Один подход к автоматическому распознаванию речи

В.И.Галунов и Г.В.Галунов
AudiTech ltd. Санкт-Петербург
auditech@online.ru

Введение

Постановка задачи

В этой заметке описывается подход к решению задачи машинного перевода устной речи основанный на следующих ограничения в постановке задачи:

1. Говорящий должен придерживаться контекста в том смысле, что в каждый отдельный момент можно говорить только на заранее определённую и вполне конкретную тему.
2. Раздельное произнесение слов, что означает наличие пауз между словами не менее 150 мс.
3. Отсутствие шумов сравнимых по амплитуде с распознаваемой речью.

Подход к решению задачи

Для решения задачи мы использовали подход основанный на следующих принципах:

1. Модульность, т.е. выделение в задаче блоков взаимодействующих через максимально простой интерфейс (что конечно же не новость в программировании, но тем не менее далеко не общепринятая вещь в распознавании речи, где более привычен метод сбора системы на колене, без каких-либо изначальных ограничений на структуру программы).
2. Дешевизна, которая вполне естественна для системы разрабатываемой на территории СНГ, и прагматизм, т.е. использование тех решений которые работают и не использование тех которые работают хуже вне зависимости от того соответствуют ли они нашим теоретическим представлениям или нет.
3. Предпочтение новых решений стандартным в силу исследовательского интереса, стремления к патентной чистоте изделия и в силу пункта 2.

Модули системы

Этих модулей получилось пять, причём их взаимодействие предельно просто из-за ограничений описанных выше:

1. «Семантический» - определяющий смысл фразы. В нашем случае в этом модуле нет ничего искусственно - интеллектуального ибо он лишь относит произнесённую фразу к одному из небольшого (100-150) количества «смыслов». Смысл здесь - это множество фраз, которые можно перевести на иностранный язык одинаково.
2. «Аналитический» - определяющий для произнесённого слова вероятность того, что оно есть слово из списка преданного ему семантическим модулем, точнее для каждого слова из этого списка он определяет вероятность того, что произнесённое слово есть данное. Максимальное возможное число слов в списке - 350.
3. Вторичное описание, т.е. представление слова в виде последовательности акустических событий. В данной реализации это не более чем результат векторного квантования последовательности векторов признаков, т.е. акустическое событие - это кластер в пространстве признаков.
4. Первичное описание - т.е. представление слова в виде последовательности векторов признаков. Вообще говоря в системе обычно использовалось два и более разных первичных описания, каждому из которых соответствовало своё вторичное описание и соответственно аналитическому блоку приходилось выбирать какое из описаний лучше сработало.
5. Разбиение фразы на слова, которое оказалось возможным в виде отдельного модуля не взаимодействующего интерактивно с остальными ввиду раздельности произнесения.

Подробнее о модулях нашей системы

1.Семантика

В работе "семантического" можно выделить два уровня:

- прагматика,
- синтаксическая семантика (семантика на уровне предложения).

Прагматика, вообще говоря, занимается не языковой информацией, а “внешней картиной мира”. То есть теми знаниями о объектах действительности, их взаимосвязях, которые есть у каждого взрослого человека. В данном случае прагматика заключена в сценариях диалога, набранных вручную. Например для ситуации «гостиница» имеется две роли: администратор (портье) и клиент У портье есть свой набор фраз - смыслов (около 90), у клиента - свой (около 100). Вся ситуация «гостиница» разделена на 3 подситуации: поселение (бронирование), проживание, отъезд. Для каждой подситуации создан свой сценарий диалога. Переход из одной подситуации в другую происходит искусственно (с помощью соответствующих кнопок). Идея диалога в данной модели состоит в следующем: когда клиент (или портье) начинает разговор, он может сказать что-то из достаточно ограниченного списка, его собеседник в ответ на произнесенную фразу может сказать еще меньше. В ответ на эту фразу также возможно произнести не очень много фраз и т. д. То есть, в зависимости от ситуации, определяемой предыдущей фразой, имеется достаточно ограниченный выбор фраз, которые можно сказать в ответ.

Синтаксическая семантика руководит определением смысла фразы по вероятностям слов, которые были получены на этапе распознавания слов.

Смысл фразы представлен целостно, нерасчленённо, за ним просто закрепляется название в виде предложения на русском или другом языке. Причем названия на разных языках за исключением небольшого кол-ва случаев ссылаются на один и тот же смысл. Сначала на уровне прагматики определяется список нулевых фраз, уместных в данной ситуации. В дальнейшем распознавание строится как выбор лучшей фразы из этого списка. Определяются ключевые слова, которые могут встретиться в этих фразах.

