УДК 004.934.1’1

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

Новые алгоритмы распознавания фонем и их классов, поиск слова по его смешанной транскрипции при распознавании слов большого словаря

Алгоритмы, о которых идет речь, реализованы в виде программы, позволяющей распознавать слова из определенного набора, содержащего тысячи слов, без предварительного создания для них голосовых эталонов (пофонемное распознавание). А именно, мы ограничиваемся задачей распознавания слов, которые содержат лишь твердые согласные (за исключением *Т* мягкого в конце слова) и удовлетворяют условию строгого чередования гласных и согласных звуков.

Целью работы является описание системы распознавания отдельно произносимых слов большого словаря, организованной в виде бинарного дерева распознавания фонем и их классов. В результате распознавания получается цепочка символов, относящихся к транскрипциям различного уровня. Предлагается метод быстрого поиска слова по такой смешенной транскрипции.

Организация системы распознавания пары фонем или пары классов фонем

Рассмотрим следующую классификацию звуков русского языка:

qаоэуы W
qаоэ Q
аоэ A
оэ O
уы U
бвгдлмнжзр C
бвгдлмнр R
бвгдлмн V
бвгд G
бгд D
лмн L
мн M
жз Z
сшщцчфхпкт-t Y
сшщцчt F
сцt S

шщчt H
фхпкт-tцч P

Здесь малыми буквами обозначены используемые нами транскрипционные знаки (см. [6]). При этом латинские символы q и t обозначают безударное *A* и мягкое *T* соответственно. В конце каждой строки стоит символ, обозначающий класс звуков, предшествующих ему в строке. Легко видеть, что ряд классов связан отношением включения, что соответствует более детальной или менее детальной классификации звуков. Звук t может распознаваться и как фрикативный (классы F,S,H), и как пауза (класс P).

Мы считаем, что запись сказанного диктором слова отсегментирована, то есть, разбита на участки, отвечающие отдельным фонемам. В рамках рассматриваемого множества фонем классы W, C и Y являются наиболее общими. W содержит все допустимые гласные, C – все допустимые голосовые согласные, Y – фрикативные, аффрикаты паузы и паузообразные звуки. Соответствующие участки сигнала выделяются и идентифицируются при сегментации и на предшествующих ей этапах обработки, которые описаны выше. То же происходит с фонемой *P*. Поэтому распознаванию подлежат следующие пары, образованные классами и отдельными фонемами:

Q/U, а/O, о/э, у/ы, V/Z, G/L, в/D, л/M, ж/з, F/P, S/H, ш/щ, ф/х.

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. Речь – сложная система, которая в целом является вполне детерминированной. В то же время общеизвестно, что любой мыслимый признак, который можно использовать при ее распознавании, является случайной величиной. В этом нет противоречия, подобные вещи происходят и в физике, когда при хаотическом движении отдельных молекул для их большого числа вырабатываются детерминированные макрохарактеристики такие, как температура, давление и так далее. Итак, каждый признак следует рассматривать как случайную величину со своей функцией распределения, которая определяется конкретным диктором, конкретным микрофоном, конкретной звуковой картой. Последние два момента являются определяющими. Какой смысл пытаться делать распознаватель, инвариантный по отношению к диктору, если он зависит от микрофона? Мы считаем, что до тех пор, пока вопросы, связанные с независимостью от аппаратного обеспечения не решены, целесообразнее разрабатывать быстро обучаемые системы распознавания речи с подстройкой под диктора. Аккуратное описание функции распределения – серьезная задача, требующая большого статистического материала. В виду крайне ограниченной сферы применения (конкретный диктор, микрофон и так далее), более или менее точное описание функции распределения становится не целесообразным. В то же время, 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

$$X < a \tag{8}$$

считаем, что объект распознавания принадлежит первому классу, при

$$X > b \tag{9}$$

- второму. При

$$a < X < b$$

не выполняется ни (8), ни (9) 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

$$a > b, \tag{10}$$

то при попадании X в (b, a) для распознаваемого объекта выполняются оба неравенства (8) и (9), то есть он должен быть отнесен к обоим классам сразу, что невозможно, так как предполагается, что класс должен определяться однозначно. Таким образом, в случае (3) 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

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

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 отказывается от распознавания. Теперь представим себе, что для полученной системы вероятность отказа от распознавания 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.

