УДК 004.934

В.Ю. Шелепов, А.В. Ниценко
Институт проблем искусственного интеллекта, г. Донецк, Украина

Амплитудная сегментация речевого сигнала, использующая фильтрацию и известный фонетический состав

В статье описывается новый условный сегментатор. Он использует априорную информацию о фонемном составе сигнала, но это не является препятствием для его применения в пофонемном распознавателе. Особенностью нового сегментатора является то, что он, в отличие от других, не имеет тенденции ставить лишние метки и совершенно правильно сегментирует большое количество слов. Очень важно, что сегментатор способен правильно разделять два подряд идущих гласных или два подряд идущих согласных, а также выделять звук «р». При этом основные функции выполняются без предварительной подстройки под диктора. Работа является развитием одного из подходов, предложенных в [1].

Общее описание сегментации

Сегментирующие метки в слове расставляются одна за другой слева направо. Для нахождения каждой следующей метки осуществляется процедура разделения двух фонем, которые начинаются с последней найденной метки. При этом участок, соответствующий этим двум фонемам, мы задаем априори приближенно, считая его равным 256×19 отсчетам.

Алгоритм поиска границ шипящих и глухих согласных звуков

На каждом этапе работы системы происходит анализ транскрипции речевого сигнала для выяснения его фонетического состава. При наличии в транскрипции шипящих (к которым в данной работе относятся [ш], [с], [ф], [х], [ц], [ч], [щ]), глухих взрывных согласных [п], [к], [т], [t], которые ассоциируются с паузой в сигнале (здесь *t* означает мягкое «т»), либо фонем [ж], [з] участок, который должен содержать одну из указанных и соседнюю с ней фонему, обрабатывается фильтром низких частот с частотой среза 500 Гц. Это уменьшает энергию участка, соответствующего каждой из перечисленных фонем. Далее происходит разделение сигнала на части с высокой и низкой амплитудой (энергией).

Для этого вычисляется средняя амплитуда всего отфильтрованного участка:

$$E_{cp} = \sqrt{\frac{\sum_{i=1}^{n} \left| x_i - 127 \right|^2}{n}},$$

где x_i – значение i-го отсчёта участка, n – количество отсчетов (длина участка). Далее отфильтрованный участок разбивается на отрезки по 256 отсчетов и вычисляется амплитуда на каждом отрезке. В результате получаем массив значений амплитуды $\{E_i\}$. Чтобы определить границы между высокоамплитудной и низкоамплитудной частями сигнала, последовательно сравниваем каждое значение E_i с пороговой величиной T. Для шипящих и пауз полагаем $T = 0.3 \cdot E_{cp}$, для [ж] и [з] полагаем $T=E_{cp}$. Если $E_{i-1}>T$ и $E_i<T$, то, возможно, значение $256 \cdot i$ – это граница между высоко- и низкоамплитудной частями участка. Чтобы избежать появления лишних границ, дополнительно проверяется расстояние до предыдущей границы. Если оно не превышает 512 отсчетов, то значение $256 \cdot i$ пропускаем, иначе считаем значение $256 \cdot i$ очередной границей и запоминаем его вместе с предыдущей границей в списке сегментов Segment_Array. Участок с этими границами считается высокоамплитудным и соответствующим очередному голосовому звуку в транскрипции (либо нескольким голосовым звукам).

Если $E_{i-1}<T$ и $E_i>T$, то, возможно, $256 \cdot i$ – граница между низко- и высокоамплитудными частями участка (при этом выполняется такая же проверка, что и в предыдущем случае). Участок с соответствующими границами считается низкоамплитудным и отвечающим очередному шипящему звуку, паузе либо [ж], [з] в транскрипции (либо сочетанию «шипящий – пауза»).

При работе алгоритма наличие транскрипции слова дает возможность контролировать правильность чередования высокоамплитудных и низкоамплитудных участков (т.е. если в транскрипции за шипящей следует голосовой звук, то алгоритм будет искать только переход от низкой амплитуды к высокой и наоборот). Это позволяет избежать появления лишних меток и повышает точность сегментации.