Эти слова единым списком передаются «аналитическому» модулю. Этот модуль для каждого реально произнесенного слова во фразе выдает последовательность вероятностей этих слов. Определяется уверенность в распознанности каждого произнесенного слова и тем самым вероятность того, что реально произнесенное слово отсутствует в подававшемся списке слов (и вообще в словаре). В результате получается матрица вероятностей слов. Распознавание фразы происходит в несколько примерно однородных этапов. Матрица вероятностей слов подается на собственно семантический модуль. В результате его работы получается последовательность фраз, (выбранных из списка фраз, полученного на прагматическом уровне), упорядоченных по убыванию вероятностей. Из этой последовательности отбирается несколько фраз, идущих первыми (т. е. лучшие), и в дальнейшем проверяются только они (начинается второй этап). Снова определяются ключевые слова, необходимые для распознавания только этих фраз, (их становится меньше по сравнению с первоначальным списком слов, и, кроме того, для каждого реально произнесенного слова во фразе получается в общем случае свой список). Далее эти списки слов подаются на модуль распознавания слов. И все повторяется до тех пор, пока по определенным критериям не нужно будет остановиться. Например это может случиться, когда семантический модуль вернет одну единственную фразу или когда кол-во выдаваемых им фраз перестает существенно уменьшаться.

Работа собственно семантического модуля заключается в следующем. Для каждого персонажа и каждого языка существует матрица, содержащие по несколько тысяч строк. Каждая строка содержит последовательность слов, которая теоретически может быть произнесена клиентом, когда он имеет в виду какую-либо из предусмотренных фраз. Клиент может произносить такую последовательность слов, в которой встречаются слова, не содержащиеся в словаре, не ключевые. Для этих слов предусмотрен специальный символ (-1). Отдельно существует структура данных, организующая соответствие этих строк определенным нулевым фразам. В среднем на одну нулевую фразу приходится несколько сотен таких строк. Процесс выбора нулевой фразы (определение смысла произнесенной фразы) происходит следующим образом. Для каждой фразы из списка, полученного на прагматическом уровне, просматриваются все строки матрицы, соответствующие ей. Затем для каждой такой строки из матрицы вероятностей слов выбираются вероятности тех слов, которые присутствуют в этой строке, и считается вероятность этой строки (последовательности слов). Эти вероятности запоминаются, а после выбирается наибольшая из них. Так определяется лучшая гипотеза для данной нулевой фразы. То же повторяется для остальных нулевых фраз и лучшие гипотезы упорядочиваются по убыванию вероятностей.

2.Аналитика

Меры близости

Как уже было сказано выше задача аналитического уровня это установить с какой вероятностью произнесённое пользователем слово соответствует каждому слову из заданного списка. При этом слово представлено в терминах вторичного описания т.е. как последовательность целых чисел - номеров состояний. В таком же виде представлена и обучающая выборка. (Ограничения наложенные на систему делают пословное распознавание предпочтительным.) Количество произнесений каждого слова в обучающей выборке было сравнительно небольшим (от 15 до 40, включая словоформы), что не позволило применить сколько-нибудь масштабные статистические процедуры типа цепей Маркова. Таким образом обучающая выборка использовалась в основном «как она есть», т.е. тестируемое слово сравнивалось с каждым словом из обучающей выборки. Конечно некоторые статистические процедуры всё же применялись, например предварительно для каждого слова была подсчитана средняя длина и из обучающей выборки были выброшены представители слова вопиющим образом не соответствующие своей средней длине. Далее, для каждого слова была подсчитана «кластерная вероятность» - вероятность для каждого состояния встретиться в данном слове. Вероятность, с точки зрения «кластерной вероятности» для тестируемого произнесения оказаться представителем данного слова использовалась в аналитическом модуле как одна из мер близости. Она считалась как произведение по всем состояниям тестируемого произнесения вероятностей, того что данное состояние встретится в данном слове. Вс е остальные меры близости, используемые в нашей системе - это варианты широко известного “динамического программирования” (ДП). Пусть D(a,b) - расстояние между a-м и b-м состояниями, а x_i $1 \le i \le m$ и y_j $1 \le j \le n$ две финитных последовательности натуральных чисел, тогда под ДП понимается нахождение пути т.е. последовательностей i_k и j_k таких, что

$$1 \le i_k \le m,\ 1 \le j_k \le n,\ i_{k+1}=i_k \le i_k+1,\ j_k=j_k \le j_k+1 \quad (1)$$

и

$$|j_k - i_k\, n/m| < p, \quad (2)$$

где p фиксированное целое число (которое часто называют (полу-)шириной коридора) минимизирующего сумму

$$\sum_k D(x(i_k),y(j_k)) \quad (3).$$

Мы использовали три варианта ДП:

