Определение расстояния между словами в алгоритмах словарной корректировки результатов распознавания

В. М. Кляцкин[1], Н. В. Котович[1], О. А. Славин[2]

[1] *Научно-исследовательский институт системных исследований Российской академии наук*,
Россия, 117218 Москва, Нахимовский пр., 36, к. 1
[2] *Институт системного анализа Российской академии наук,*
Россия, 117312 Москва, пр. 60-летия Октября, 9

Рассматриваются алгоритмы словарной коррекции символов, распознанных с помощью шрифтозависимых и шрифтонезависимых механизмов, выделяется случай механизмов с самообучением. Приводится модель словарной корректировки. Определяются простые случаи словарной корректировки и соответствующие им функции расстояния. На основе элементарных функций расстояния определяется общий случай расстояния между распознанным словом и словарной гипотезой.

Введение

Словарная корректировка результатов распознавания является мощным средством ликвидации ошибок. В общем случае, когда результаты распознавания представлены в виде графа с двумя выделенными вершинами (началом и концом), сопоставление некоторого слова-образца с результатом распознавания основывается на алгоритмах динамического программирования [1]. Расстояние между образцом и результатом распознавания может быть определено алгоритмом нахождения минимального расстояния Левенштейна [2].

В настоящей работе мы рассмотрим несколько простых функций расстояния между двумя словами, которые могут быть использованы в словарной корректировке.

1. Словарная коррекция

Мы рассматриваем словарный механизм как средство формирования одной или нескольких гипотез изменения нескольких подряд идущих символов. То есть для последовательности, состоящей из нескольких символов $W_0 = C_1, \ldots, C_n$, словарный механизм генерирует несколько близких последовательностей символов:

$$W_1 = c_1^{(1)}, \ldots, c_{n_1}^{(1)}, \ \ldots, W_k = c_1^{(k)}, \ldots, c_{n_k}^{(k)}.$$

Близость каждого из слов W_i ($1 \le I \le k$) к исходному слову W_0 оценивается с помощью некоторой функции расстояния $d(W_0, W_i)$.

Возможны различные способы формирования гипотез, например:

- *механизм корпуса слов*, базирующийся на достаточно большом словаре определенного языка;
- *механизм сочетаний* подряд идущих последовательностей (пар и троек) букв, учитывающий частоты встречаемости.

В дальнейшем будем говорить о словарной корректировке, абстрагируясь от алгоритма формирования гипотез. Словарной корректировке подвергаются слова, состоящие из нескольких образов символов $W = S_1, S_2, \ldots, S_n$, каждому из которых соответствует коллекция альтернатив распознания:

$$R(S_1) = \left\{\left(C_1^1, w_1^1\right)\ldots\left(C_{M_1}^1, w_{M_1}^1\right)\right\}, \ \ldots, R(S_n) = \left\{\left(C_1^n, w_1^n\right)\ldots\left(C_{Mn}^n, w_{Mn}^n\right)\right\}. \quad (1)$$

При этом в слове могут существовать нераспознанные символы S_i с пустыми коллекциями $R(S_i) = \varnothing$ ($M_i = 0$).

Возможен случай, когда слово W подтверждается (если в словаре было найдено слово $C_1^1 \ldots C_1^n$), при этом все символы слова W приобретают признак соответствия словарному корпусу.

В случае если распознанное слово отсутствует в словаре, оно может быть заменено на иное, присутствующее в словаре. При этом возможны две стратегии замены:

- *стандартная*, в которой происходит замена на символы, присутствующие среди альтернатив распознавания со сравнимыми оценками;
- *агрессивная* — символы могут быть заменены иными символами, даже отсутствующими среди альтернатив.

Стандартная словарная корректировка приводит к изменению коллекции альтернатив распознавания у каждого из символов слова:

$$R\{S_1\} = \left\{\left(\dot{C}_1^1, \dot{w}_1^1\right)\ldots\left(\dot{C}_{M_1}^1, \dot{w}_{M_1}^1\right)\right\}, \ldots, R\{S_n\} = \left\{\left(\dot{C}_1^n, \dot{w}_1^n\right)\ldots\left(\dot{C}_{Mn}^n, \dot{w}_{Mn}^n\right)\right\}.$$

Для каждого символа S_i замена C_1^i на $\dot{C}_1^i$ возможна, если $\dot{C}_1^i$ совпадает с одним из кодов C_j^i $(1 \le j \le M_i)$, при этом должна быть ограничена разница между оценками надежности $\left|w_1^i - w_{M_i}^i\right|$. Также возможна подстановка произвольного кода символа S_i для образа с пустой коллекцией распознавания $R\{S_i\} = \varnothing$. Границы образов символов и их количество при стандартной словарной корректировке не изменяются.