Алгоритм разделения шипящих и глухих согласных

Если в результате работы предыдущего алгоритма в списке сегментов присутствуют участки с низкой энергией, соответствующие сочетанию шипящих и глухих согласных звуков [п], [к], [т], [t], то для того, чтобы определить границу между ними, используется следующий алгоритм.

Соответствующий участок исходного сигнала обрабатывается фильтром высоких частот с частотой среза 1500 Гц, и на нем строится массив значений амплитуды $\{E_i\}$ (от левой границы до правой). Чтобы определить границу между шипящей и паузой, последовательно сравниваем каждое значение E_i с пороговой величиной, равной E_{cp}. Если $E_{i-1} >E_{cp}$ и $E_i<E_{cp}$, то, возможно, значение $256 \cdot i$ – это искомая граница. Чтобы избежать появления лишних границ, дополнительно проверяется расстояние от левой границы всего участка до найденной границы. Если оно не превышает 512 отсчетов, то значение $256 \cdot i$ пропускаем, иначе считаем значение $256 \cdot i$ очередной границей и запоминаем это значение вместе с левой

границей участка в списке сегментов Segment_Array. Участок с этими границами считается высокоамплитудным и соответствующим шипящей, а следующий за ним низкоамплитудный участок – паузе. В список сегментов добавляются два новых участка.

Алгоритм поиска границ гласных и звонких согласных звуков

Если после работы предыдущих алгоритмов в списке сегментов присутствуют участки с высокой амплитудой, соответствующие сочетаниям подряд идущих гласных и звонких согласных звуков, для их разделения применяется фильтрация соответствующих участков сигнала фильтром высоких частот. Такая фильтрация сильно понижает энергию участков, соответствующих звонким согласным звукам, в то время как энергия гласных уменьшается незначительно.

Если участок содержит звук «и», то он обрабатывается фильтром высоких частот (ФВЧ) с частотой среза 3500 Гц (за исключением пар «ил» и «lи»). Для сочетаний «вы», «ил», «lи», «лу», «ду», «ву», «бу», «гу» используется ФВЧ с частотой среза 250 Гц, для сочетания «му» – частота среза 750 Гц. В случае остальных сочетаний участок обрабатывается фильтром высоких частот с частотой среза 500 Гц. Далее строится массив значений амплитуды $\{E_i\}$ на рассматриваемом участке сигнала (от левой границы до правой). Чтобы определить границы между высокоамплитудными и низкоамплитудными участками сигнала, последовательно сравниваем каждое значение E_i с пороговой величиной T. Если согласная есть одна из фонем [б], [д], [г], то $T = 0.5 \cdot E_{cp}$, иначе $T = E_{cp}$. Если $E_{i-1} > T$ и $E_i < T$, то, возможно, значение $256 \cdot i$ – это граница между высоко- и низкоамплитудным участками сигнала. Чтобы избежать появления лишних границ, дополнительно проверяется расстояние до предыдущей границы. Если оно не превышает 600 отсчетов, то пропускаем значение $256 \cdot i$, иначе считаем значение $256 \cdot i$ очередной границей и запоминаем его вместе с предыдущей границей в списке сегментов Segment_Array. Участок с этими границами считается высокоамплитудным и соответствующим очередному гласному звуку в транскрипции (либо нескольким гласным звукам). К гласным относятся [а], [о], [у], [е], [ы], [и], [э], [ю], [я].

Если $E_{i-1} < T$ и $E_i > T$, то, возможно, $256 \cdot i$ – граница между низко- и высокоамплитудным участками сигнала (при этом выполняется такая же проверка, как и в предыдущем случае). В случае положительного результата первый участок считается низкоамплитудным и соответствует очередному звонкому согласному звуку в транскрипции ([б], [в], [г], [д], [л], [l], [м], [н]) (здесь l обозначает мягкое «л»). Работа алгоритма завершится, когда будут просмотрены все значения E_i.

Поиск границы между двумя гласными или между двумя согласными

Если в транскрипции есть сочетания из двух гласных, то соответствующий участок вначале выделяется с помощью вышеописанных методов целиком как высокоамплитудный, а затем обрабатывается ФВЧ и делится на низкоамплитуд-

ную и высокоамплитудную части. Если в транскрипции есть сочетания из двух согласных, то соответствующий участок сигнала вначале выделяется целиком как низкоамплитудный, а затем обрабатывается ФВЧ и делится на низкоамплитудную и высокоамплитудную части. При этом для различных сочетаний используются следующие частоты среза ФВЧ (подчеркнуты фонемы, которые при обработке сочетания считаются высокоамплитудными) (табл. 1).

