2

Э. А. ДЕДИКОВ, канд. техн. наук,
Р. Н. ЧЕН, канд. техн. наук

НЕКОТОРЫЕ ПРИНЦИПЫ РАСПОЗНАВАНИЯ ОШИБОК В ИМЕНАХ

Одним из достоинств любой автоматизированной системы обработки данных (АСОД) является возможность автоматически установить правильность написания имен (или запросов) пользователем этой системы. Анализ наиболее часто совершаемых ошибок (или неточностей в формулировании и написании) в именах запрашиваемых документов позволил выделить следующие основные их типы [1]: 1) наличие орфографической ошибки в одном из слов, входящих в имя, т. е. имеется одно неправильное слово; 2) наличие более одного неправильного слова в имени; 3) перестановка двух смежных слов в имени; 4) перестановка двух несмежных слов в имени; 5) перестановка более двух слов в имени; 6) пропуск одного слова; 7) пропуск более одного слова; 8) добавление лишнего слова в имя; 9) добавление более одного слова в имя.

В соответствии с перечисленными типами ошибок предлагается следующий принцип распознавания их в именах. Установление неточности в словах (т. е. наличие ошибки) производится с помощью построения корректирующего словаря [1]. Для распознавания ошибок в именах предлагается строить отдельный словарь связей. Таким образом, имеется два этапа в организации словарей: формирование словаря слов; формирование словаря имен и связей.

Организация словаря ключевых слов. Каждое слово, входящее в имя, записывается в собственно словарь ключевых слов [2]. Применение метода хеширования в данном случае дает наилучшие результаты, так как он является самым быстродействующим методом программного поиска. Это его качество проявляется особенно ярко при работе с наборами данных большого размера [3]. В соответствии с этим методом каждое ключевое слово помещается в ячейку памяти (а затем выбирается оттуда), адрес которой — значение простой арифметической функции содержимого ключевого слова. В качестве «содержимого», к которому применяется функция хеширования, можно взять все или некоторые символы, входящие в ключевое слово.

Выбор числа символов, рассматриваемых как новое хешируемое ключевое слово, зависит от объема имеющейся памяти

[1] Два имени считаются равными, если равны входящие в него слова на соответствующих позициях, при этом имена должны быть равной длины, т. е. содержать одинаковое количество слов. Слова A и B равны ($A = B$), если равны их длины ($l_A = l_B$) и символы этих слов, расположенные на одинаковых местах, совпадают [1].

ЭВМ, а также от используемых методов хеширования. Определение нового ключевого слова зависит как от содержания исходного ключевого слова, так и от их количества. Для хеширования слов русского языка использовать монограммы, биграммы или все символы слова неэффективно, так как в первом случае можно получить всего $32+32^2$ адресов (это слишком мало для размещения даже минимума слов), а во втором — 32^{18} адресов (18 — это среднестатистическая длина слов в общелитературном русском языке). При этом получается избыточное число адресов, получаемых из несуществующих слов, что ведет к очень низкому коэффициенту заполнения памяти. Чтобы получить разумное число адресов, требующее минимум памяти, предлагается использовать в качестве новых ключевых слов триграммы, т. е. хешировать первые три символа из каждого ключевого слова. Такое хеширование позволит обнаружить одиночную ошибку[1] только в первых трех символах. Разрешение коллизий в словаре ключевых слов (рис. 1) производится методом внутренней адресации. Предпочтение этому методу отдано в силу того, что для больших наборов данных создается большая хеш-таблица, которую полностью можно поместить на внешних запоминающих устройствах. Обработка таких внешних файлов производится частями, которые можно пересылать в оперативную память и которые содержат всю информацию, требующуюся для проверки сходства слов. Для поиска пустой ячейки производится последовательный просмотр всей хеш-таблицы. В случае, когда по вычисленному адресу нет требуемого слова, процедуру пробинга необходимо выполнять по цепочке UKA до тех пор, пока искомое слово не будет найдено.