Агрессивная словарная корректировка может изменять количество образов символов с целью исправления ошибок сегментации на первом и последующем проходах распознавания. Но в то же время при таком подходе возрастает вероятность внесения ошибки — в исходном тексте всегда могут встретиться слова, отсутствующие в словаре, например фамилии, аббревиатуры и т. п., и эти слова не должны быть заменены на словарные.

При любом способе словарной корректировки важным вопросом является выбор функции расстояния между словами.

2. Функция расстояния между двумя словами

Рассмотрим слово $A = \{A_1, \ldots, A_n\}$, соответствующее последовательности распознанных символов, и слово $B = \{B_1, \ldots, B_m\}$, являющееся словарной гипотезой.

Определим несколько расстояний между отдельными символами для следующих случаев:

- *замена* символа a на символ b: $d_s(a, b)$;
- *удаление* символа a: $d_d(a)$;
- *вставка* символа b: $d_i(b)$;
- *слияние* символов a_1 и a_2 в символ b: $d_g(a_1, a_2, b)$;
- *разбиение* символа a на символы b_1 и b_2: $d_c(a, b_1, b_2)$.

Расстояние между словами A и B будем вычислять следующим образом:

$$D(A, B) = \sum_{i=1}^{k} d(i), \qquad (2)$$

где $d(i)$ — одно из пяти расстояний, определенных выше, выбранное таким образом, чтобы минимизировать $D(A, B)$ во время построения оптимального соответствия между двумя последовательностями:

$$M = \{m_i, i = 1, \ldots, k\},$$

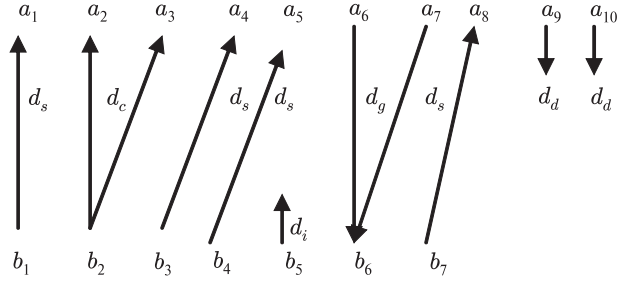

Рис. 1. Случаи соответствия символов двух слов

где m_i — элементарное отображение (отображение «пары» подмножеств последовательностей A и B).

Замечание: в случаях удаления и вставки символа одно из подмножеств $A(m_i)$ или $B(m_i)$ является пустым:

$$d_d(a) = d_d(a,\varnothing)\ ,\ d_i(b) = d_i(b,\varnothing)\ .$$

Наилучшее соответствие (расстояние) между двумя сравниваемыми последовательностями определим следующим образом (рис. 1):

$$M^* = \arg\min\{D(M)\}\ ,$$

где

$$D(M) = \sum_{i=1}^{k} d(A(m_i), B(m_i))\ ,$$

$d(A(m_i), B(m_i))$ — общая функция расстояния между двумя подпоследовательностями слов.

Конкретный вид введенных пяти элементарных функций расстояния сильно зависит от решаемой задачи. Например, расстояние замены символа может быть определено различным образом для случая коррекции напечатанной орфографической ошибки, когда вероятность ошибки напрямую зависит от геометрической близости клавиш-букв на клавиатуре печатающего устройства (например, вероятность будет высока для близкой на клавиатуре пары совершенно несхожих по написанию букв «ж» и «э») и коррекции ошибок распознавания, когда ошибка повышается для схожих по написанию букв («и», «н»).

Дадим определение расстояния замены для распознанных символов:

$$d_s(a, b) = d_s^{clust}(a, b), \text{ если } b \in \textbf{\textit{clust}}(a),$$

где ***clust***(a) — класс графем символа a, состоящий из символов, неразличимых по начертанию от начертания a, например {o, O, 0} или{1, l, i}. Нераз-

личимость определяется алгоритмом распознавания образов символов. Если используется шрифтонезависимый метод [3], то приведенные примеры классов содержат неразличимые образы. Если же используется шрифтозависимый метод [4], то каждый из приведенных примеров классов распадается на три отдельных класса.

Другим случаем расстояния d_s является

$$d_s(a, b) = d_{similar}(a, b), \text{ если } b \in \textbf{\textit{similar}}(a),$$

где ***similar*** (a) — класс графем символа a, состоящий из символов, сходных по начертанию, например, {o, O, 0, D} или {s, S, 5}. В отношении сходства по начертанию справедливо сделанное выше замечание о природе и характеристиках применяемого алгоритма распознавания образов символов.

Наконец, в оставшихся случаях

$$d_s(a, b) = P_{sub},$$

где P_{sub} — штраф за различие между символами a и b.

Если символ b принадлежит классу разделителей, необходим специальный штраф. Например, к классу разделителей относятся пробелы и знаки пунктуации. Штраф за замену нераспознанного символа на любой символ алфавита будет наиболее мягким. Ниже не детализируется специфика учета нераспознанных символов для случаев типа слияния, разделения, вставки и удаления.

