Искусственный интеллект. Интеллектуальные системы. Материалы международной научно-технической конференции 24-29 сентября 2007. Донецк-Таганрог-Минск.

УДК 004.934.1'1

В.Ю. Шелепов
Институт проблем искусственного интеллекта
Мин образования и науки и НАН Украины
shel@iai.donetsk.ua

Концепция пофонемного распознавания отдельно произносимых слов русской речи. Распознавание синтаксически связных фраз

Организация записи речевого сигнала, определение его начала и конца

За основу принимается 8-битная оцифровка звукового сигнала с частотой дискретизации 22050 Гц, так что его значения имеют $2^8 = 256$ градаций: от 0 до 255.

Предполагается использование системы в лабораторных условиях, при отсутствии существенного внешнего шума. При настройке системы записи по нажатии соответствующей кнопки записывается 30000 отсчетов «тишины» и в записанном сигнале анализируются последовательные отрезки по 256 отсчетов в каждом. Для каждого из них вычисляется отношение

$$V / C, \tag{1}$$

где

$$V = \sum_{i=1}^{256} \left| x_{i+1} - x_i \right| \tag{2}$$

- численный аналог полной вариации, C - число точек постоянства, то есть моментов времени, для которых в следующий момент величина сигнала остается неизменной. Автоматически определяется значение величины (1), характерное для используемой звуковой карты, как наиболее часто встречающееся в массиве значений. Оно увеличивается на 0,1 и заносится в управляющий файл $recorder.ini$ под именем $StartPorog$, а результат, увеличенный в 10 раз, – под именем $EndPorog$.

При распознавании речи по нажатии кнопки записи компьютер начинает записывать сигнал, поступающий с микрофона, и вычислять для последовательных отрезков по 256 отсчетов величину (1), Определяется момент, после которого эта величина впервые не менее пяти раз подряд превышает $StartPorog$ и, начиная с него, в буфер1 заносится 30000 отсчетов, после чего запись останавливается. Далее выполняется аналогичная операция в обратном направлении от конца к началу, определяется момент, когда величина (1) не менее пяти раз подряд превышает $EndPorog$ и проставляется

метка конца речи. Отрезок от начала буфера1 до метки заносится в буфер 2. Поскольку превышение *StartPorog* может произойти не за счет появления речи, а за счет случайного шума, содержимое буфера2 анализируется на наличие речи путем вычисления последовательности квазипериодов. Если найдется не менее 5 идущих подряд квазипериодов, значения которых превышают заданную пороговую величину, то содержимое буфера 2 считается речью и передается в окончательный буфер3, как объект для визуализации и дальнейшего распознавания. Упомянутый порог определяется высотой голоса диктора, для тенора его можно взять (при используемой частоте дискретизации) равным 100.

Транскриптор

Очевидно, что пофонемное распознавание слов, список которых задан в текстовом виде, требует предварительного создания транскрипции каждого из этих слов. Эта процедура, разумеется, должна осуществляться автоматически. В отличие от синтеза речи, когда транскрипция должна быть максимально подробной, при распознавании допустима и даже желательна упрощенная транскрипция. Мы используем простой транскриптор, который работает по принципу замены букв и их комбинаций транскрипционными символами. В качестве транскрипционных знаков использованы в основном соответствующие русские буквы, исключения: t –мягкое Т, l – мягкое Л, h – ягкое Х, j - обозначает согласный типа того, что возникает первым при произнесении названий букв «е», «ё», «ю», «я». Правила замены представлены списком в управляющем файле. Пример:

сд=зд

- озвончение глухой согласной перед звонкой. Управляющий файл дополняется файлом исключений. Пример:

кого=каво

Структурная классификация русских слов

Разобьем все участвующие в транскрипциях фонемы на несколько естественных классов:

1) аеёоэя ; 2) иуыю ; 3) бвгд ; 4) jлlмн ; 5) жз ; 6) р ; 7) пкт-t ; 8) сшщцчh ; 9) фх; (3)

Первый – компактные гласные, второй – диффузные гласные, третий – взрывные голосовые согласные плюс «в», четвертый – основные сонорные согласные плюс j, пятый – голосовые согласные с шумом, шестой – «р», седьмой - то, что при произношении выступает как пауза в слове, восьмой - шипящие, аффрикаты и мягкое «х», девятый – глухие протяженные «ф» и «х».