Для дальнейшего введем следующие величины, которые мы вычисляем на окнах длины 256 отсчетов:

$E-$ среднее отклонение (см. (5));

E_{a-b} - среднее отклонение для сигнала, обработанного фильтром с полосой пропускания от a Гц до b Гц;

V - вариация сигнала (см. (4));

V_a - вариация сигнала после a - кратного сглаживания;

C - количество точек постоянства, то есть моментов времени, для которых в следующий момент величина сигнала остается той же самой (см. [4]),

C_{a-b} - количество точек постоянства для сигнала, обработанного фильтром с полосой пропускания от a Гц до b Гц;

C_a - количество точек постоянства после a - кратного сглаживания;

N - количество точек непостоянства;

S_{a-b} - сумма длин полных колебаний длины от a до b;

(Длиной полного колебания мы называем число отсчетов между соседними локальными максимумами. Более точное определение см. в [5]).

Для произвольного участка сигнала будем обозначать теми же буквами результат усреднения указанных величин по всем последовательным 256-окнам, содержащимся в выделенном участке.

Признаки для распознавания гласных фонем и их классов

Признаки для распознавания пары E/Q

$V_{600-800} \cdot E_{600-800}$,

V / C,

V,

$\max(E_{400-600}, E_{600-800}) - E_{200-400}$.

-(W_{17}) , W – вектор коэффициентов вейвлет-разложения Морле.

Признаки для распознавания пары U/I

$(E_{3400-3600} + ... + E_{4800-5000}) / E$,

$(V_{3400-3600} + ... + V_{4800-5000}) / V$,

Количество нулевых компонент в векторе $(E_{0-200,....,} E_{4800-5000})$ при округлении до 2-х знаков после запятой,

$E_{600-800} / E$,

-($\sum_{i=1}^{22} W_i$),

-(W_5).

Признаки для распознавания пары о/э

Признаки для распознавания пары А/О (О – классфонем о,э)

$(E_{600-800} - E_{400-600}) / E$,

$(V_{600-800} - V_{400-600}) / V$,

-(W_{17}).

Признаки для распознавания пары о/э

$V_{600-800} / V$,

$E_{600-800} / E$,

$(S_{10} + S_{11}) / (S_{19} + S_{20})$.

Признаки для распознавания пары О/J

-(W_{18}).

Признаки для распознавания голосовых согласных и их классов

Признаки для распознавания пары V/Z

$$\frac{V^2}{V_5 \cdot 100} \cdot (E_{3800\text{-}4000} + ... + E_{4800–5000}) / E \ ,$$

N_{max} -максимальное количество переходов от возрастания к убыванию на квазипериоде

$$\frac{V}{C},$$

$$(E_{3200-3400} + ... + E_{4800-5000}) / E \ ,$$

V разности исходного и сглаженного сигналов,

-(W_5).

Признаки для распознавания пары G/L

$$E \cdot V \ ,$$

$$V_{800\text{-}1000} / V \ ,$$

$V_{800\text{-}1000}$,

$$\frac{V}{C},$$

V после 50-кратного сглаживания.

-(W_{21})

Признаки для распознавания пары в/Д

Количество нулевых компонент в векторе S,

$V_{800\text{-}1000} / V$,

$E_{800\text{-}1000} / E$.

Признаки для распознавания пары л/М

$N_{\max}$,

$100 \cdot KLP[10]$ (второй множитель – 10-я компонента вектора коэффициентов линейного предсказания)

$100 \cdot KLP[2]$

Признаки для распознавания пары з/ж

$(E_{1200-1400} + ... + E_{2600-2800}) / E$,

$(V_{1800-2000} + ... + V_{2800-3000}) / V$,

$(E_{4000-4200} + ... + E_{4800-5000}) / E$,

$(V_{4000-4200} + ... + V_{4800-5000}) / V$,

$(V_{4000-4200} + ... + V_{4400-4600}) / V$.

Признаки распознавания шипящих и пауз (классов F и P)

Признаки для распознавания пары F/P

$K[1]$ -первая компонента вектора кепстральных коэффициентов,

C,

$\frac{V}{C}$.

Признаки для распознавания пары S/H

$V_{2200-2400} / V$,

$(E_{1200-1400} + ... + E_{2600-2800}) / E$,

$(V_{1800-2000} + ... + V_{2800-3000}) / V$,

$K[2]$.

Признак для распознавания пары ш/щ

$E_{1200-1400} / E + V_{1200-1400} / V + V_{1400-1600} / V$.