Таблица 1

Сочетание	Частота среза	Сочетание	Частота среза
А О	1000	ВЗ	1500
А У	1000	ВЛ	250
А И	500	ВМ	0
ОЭ	1500	ВJ	2500
ОУ	1500	ВL	1500
ОИ	500	ЖЛ	1500
УИ	4000	ЖМ	2500
ЭУ	1000	ЖJ	2500
ЭИ	500	ЖL	4000
БВ, ГВ, ДВ	250	ЗЛ	2500
БЖ, ГЖ, ДЖ	1500	ЗМ	4500
БЗ, ГЗ, ДЗ	1500	ЗJ	4500
БЛ, ГЛ, ДЛ	250	ЗL	2500
БМ, ГМ, ДМ	0	ЛМ	500
БJ, ГJ, ДJ	2500	МJ	2500
БL, ГL, ДL	2500	LJ	2500
ВЖ	1500		

Если сочетание встречается в обратном порядке, то частота среза прежняя. Нулевая частота среза соответствует случаю отсутствия фильтрации.

В результате работы всех алгоритмов на выходе получаем список участков сигнала, соответствующих символам транскрипции.

Выделение звука «р»

Механизм образования русского «р» коренным образом отличается от механизма образования всех других звонких фонем. Этот звук возникает за счет ударов языка по нёбу. При этом число таких ударов, обычно равное одному или двум, в случае подчеркнутого раскатистого «р» может доходить до четырех. На рис. 1 показано амплитудно-временное представление слова «ура», автоматически просегментированное нашей программой.

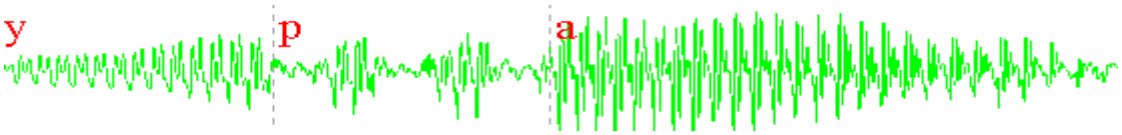

Рисунок 1

Сигнал не подвергается какой-либо фильтрации. При обработке первой пары «ур» участок, где находится граница между «у» и «р», классифицируется как низкоамплитудный. Аналогично обрабатывается вторая пара «ра». Метки ставятся в начале первого и в конце последнего низкоамплитудного участка. При разделении «р» и соседней звонкой согласной применяется предварительная обработка соответствующего участка фильтром высоких частот с частотой среза 500 Гц и участок, соответствующий звуку «р», интерпретируется как высокоамплитудный. На рис. 2 показан пример результата автоматической сегментации для слова «собрать» (безударное «о» транскрибируется как «а»).

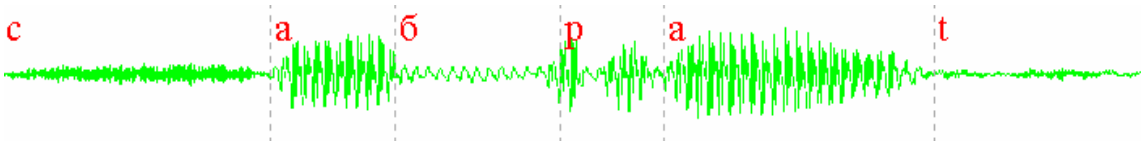

Рисунок 2

Ввиду того, что мягкое «р» в русском языке всегда короткое, программа априори отводит для соответствующего участка длину не превышающую 256×3 отсчетов. Приведем пример результата автоматической сегментации слова «дари» (рис. 3).