Пусть далее есть большой словарь, который размечаем, сопоставив каждому слову его транскрипцию и заменив каждый транскрипционный символ номером его класса. При этом, если цифра повторяется подряд несколько раз, заменяем ее одной. Вот отрезок размеченного таким образом словаря Зализняка:

машина 418241; машинальность 418241418; машинальный 41824142; машинизация 4182425182; машинизировать 418242526131; машинизироваться 418242526131782; машинист 4182428; машинистка 418242871;

Про слова с одинаковой разметкой будем говорить, что они имеют одинаковую структуру. Таким образом, структура – это некая модель чередования гласных, согласных, шипящих и т. д. Оказывается, что в русском языке слов с одинаковой структурой относительно мало. Вот, например, все слова со структурой 314251:

долежать 314251, долезать 314251, долизать 314251, донизать 314251

На почти 100-тысячный словарь Зализняка максимальное число слов с одинаковой структурой (171) равно106, то есть около 0,1 процента. Причем это фактически исключительный случай. Все остальные структуры содержат единицы, или несколько десятков слов. Это доказано нами с помощью программы, которая автоматически делает разметку и выбор слов с одинаковой структурой, при этом выбор классов (3) можно менять.

Алгоритмы сегментации речевого сигнала

Вначале определяются участки сигнала, соответствующие шипящим и паузам (см. следующий пункт). Затем остальные участки сигнала разбиваются на отрезки по 256 отсчетов, и на каждом из них вычисляется значение вариации (2). Далее от начала участка последовательно берется 20 таких отрезков (или менее, столько, сколько позволяет длина участка) и вычисляется среднее значение соответствующих величин - порог. Тем отрезкам, для которых величина больше среднего присваивается символ «В» (выше порога), остальным символ «Н» (ниже порога). Чтобы устранить случайные единичные включения для каждого i-го элемента полученной символьной последовательности S выполняется обработка тройками:

если $s[i\text{-}1] = s[i+1]$ и $s[i] \neq s[i-1]$, то полагается

$$s[i] = s[i-1]; \qquad (4)$$

и обработка «четверками»:

если $s[i] = s[i+3]$ и $s[i+1] \neq s[i]$, $s[i+2] \neq s[i]$, то полагается

$$s[i+1] = s[i] \quad \text{и} \quad s[i+2] = s[i]. \qquad (5)$$

Затем интервал, на котором выполняются описанная процедура, сдвигается вправо на одно окно и процедура повторяется. Это происходит до тех пор, пока упомянутый интервал находится в пределах сигнала. В результате возникает таблица, состоящая из строк символов В и Н.

Далее просматриваются все строки полученой таблицы и создается новая символьная последовательность S1. Если текущая *i*-я строка таблицы начинается и заканчивается одним и тем же символом («Н» или «В»), то в S1 на *i*-ю позицию записывается соответствующий символ. Иначе считается количество вхождений каждого из символов в данной строке. Если количество «В» превышает количество «Н» или равно ему, то в S1 на соответствующую позицию записывается «В», иначе «Н». К полученой последовательности применяется обработка (1) и (2). Метки сегментации ставятся там, где происходит смена символов «Н» на «В», или «В» на «Н».

Способ выделения шипящих и пауз, не использующий фильтрацию

Предлагается m раз последовательно обработать сигнал трехточечным сглаживающим фильтром

$$y_i = \frac{y_{i-1} + y_i + y_{i+1}}{3},$$

взяв в качестве m минимальное число, при котором участок шипящей превращается в прямую (для автора этих строк и используемого им микрофона m =25). После этого для записанного речевого сигнала формируется массив величин (1), вычисляемых для последовательности окон по 256 отсчетов. Для этого массива осуществляется «В-Н»-обработка с порогом 0,1 а также обработка «тройками» (4) и «четверками» (5). Начало и конец «Н»-участка отмечаются метками. Они являются концами соответствующей шипящей или паузы.