- ДП с полным перебором, т.е. вариант с полным перебором всех путей удовлетворяющих (1) и (2). Существенным недостатком этого метода является относительная его вычислительная ёмкость.
- Смесь ДП с градиентным спуском, т.е. вариант когда строится локально оптимальный путь - из трёх вариантов перехода от k к k+1, удовлетворяющих (1): вниз, направо и по диагонали выбирается тот для которого $D(x(i_{k+1}),y(j_{k+1}))$ минимально, разумеется если он удовлетворяет условию (2). Это ДП примерно в 8 раз быстрее первого но на 10 процентов хуже его, в том смысле что на словаре в 350 слов проверяемое слово в 85 процентах случаев оказывалось ближе к своим аналогам из базы данных при использовании полного ДП, и лишь в 75 процентах при использовании ДП с градиентным спуском.
- Вариант ДП аналогичный предыдущему с тем лишь отличием что анализируются варианты развития удовлетворяющие условиям (1) и (2) но не на один шаг вперёд, а на два. Этот вариант ДП в 3 раза быстрее полного перебора и лишь на пол процента хуже.

Наиболее приятной особенностью трёх описанных нами алгоритмов является то, что они ошибаются в разных случаях, что мы и попытались использовать, применяя их совместно.

Пока, что мы не затронули вопрос о том откуда взялась матрица D(i,j). Вопрос о выборе наилучшем этой матрицы можно рассматривать как отдельную задачу оптимизации, но в силу её достаточно большой размерности матрицы эта задача не имеет тривиального решения. Мы использовали две естественные матрицы : геометрическую - когда D(i,j) есть расстояние между центрами соответствующих кластеров

$$D(i,j)=\mathrm{dist}(z_i,z_j) \quad (4)$$

и вероятностную, когда D(i,j) есть величина обратная сумме вероятностей переходов из состояния i в состояние j и наоборот из состояния j в состояние i

$$D(i,j)=(P(i \to j)+P(j \to i))^{-1} \quad (5)$$

(Общий объём речевого материала в обучающей выборке был достаточно велик для того, чтобы подсчитать эти вероятности довольно точно.) Несмотря на то, что матрица (5) кажется более соответствующей решаемой задаче, т.к. инкапсулирует информацию о совместной встречаемости в речи разных состояний, она несколько уступает матрице (4). Было установлено, что оба эти варианта являются локально оптимальными, т.е. их небольшие случайно выбранные возмущения ухудшают распознавание по ДП.

Их применение

Разрабатывая схему применения описанных выше мер близости мы преследовали две цели

- из влечь максимум информации, т.е. максимально использовать их (незначительную) нескоррелированность и
- оптимальным образом использовать вычислительные ресурсы (вписаться в P III 500).

Для каждого из вторичных описаний мы имеем 14 мер близости: минимальное расстояние и среднее расстояние в смысле каждого из трёх ДП по обеим матрицам, кластерная вероятность и разница длин. Минимальное расстояние здесь это минимум ДП расстояний от тестируемого слова до всех представителей какого-либо слова, аналогично - среднее.

Вычислительная оптимальность диктует применение мер близости в порядке замедления, т.е. начиная с самых быстрых и заканчивая самыми медленными. Фактически применение мер близости было следующим: пусть δ_i i-я по быстроте мера близости (длина - самая быстрая, затем кластерная вероятность, второе ДП, третье ДП, первое ДП), пусть словарь насчитывает N_0 слов и δ_i i=1,...,14 последовательность чисел такая что $\delta_1+...+\delta_{14}=\delta$, где δ достаточно маленькое число характеризующее суммарную ошибку, которую мы планируем допустить. Найдём числа $N_1>N_2>...>N_{14}$ обладающие тем свойством, что если мы сравним тестируемое слово со всеми словами из обучающей выборки по мере φ_1 и выстроим получившиеся N чисел в порядке возрастания то вероятность того что число соответствующее тому слову, которое было сказано окажется дальше N_1-го места меньше или равна δ_1, далее, сравним тестируемое произнесение по мере φ_2 с оставшимися словами и выделим из них N_2 наилучших и т.д. Получившаяся конструкция из сужающихся списков позволяет естественным образом определить окончательную вероятность P_a того, что произнесённое слово - это a-е слово. Например: изначально припишем всем словам вес $1/N_0$, затем к весам слов попавшим в список из N_1 штук прибавим $1/N_1$ и т.д. , результат отнормируем на единицу. Если вспомнить, что семантический уровень запрашивает результаты сравнения не со всем словарём, а лишь с некоторой выборкой, то нам останется сделать ещё один подсчёт просуммировать вероятности слов из словаря не вошедших в запрашиваемую выборку и назвать результат «вероятностью мусора».