Признак для распознавания пары ф/х

S_2

Признак для распознавания пары S/X

$V_{600-800} / V$

$V_{600-800} / V - V_{4400-4600} / V$

Признак для распознавания пары P/X

Количество нулевых компонент в векторе S,

V / E

Описание дерева поиска в большом словаре слова по результату распознавания классов составляющих его фонем (использование смешанных транскрипций различного уровня)

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. Ниже описан алгоритм, который позволяет сделать это и сделать очень быстро.

Существующие на данный момент алгоритмы позволяют найти требуемое слово или установить его отсутствие в словаре из N слов за N операций сравнения строк, если словарь неупорядочен, или за $\log_2 N$ операций сравнения строк, если словарь упорядочен. Для некоторых задач такое время поиска в достаточно большом словаре (10000 слов и более) является неприемлемым. Так, в задаче распознавания отдельно произнесённых слов возникает необходимость поиска в словаре всех слов, соответствующих некоторой обобщённой транскрипции, полученной в результате обработки звука. Такая транскрипция при достаточной длине слова может давать до нескольких сотен тысяч возможных вариантов слов, причём лишь ничтожная часть этих вариантов содержится в словаре. Таким образом, для данной задачи количество операций сравнения строк для получения списка слов, соответствующих построенной обобщённой транскрипции и содержащихся в словаре, составляет

$M \cdot \log_2 N$,

где M - количество возможных вариантов слов, соответствующих обобщённой транскрипции.

Для преодоления 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). Более того, этот же подход оказывается полезным и в случае поиска слов по обобщённой транскрипции. Обход дерева с подстановкой различных вариантов написания слова по обобщённой транскрипции на каждом уровне дерева (т. е. генерация вариантов написания слова по обобщенной транскрипции в контексте исходного словаря) сокращает значение M на несколько порядков.

Представить словарь можно, например, при помощи дерева, каждый узел которого имеет столько потомков, сколько символов содержится в алфавите языка словаря (обозначим его через K). Структура такого узла приведена на рисунке 1.1.

символ		признак окончания слова	
Потомок 1	Потомок 2	…	Потомок K

Рисунок 1- Структура узла с фиксированным количеством потомков

Каждый уровень дерева соответствует позиции символа в слове. Каждый узел в рамках каждого уровня представляет собой символ в слове на соответствующей позиции. Максимальная глубина дерева соответствует максимальной длине слова в словаре. Поиск слова в словаре в данном случае соответствует поиску пути в дереве, содержащем на каждом уровне соответствующий символ искомого слова. Учитывая, что количество потомков на каждом уровне дерева фиксировано (и требуемый символ-потомок искать не нужно), весь поиск можно осуществить за n операций индексирования в массиве потомков. Таким образом, сложность поиска оказывается равной длине искомого слова.

Алгоритм поиска слова в словаре, представленном в виде дерева, может быть записан следующим образом (на примере узла U i-го уровня):

1) если $i = длина(\mathrm{S})$ то
   a. если $U.[признак\,окончания\,слова] = True$ то слово найдено, переход на 2
      иначе слово не найдено, переход на 2

иначе
   a. вычисляем индекс потомка j
   b. если Потомок j не существует, то слово не найдено, переход на 2
      иначе выбираем Потомка j для дальнейшего поиска
      $$U := Потомок\ j,$$
      наращиваем i:

$i := i + 1;$

   c. переход на 1)
2) конец алгоритма

Другими словами, считается, что слово есть в словаре, если можно построить путь от вершины, в котором содержатся все символы данного слова в правильном порядке, и в последнем узле которого установлен признак окончания слова.

Очевидно, что полностью заполненное дерево заданной глубины L представляет собой всевозможные сочетания символов алфавита языка словаря длины L. Также очевидно, что не все такие сочетания будут входить в какой-либо словарь. Получить не полностью заполненное дерево можно за один проход исходного словаря.

Алгоритм заполнения дерева можно представить следующим образом (рассмотрим на примере узла U i-го уровня):

1) если $i = длина(\mathrm{S})$, то
   a. устанавливаем признак окончания слова:
      $$U.[признак\,окончания\,слова] := True;$$
   b. переход на 7,

   иначе вычисляем индекс потомка j
2) если Потомок j не существует, то создаём его,
   иначе переход на 4;
3) устанавливаем значение символа Потомка j:

$Потомок\ j.[с.[симв := S[i+1]$;

4) наращиваем i:

$i := i+1$;

5) выбираем Потомка j для дальнейших преобразований:

$U := Потомок\ j$;

6) переход на 1;

7) конец алгоритма.