Организация системы распознавания пары фонем или пары классов фонем

Мы исходим из представления, что фонема и слово – это акустически принципиально разные фонетические объекты. Фонема (и даже класс близких фонем) – объект спектрально сравнительно однородный, слово же, напротив, состоит из спектрально разнородных частей. Поэтому, распознавая слова целиком, мы должны использовать тот или иной *вектор* признаков. Для распознавания же фонем (и их классов) более целесообразно использовать подходящий скалярный признак или набор независимых скалярных признаков, каждый из которых должен обеспечивать свой результат распознавания. Как правило, на основе нескольких примеров можно указать интервалы, куда чаще всего попадают значения признака для каждого члена рассматриваемой пары классов или фонем. Значения за пределами этих интервалов разумно интерпретировать как отказ от распознавания.

Итак, создавая обучаемую систему распознавания пары классов, использующую один скалярный признак X , задаем два числа a, b. При

$$X < a \tag{6}$$

считаем, что объект распознавания принадлежит первому классу, при

$$X > b \tag{7}$$

- второму. При

$$a < X < b$$

не выполняется ни (6), ни (7) и мы имеем отказ от распознавания.

Вначале задаются достаточно малое инициальное значение a и достаточно большое инициальное значение b. Если, пользуясь ими при распознавании, система не определит объект первого класса, то число a слишком мало. После того, как пользователь укажет истинный результат, система должна заменить a вычисленным значением признака, увеличив последнее скажем на 0,1. Таким образом, в процессе обучения число a может только расти. Аналогично число b может только убывать. При этом обеспечивается все большая надежность в случае принятия решения. Если, начиная с некоторого момента, окажется

$$a > b, \qquad (8)$$

то при попадании X в (b,a) для распознаваемого объекта выполняются оба неравенства (6) и (7), то есть он должен быть отнесен к обоим классам сразу, что невозможно, так как предполагается, что класс должен определяться однозначно. Таким образом, в случае (8) попадание X в (b,a) должно означать отказ от распознавания. Суммируя сказанное, получаем, что при

$$X < \min(a.b)$$

объект относится к первому классу, при

$$X > \max(a,b)$$

- ко второму классу, при

$$\min(a,b) \le X \le \max(a,b)$$

система отказывается от распознавания. Обучение состоит в модификации констант a,b и продолжается до тех пор, пока система не проработает без ошибок на протяжении, скажем, пяти циклов распознавания. Тогда распознаватель будет либо с высокой надежностью принимать правильное решение либо отказываться от распознавания. Теперь представим себе, что для полученной системы вероятность отказа достаточно мала. Если мы для той же пары введем еще несколько таких систем, использующих другие признаки, то ввиду схемы независимых испытаний Бернулли, вероятность того, что все они одновременно будут отказываться от распознавания, станет существенно меньше. Все вместе построенные системы дадут желаемый распознаватель для рассматриваемой пары классов, если случай противоречия в результатах отдельных систем мы также будем интерпретировать как отказ от распознавания.

Из других подходов наилучшие результаты на сегодняшний день даёт использование нейросетей и признаков, построенных на основе вейвлет-преобразований.

Поиск в большом словаре по смешанной транскрипции

В идеале результатом распознавания фонем, образующих слово, служит его транскрипция, по которой слово в большинстве случаев однозначно восстанавливается.

Однако, как ранее отмечалось, любые признаки, используемые при распознавании речи, имеют характер случайных величин. Поэтому на любом этапе возможен отказ от распознавания и в результате вместо цепочки транскрипционных знаков на выходе получится последовательность символов, обозначающих те или иные достаточно широкие классы фонем. Ее можно рассматривать как результат смешения транскрипций разного уровня детализации. Возникает проблема, как по такому разнородному результату в большом словаре отыскать слова, которые ему удовлетворяют.

