КОНТРОЛЬ ОШИБОК В ТЕКСТАХ НА ОСНОВЕ ОПТИЧЕСКОГО РАСПОЗНАВАНИЯ ТЕКСТОВ НА УЗБЕКСКОМ ЯЗЫКЕ

С. Холмонов

Узбекистан, Самаркандский государственный университет

Постановка задачи. В настоящее время специалистами уделяется большое внимание контролю и коррекции орфографических ошибок текстов, представленных на естественном языке [1,2]. В данной работе рассмотрена задача контроля ошибок в текстах на узбекском языке путем реализации системы оптического распознавания текстов (СОР). В качестве базового инструмента рассмотрены основные принципы распознавания текстов и алгоритм распознавания символов WSA. Рассматриваемая реализация СОР обладает возможностью распознавания многоколоночного текста, позволяет автоматически определять сегменты текста, подлежащие распознаванию. Позволяет автоматически удалять картинки из области распознаваемого текста.

Реализация СОР. Задачу распознавания текста можно разбить на несколько подзадач. Текст может включать рисунки, может быть разбит на столбцы или располагаться на странице как-то иначе. Таким образом, на первой стадии необходимо найти на странице собственно текст, который в дальнейшем будет подвергнут распознаванию и коррекции. Далее найденный участок текста следует разбить на символы и выделить строки. Когда строки и символы найдены, то можно произвести распознавание символов и проверить правильность полученных слов, например, при помощи словаря.

Блок и разбиение растра на блоки. Будем полагать, что оригинальное изображение - двухцветное. Буквы - черные, а фон - белый.

Назовем блоком такую прямоугольную область изображения, что:

- внутри области у каждой из сторон прямоугольника есть хотя бы один черный пиксель;
- область не содержит черных пикселей, соседствующих с черными пикселями, не принадлежащими данной области;
- область нельзя разбить на меньшие непересекающиеся блоки.

Очевидно, что любой растр может быть разбит на блоки.

Область не может являться блоком только тогда, когда не удовлетворен второй пункт определения блока или она пересекается с другой областью. В таком случае объединяем две области в одну и продолжаем до тех пор, пока все области не станут блоками.

Процедура разбиения на блоки используется в преобразованиях, описанных ниже.

Разбиение текста на сегменты и колонки. Уменьшим исходное изображение. Все символы, принадлежащие одному сегменту текста, сольются и будут образовывать один большой блок. Таким образом, разбивая на блоки уменьшенное изображение, можно выявить сегменты текста.

Предположим, что все колонки в тексте имеют одинаковую ширину. Тогда ширина растра, деленная на среднюю ширину сегмента будет определять число колонок в тексте.

Далее предполагается, что данная процедура успешно разбивает изображение на сегменты, каждый из которых представляет собой одну колонку текста.

Разбиение колонки (сегмента) на символы. Разбиение сегмента на символы осуществляет процедура разбиения растра на блоки. Предполагается, что в один блок попадает только один символ или его часть. В принципе одному символу могут соответствовать несколько блоков. Например, символы '!', '?', или, например, буква 'ы' будут попадать в два блока. После выявления строк в сегменте будет проведена процедура коррекции блоков, исправляющая этот недостаток.

Выявление строк. Выбираем в сегменте самый верхний правый блок. Каждому блоку ставим в соответствие два флага: flag_1 и flag_2.

Для первого блока flag_2 = false, а flag_1 = true, если справа есть блоки с такой же Y-координатой. И так далее, для символа, стоящего справа. Причем flag_2 равен flag_1 у предыдущего блока.

Таким образом, возможны следующие ситуации:

flag_1	flag_2	Положение блока.
True	True	В строке.
True	False	В начале строки.
False	True	В конце строки.
False	False	Не в строке.

В принципе в одну XY-область могут попадать несколько блоков. Например, для символа '?' или '!'.

Если два блока, расположенные в одном столбце находятся в одной строке, то такие два блока объединяются в один. Строка начинается и заканчивается в одной Y-позиции.

В большинстве строк чаще встречаются строчные буквы. Маленькие буквы располагаются по другой линии, отличной от линии прописных букв.

Необходимо заранее отделить заглавные буквы от прописных т.к. алгоритм распознавания не может различать большие и маленькие символы, так, например, буквы 'X' и 'x'.

Для построения двух линий проведем два вектора из правых углов блока в левые углы правого соседнего. Нижний вектор должен быть почти горизонтальный, а Y-компонента верхнего - почти половина высоты строки.

Если данное условие не выполняется с некоторой погрешностью, то данная пара векторов игнорируется.

Проверка этого условия необходима, т.к. кроме прописных и строчных букв могут встречаться, например, кавычки или знаки препинания

Рассмотрев множество "хороших" векторов, можно определить среднюю высоту строчных букв.

В дальнейшем данная линия позволяет определить, является ли буква строчной или прописной.

Для поиска пробелов в строке достаточно определить среднее расстояние между блоками вдоль линии прописных букв. Если расстояние заметно больше среднего, то между двумя символами стоит пробел. Полученные блоки готовы к распознаванию символов.

Распознавание символов. Для распознавания символов используется алгоритм WSA. На вход распознавателя подается растр с изображением символа, который приводится к фиксированному размеру. В данной реализации - 16x16 пикселей.

Над изображением проводится несколько параллельных прямых под определенным углом. В данной реализации проводится 16 прямых. Также рассматривается 128 значений углов в диапазоне от 0 до 180 градусов. Для каждого угла поворота и каждой прямой вычисляется число точек, попадающих на данную прямую

Для каждой прямой вычисляется количество черных пикселей, попавших на прямую. На одной прямой может находиться от 0 до 24 пикселей. Таким образом мы вычисляем 128x16 чисел, лежащих в диапазоне [0..24].

Если данное множество значений рассмотреть как 128-мерное пространство векторов, то все выученные символы можно представить как векторы в этом пространстве.

Процесс выбора символа представляет собой поиск вектора, расстояние до которого наименьшее. Если расстояние мало, то символ считается хорошо распознанным.

В противном случае система просит пользователя подтвердить правильность распознавания.

Проверка полученных слов при помощи словаря. Данная реализация СОР проверяет распознанные слова в словаре, который представляет собой текстовый файл с возможными словами. В случае отсутствия распознанного слова система просит пользователя подтвердить правильность и в случае необходимости вносит исправления в словарь.

Литература:

1.Джураев М.К., Жуманов И.И. Основные принципы и методы контроля орфографических ошибок естественных языков. В сб. «Вопросы Кибернетики» № 169, Ташкент, 2004 г., с. 32-38.

2.Джураев М.К., Жуманов И.И. Программная система контроля орфографических ошибок в текстах на узбекском языке. В матер. Межд.научно-практ. Конф. «Инновации и информационные технологии –2004». Ташкент, ТГТУ, 20-22 октября 2004 года, Ташкент, с. 272-273.

Научный руководитель, д.т.н., профессор И.И.Жуманов