Другими словами, остаток слова (без первого символа) рекурсивно добавляется в поддерево, вершина которого соответствует первому символу остатка (если такое поддерево не существует, то оно создаётся).

Однако в каждом узле по-прежнему будет содержаться K указателей на возможных потомков. Поэтому более оптимальной с точки зрения использования памяти будет структура узла с переменным количеством потомков, приведенная на рисунке 2. Здесь K_r - реальное количество потомков данного узла, $K_r \le K$.

символ			признак окончания слова	
K_r	Потомок 1	Потомок 2	…	Потомок K_r

Рисунок 2- Структура узла с переменным количеством потомков

В данном случае выигрыш в объёме используемой памяти приводит к небольшому падению скорости поиска, поскольку вместо обычного индексирования приходится выполнять поиск нужного символа в отсортированном списке потомков. В наихудшем случае количество операций сравнения при таком поиске составит $\log_2 K$. Поэтому максимальная сложность поиска равна $n \cdot \log_2 K$. Сложность по-прежнему оказывается пропорциональной длине слова n, а не размеру словаря N.

Данная древовидная структура представления словаря позволяет осуществлять не только оптимальный поиск конкретных слов, но и получение всех слов в словаре, соответствующих заданному шаблону. Для этого в процедуру поиска слова в словаре необходимо на каждом уровне дерева добавить возможность генерации необходимых вариантов, если в качестве искомого символа в слове содержится знак шаблона. Такой поуровневый анализ позволяет пропускать на следующий уровень только те варианты, уже рассмотренная часть которых точно содержится в словаре. Таким образом, если словарь содержит M_r слов, подходящих под шаблон, а также более длинных слов, включающих слова, подходящие под шаблон, то на поиск слов, подходящих под шаблон, будет в худшем случае затрачено $M_r \cdot n \cdot \log_2 K$.

Выводы.

По нашему мнению предложенный механизм распознавания отдельно произносимых слов, использующий отказ от распознавания некоторых фонем (так что вместо них распознаются некоторые более общие классы) является адекватным,
ввиду того, что фонемы являются достаточно мелкими фонетическими единицами.

Перечень ссылок

1. Дорохин О.А., Старушко Д.Г., Федоров Е.Е., Шелепов В.Ю. Сегментация речевого сигнала // Искусственный интеллект. – 2000. - №3. – С.450-458.3.

2. Шелепов В.Ю., Ниценко А.В. Амплитудная сегментация речевого сигнала, использующая фильтрацию и известный фонетический состав // Искусственный интеллект. – 2003. - №3. – С.421-426.

3. Ниценко А.В., Шелепов В.Ю. Алгоритмы пофонемного распознавания слов наперед заданного словаря // Искусственный интеллект. – 2004. - №. – С.633-639.
4. В.Ю.Шелепов, А.В. Ниценко. К проблеме пофонемного распознавания // Искусственный интеллект. – 2005. - №4. – С.662-668.
5. Засыпкин А.В., Мицевич А.Т., Овецкий М.В., Шелепов В.Ю. О дикторонезависимой системе голосового телефонного номеронабирателя // Труды международной конференции “Знание-Диалог-Решение”.-Ялта.-1995.-С. 427-430.
6. Структурная классификация слов русского языка. Новые алгоритмы сегментации речевого сигнала, распознавания некоторых классов фонем// Искусственный интеллект. – 2007. - №1.- 213-224.

В.Ю. Шелепов, А. В. Ніценко, А.В. Жук. Нові алгоритми розпізнавання фонем та їх класів, пошук слова за його змішаної транскрипції при розпізнаванні слів великого словника

Алгоритми цієї роботи є реалізованими як програма, яка дозволяє розпізнавати тисячі слів без попереднього створення голосових еталонів (пофонемне розпізнавання). Ми обмежуємось задачею розпізнавання слів, яки містять тільки тверді приголосні (окрім м’якого *Т* у кінці слів) та задовольняють умові строгого чергування голосних та приголосних.

V.Ju. Shelepov, A.V. Nicenko, A.V. Zhuk. New recognition algorithms of phonemes and their classes, searching of word using mixed transcription when words of large vocabularies are recognizing.

Algorithms of this article are realized like the program recognizing thousands words without a priory creation of voice templates (phoneme recognition). We restrict ourselves with recognition of words which contain only hard consonants (except soft *T* in the end of the word) and strict interchange of vowels and consonants.