Для решения этой проблемы был разработан алгоритм поиска заданного слова в словаре, позволяющий получить информацию о наличии или отсутствии слова за n операций сравнения символов (n- длина искомого слова), что соответствует **одной** операции сравнения строк. Такой выигрыш в производительности стал возможен благодаря древовидной структуре представления словаря в памяти ЭВМ (в отличие от традиционной структуры, подобной списку). Представить словарь можно, например, при помощи дерева, каждый узел которого имеет столько потомков, сколько символов содержится в алфавите языка словаря.

Распознавание синтаксически связных фраз

В отличие от английского русский язык относится к числу так называемых флективных языков. Большинство слов помимо начальной или словарной формы, также называемой «леммой», имеют достаточно развитую систему косвенных форм, образуемых с помощью флексий – частей, изменяемых при склонении, спряжении и т. д. Правильное использование этих форм есть непременное условие синтаксически связной русской речи. Наличие многочисленных косвенных форм создает дополнительные трудности при компьютерном распознавании русской речи, ибо каждая из косвенных форм является для компьютера новым словом, в результате чего резко возрастает объем распознаваемого словаря. Различие же между отдельными формами зачастую сводится к безударным гласным в окончании, различать которые на сегодняшний день при обычной речи не представляется возможным. Последнее связано с редукцией упомянутых безударных гласных.

Имея дело с технической системой (хотя бы и относящейся к искусственному интеллекту), каковой является система компьютерного распознавания речи, мы вправе предложить пользователю соблюдение некоторых правил, относящихся к самой речи. Например, можно на первых порах настаивать на подчеркнутой артикуляции, когда при слитном произнесении слова в нем слегка подчеркивается слоговая структура, так что каждая гласная становится как бы ударной.

Далее, может быть предложена специальная архитектура системы, которая наряду с распознаванием речи использует элементы выбора из небольших словарей. Опишем одну из таких возможных систем. Распознаватель работает с первоначальным списком, содержащим все словоформы слов известного словаря Зализняка [1]. Получается достаточно полный словарь **всего русского языка**. Запись сказанного слова происходит при нажатии клавиши, отвечающей его первой букве, так что первая фонема при распознавании фактически заранее задается. Далее можно даже не заниматься точным распознаванием фонем, а ограничится разбиением сигнала, соответствующего отдельно сказанному слову, на участки, отвечающие отдельным фонемам с одновременной классификацией этих участков по принципу «ГЛАСНАЯ - ГОЛОСОВАЯ СОГЛАСНАЯ – ШИПЯЩАЯ - ПАУЗА». В результате после распознавания мы получаем цепочку символов, соответствующих этим наиболее общим классам (обобщенная транскрипция). Применение системы поиска, представляющей словарь в виде дерева, позволяет очень быстро и эффективно получить список слов, имеющих данную обобщенную транскрипцию. В нашем случае получаемый список будет содержать от нескольких слов до нескольких сотен слов. Это некоторый набор словоформ. Далее используется наличие быстрого лемматизатора (система восстанавливающая начальную форму слова по косвенной). Применяя его к полученному списку, мы получим соответствующий набор начальных форм, который в несколько раз короче ранее полученного списка словоформ. По указанию пользователя (например, щелчок мыши) по исходному звуковому файлу строится эталон и ему сопоставляется соответствующая лемма. Тогда, как показывает опыт, в подавляющем большинстве случаев при распознавании с использованием алгоритма DTW любая словоформа этого слова будет отождествляться с указанной леммой. Исключения составляют ситуации типа «ИДТИ-ШЕЛ», «ЧЕЛОВЕК-ЛЮДИ». Наконец, отдельным списком выводятся те словоформы исходного списка, которые соответствуют данной лемме. Нужную из них пользователь передает в набираемый текст. Подчеркнем, что эталон строится по произвольной произнесенной словоформе, получает имя соответствующей леммы и по нему распознаются (с точностью до леммы) все другие словоформы этого слова.

Из изложенного вытекает, что при пофонемном распознавании синтаксически связных фраз успех достигается путем использования дополнительных модулей работы с текстовыми списками: лемматизатора и модуля образования парадигм. Путь к

сокращению получаемых по ходу дела списков – в более детальном распознавании фонем. Наилучшие результаты в этом отношении на сегодняшний день даёт использование нейросетей и признаков, построенных на основе вейвлет-преобразований.