Итого: для каждого из вторичных описаний у нас есть способ сосчитать вероятность того, что тестируемое слово есть реализация каждого из слов обучающей выборки. Осталось согласовать результаты даваемые разными реализациями. Вполне естественным кажется следующий метод: пусть H_1 -нормированная на единицу энтропия первого описания

$$H=\sum_{1\le i\le n} p_i\log_n(p_i) \quad (6),$$

где n - число слов в запрашиваемом списке, а H_2 - нормированная энтропия второго описания, тогда возможны три варианта:

- одно из описаний почти однозначно т.е. его H меньше ξ - нижнего порога (где-то 0.2 - 0.4) - тогда используется это описание
- если для обоих описаний H больше верхнего порога η где-то 0.7-0.9) то - это слово считается «мусором», т.е. словом не из списка.
- в остальных случаях окончательная вероятность находится как взвешенная сумма вероятностей описаний:

$$p_i=p_{1i}/H_1+p_{2i}/H_2 \quad (7),$$

3.Вторичное описание

Вторичное описание - это видимо модуль в котором из первичного описания извлекается информация о наличии (или отсутствии) в данном месте речевого сигнала каких-то акустических событий. В описываемой системе, по крайней мере в текущей её реализации вторичное описание есть не более чем векторное квантование пространства признаков. Исторически оно (квантование) возникло на месте неудачной попытки построить «экспертное» описание акустических событий.
Практических оправданий квантования два:

- экономия места - вместо вещественного вектора хранится одно целое число (точнее unsigned char)
- улучшение результатов распознавания. Как это не странно процент дикторонезависимого распознавания на ДП после квантования в два с половиной раза выше чем при ДП распознавании «сырых» признаков. Можно сказать, что квантование позволяет избавиться от лишней информации описывающей индивидуальные особенности диктора. Эти $2^1/_2$ раза относятся к специфической процедуре векторного квантования, описанной ниже.

Векторное квантование

В векторном квантование можно выделить две подзадачи:

- собственно векторное квантование т.е. сопоставление заданному вектору вектора из фиксированного конечного набора (кодовой книги). Эта задача в нашем случае тривиальна - есть расстояние

$$dist(x,y)=\sum_i \alpha_i |x_i-y_i|, \quad (8)$$

где x_i и y_i координаты векторов x и y, и входному вектору сопоставляется ближайший к нему по этому расстоянию представитель кодовой книги.

- построение кодовой книги. Оно было нами модифицировано, а именно к алгоритму K- средних или самоорганизующиеся карте Кохонена, (в зависимости от личных пристрастий) была добавлена адоптация коэффициентов α_i из формулы (8).

Суть этой адоптации состоит в том, что изначально в качестве α_i берётся величина обратная среднему расстоянию по i-ой координате:

$$<|x_i-y_i|>\alpha_i = 1. \quad (9)$$

После того как кодовая книга построена из неё выбрасываются кластеры представляющие наименьшее количество векторов (допустим меньшее чем один процент от количества векторов в наибольшем кластере). После этого α_i пересчитываются по формуле (9), но вместо векторов x,y из обучающей выборки подставляются уже центры кластеров. И так до тех пор пока кодовая книга не достигнет желаемого размера или не остановится в росте.
Таким образом результатом построения кодовой книги оказывается также и расстояние (8), которое ранее встретилось в формуле (4).

4.Первичное описание

Главным критерием которым мы руководствовались при подборе признаков была их «графическая выразительность», т.е. то, что график каждого из этих признаков даёт эксперту возможность выделить какой-либо сорт акустических событий из потока речи. К сожалению ни один из стандартных наборов признаков таких как LPC,LSP,MFCC и т.д. такой особенностью не обладает, хотя в этом плане MFCC наиболее привлекателен. Нами было подобрано несколько разных наборов признаков приблизительно эквивалентных друг другу в смысле эффективности распознавания, и видимо нет необходимости останавливаться на них подробно.

2.Разделение речевого сигнала на слова

Задачей этого блока является создание простого и помехоустойчивого алгоритма для автоматического определения границ слов при их раздельном произнесении с небольшими паузами (порядка 0.15 с.). Основные трудности решения поставленной задачи обусловлены изменчивостью произнесения и разнообразием используемого словаря, наличием специфических пауз перед смычками внутри слов, воздействием нестационарного шума. Были предложены и опробованы разные алгоритмы, но наиболее эффективным оказался следующий:
«сочетание трехканальной фильтрации и двухпороговой процедуры поиска моментов начала и окончания слов с итеративным уточнением результатов». Этот алгоритм состоит в том, что для каждой

из трёх равномерных полос спектра сначала ищутся точки в которой сигнал начинает превышать порог вычисленный из среднего шума в канале, а потом точки в которой амплитуда в полосе окажется ниже порога вычисленного из среднего значение в предполагаемом слове. Далее применяется несложная логика относительно взаимного расположения этих точек.