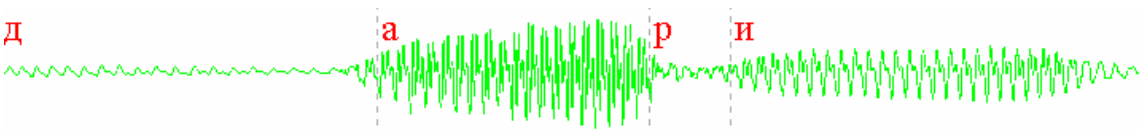

Рисунок 3

В заключение отметим следующее. Если задать машине транскрипцию длинного слова, а вместо этого произнести короткое, то в результате сегментации будет выделено участков меньше, чем символов в транскрипции. Результат «сегментации» слова «сон» при задании транскрипции слова «машинист» показан на рис. 4.

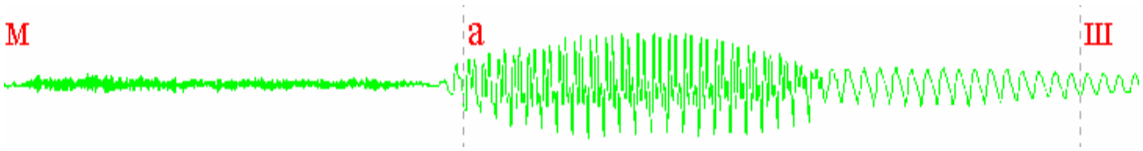

Рисунок 4

Если же задана транскрипция короткого слова, а произнесено длинное, то последний выделенный участок окажется недопустимо длинным. Результат «сегментации» слова «машинист» при задании транскрипции слова «сон» показан на рис. 5.

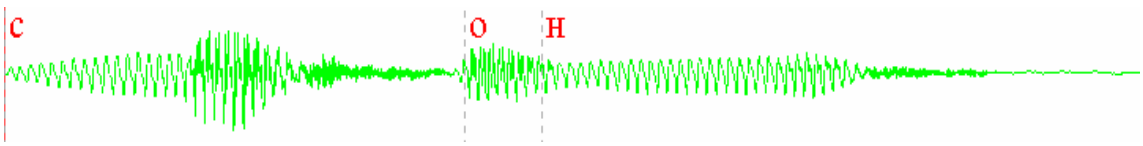

Рисунок 5

В этих и подобных случаях наша программа выводит сообщение «Сказанное не соответствует транскрипции!». Если теперь произнести любое слово некоторого заданного словаря и сказанное последовательно обработать сегментатором на основании всех имеющихся транскрипций, то слова, для которых результат сегментации не соответствует транскрипции, можно не включать в число кандидатов на распознавание. Эксперимент показывает, что практически для любого реального словаря число кандидатов на распознавание при этом сокращается не менее чем наполовину, а иногда и до одного единственного слова. Польза такого «частичного распознавания» на основе одной лишь сегментации (без распознавания фонем) очевидна.

Литература

1. Дорохин О.А., Старушко Д.Г., Федоров Е.Е., Шелепов В.Ю. Сегментация речевого сигнала // Искусственный интеллект. – 2000. – № 3. – С. 450-458.

The article describes new conditional segmentator on the basis of filtration and the subsequent amplitude processing of a speech signal. Segmentator uses the aprioristic information about phoneme signal structure but it is not an obstacle for its application in phoneme-by-phoneme recognizer. The feature of a new segmentator is that it, as against others, has no tendency to put superfluous labels and perfectly segments a plenty of words. It is very important, that segmentator is capable to divide correctly two successively going vowels or two successively going consonants and also to allocate a sound “p”. Thus, the basic functions are carried out without preliminary retraining under the announcer. The work develops one of the approaches suggested in [1].

У статті описується новий умовний сегментатор. Він використовує апріорну інформацію щодо фонемного складу сигналу, але це не є перешкодою для його застосування в пофонемному розпізнавачі. Особливістю нового сегментатора є те, що він, на відміну від інших, не має тенденції ставити зайві мітки й абсолютно правильно сегментує велику кількість слів. Дуже важливо, що сегментатор здатен правильно розділяти два голосних, що йдуть підряд, або два приголосних, що йдуть підряд, а також виділяти звук «р». При цьому основні функції виконуються без попереднього підстроювання під диктора. Робота є розвитком одного з підходів, запропонованих у [1].

Статья поступила в редакцию 01.08.03.