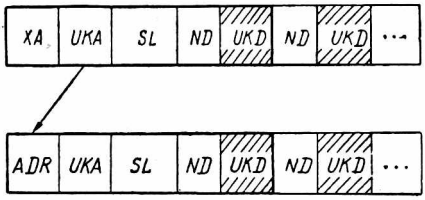

Рис. 1. Структурная схема словаря ключевых слов: *SL* — хешируемая триграмма исходного ключевого слова; *XA* — хеш-адрес слова *SL*; *UKA* — указатель на адрес свободной ячейки (для разрешения коллизий); *ND* — номер имени документа, в котором встречается слово *SL*; *UKD* — указатель, т. е. адрес ячейки на следующий номер документа, который в имени также содержит данное слово *SL*; *ADR* — адрес свободной ячейки

Организация словаря имен и связей. Такой словарь (рис. 2) содержит информацию о связях имен документов по входящим в них словам. Основное его назначение заключается в том, чтобы при необходимости можно было выбрать все имена документов, которые содержат 1, 2, ..., n равных триграмм. Значение $n=15$, так как из практики видно, что большинство имен документов содержит не более 15 слов.

Таким образом, в словаре связей для каждого номера доку-

[1] Ошибкой может быть и случай, когда первые три символа двух различных правильных слов отличаются одним символом в одной позиции.

мента *ND* с именем *ИD* записываются все номера документов, которые содержат в имени 1, 2, ..., *n* триграмм, входящих и в *ИD*.

Организация поиска подобных имен. Имена документов, содержащие хотя бы одно одинаковое слово, назовем подобными. Степень подобия определяется количеством совпавших слов, т. е. триграмм. В зависимости от заданной степени подобия *n* из файла связей выбираются все номера документов, которые содержат в имени *n* триграмм, имеющихся также и в исходном имени документа. Алгоритм поиска подобных имен следующий.

1. Найти входное имя документа. Если найдено, перейти на КОНЕЦ.

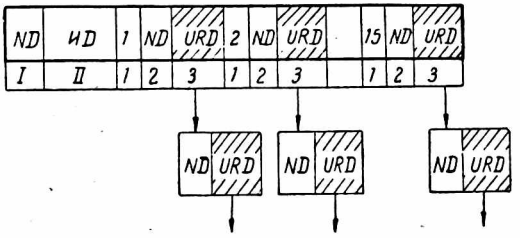

Рис. 2. Структурная схема словаря имен и связей: *I*, *II*, *1*, *2*, *3* — номера полей; *I* — *ND* — номер имени документа; *II* — *ИD* — имя документа, состоящее из триграмм; *1* = 1, 2, . . ., 15 — номера участков памяти, указывающие на количество совпавших триграмм в именах документов; *2* — *ND* — номер документа, который содержит *n* равных слов с *ИD* (поле *II*); *3* — *URD* — указатель, т. е. значение адреса, на следующий номер документа, содержащий в своем имени *n* равных слов

2. Определить все входящие в поисковое имя триграммы.

3. Для каждой триграммы в словаре ключевых слов найти все номера документов {*N*}, в которые она входит.

4. Через консоль (или другим каким-нибудь образом) получить значение степени подобия *n*, т. е. число из интервала [1, 15].

5. Выбрать из {*N*} все номера документов, в имена которых входит *n* триграмм, входящих также во входное поисковое имя.

6. Выдать все выделенные номера документов на консоль или АЦПУ.

7. Конец.

В результате поиска пользователю выдаются все имена, подобные входному, с заданной степенью подобия. Из них пользователь может указать на одно, являющееся правильным относительно входного. Если будет найдено только одно имя, то, по соглашению, оно может обрабатываться дальше как правильное, либо пользователь должен подтвердить правильность найденного корректирующего имени документа.

Таким образом, для распознавания ошибок в именах предлагаются следующие этапы работ: создание корректирующего словаря ключевых слов; организация словаря связей имен документов; установление степени подобия имен документов;

поиск имен документов, подобных входному, в соответствии с заданной степенью подобия; выдача пользователю множества подобных имен для выбора корректирующего.

Список литературы: 1. *Дедиков Э. А., Чен Р. Н.* Организация корректирующего машинного словаря имен с помощью аддитивной функции хеширования.— Пробл. бионики, 1982, вып. 28, с. 14—19. 2. *Чен Р. Н.* Об алгоритме исправления ошибок орфографии с использованием метода корректирующего словаря.— Пробл. бионики, 1982, вып. 29, с. 32—35. 3. *Кнут Д.* Искусство программирования для ЭВМ: Сортировка и поиск.— М.: Мир, 1978.— 846 с.

Поступила в редколлегию 19.06.84.