Дадим определение расстояния слияния распознанных символов:

$$d_g(a_1, a_2, b) = d_g^{clust}(a_1, a_2, b), \text{ если тройка } <a_1, a_2, b> \in \textbf{\textit{clust}}_{\mathbf{21}}(a_1, a_2, b),$$

где $\textbf{\textit{clust}}_{\mathbf{21}}(a_1, a_2, b)$ — класс, состоящий из таких символов, что пара образов a_1 и a_2 неразличима (либо схожа) по начертанию от начертания b, например {<nn> m }, { <ll> n}.

Заметим, что в общем случае штрафы каждого из кластеров слияния индивидуальны.

В противном случае

$$d_g(a_1, a_2, b) = P_{glue},$$

где P_{glue} — штраф за подстановку одного символа вместо двух.

Дадим определение расстояния разбиения распознанных символов:

$$d_c(a, b_1, b_2) = d_c^{clust}(a, b_1, b_2), \text{ если тройка } <a_1, a_2, b> \in \textbf{\textit{clust}}_{\mathbf{12}}(a, b_1, b_2),$$

где $\textbf{\textit{clust}}_{\mathbf{21}}(a, b_1, b_2)$ — класс, состоящий из таких символов, что пара образов b_1 и b_2 неразличима по начертанию от начертания a, например {w, <vv> }.

В противном случае

$$d_g(a, b_1, b_2) = P_{cut},$$

где P_{glue} — штраф за подстановку двух символов вместо одного.

Штрафы удаления d_d и вставки d_i определяются более просто как некоторые константы, зависящие от контекста задачи (ввод текста оператором или корректировка ошибок распознавания), а также от типа вставляемого/удаляемого символа (очевидно, штраф за удаление геометрически малого объекта типа точки должен быть меньше, нежели штраф за удаление буквы стандартного размера).

Таким образом, в данном разделе введено понятие расстояния на последовательностях текстовых символов, основанное на описаниях элементарных функций расстояния. Отметим, что в реальных задачах большое значение имеет подбор констант, участвующих в вышеприведенных формулах.

3. Применение альтернативности результатов

Представление результатов распознавания (1) предполагает широкий спектр альтернатив слова из n символов:

$$W_{i_1 \ldots i_n} = C^n_{i_1}, \ldots, C^n_{i_n},\ 1 \le i_k \le M_k,\ 1 \le k \le n,$$

т. е. число альтернатив слов составляет $\prod_{i=1}^{n} M_i$. Каждое слово $W_{i_1 \ldots i_n}$ может быть сопоставлено со словарными гипотезами с помощью формулы (2).

Возможно сокращение перебора комбинирования альтернатив слов за счет использования оценок надежности распознавания символов. Символы S_i с оценками w_i, меньшими пороговой оценки w_0, могут подвергаться любому способу корректировки, описанному в разделе 2. Оценки символов w_i, большие пороговой оценки w_0, должны быть учтены в модификации формулы (2):

$$D(A, B) = \sum_{i=1}^{k} w(i) d(i). \qquad (2')$$

Для случаев замены (d_s), удаления (d_d), вставки (d_i) и разбиения (d_c) оценки $w(i)$ в формуле (2′) совпадают с оценкой надежности символа a. Для случая слияния символов a_1 и a_2 в один символ (d_g) $w(i) = \min(w_{a1}, w_{a2})$.

Особой является словарная корректировка слов, распознанных с помощью второго прохода многопроходной схемы распознавания документов с самообучением, описанной в [4]. Символы, надежно распознанные на пер-

вом проходе и затем подтвержденные посредством использования кластерных эталонов на втором проходе, запрещается заменять при словарной корректировке. В то же время ненадежно распознанные символы разрешается менять при помощи словаря.

Указанные приемы позволяют как сокращать время перебора альтернатив распознанных слов, так и уменьшать число ошибок, совершенных за счет артефактов словарной корректировки.

Литература

1. *Sholomov D. L.* Syntactical Approach to Post-Processing of Fuzzy Recognized Text // Proc. of The International Conference on Machine Learning, Technologies and Applications. USA: CSREA Press, June 2003. P. 115–121.
2. *Левенштейн В. И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. 1965. Т. 163. № 4. С. 845–848.
3. *Гавриков М. Б., Мисюрев А. В., Пестрякова Н. В.*, *Славин О. А.* Об одном методе распознавания символов, основанном на полиномиальной регрессии // Автоматика и телемеханика. 2006. № 3. С. 119–134.
4. *Арлазаров В. Л., Котович Н. В., Славин О. А.* Адаптивное распознавание // Информационные технологии и вычислительные системы. 2002. № 4. С. 11–22.