более сложные функции и таким образом удешевляя для компании эту службу.

Другой чрезвычайно актуальной для финансовых систем областью применения речевых технологий является аутентификация пользователей с помощью отпечатков голоса. Эта технология способна эффективно дополнить средства обеспечения безопасности и контроля доступа, а в некоторых случаях заменить принятую сейчас систему запоминания идентификаторов и паролей.

ЗАКЛЮЧЕНИЕ

В настоящее время растет популярность электронных платежных систем. Стратегия банков и платежных систем заключается в том, чтобы привлечь как можно больше клиентов путем обеспечения своей надежности и предоставления клиентам как можно большего числа услуг и удобных интерфейсов. Речевые технологии являются как раз тем инструментом, который создает новый удобный интерфейс и одновременно может быть использован как средство дополнительной безопасности.

ЛИТЕРАТУРА

1. *Генкин А. С.* Планета Web-денег. — Альпина Паблишер, 2003. — 510 с.
2. *Экслер А.* WebMoney: — М.: Экспромт; СПб.: Геликон Плюс, 2003. — 312 с.
3. http://www.webmoney.ru
4. http://www.money.yandex.ru
5. http://www.telepat.ru

☎ *(495) 334-90-60*

E-mail: serena@ipu.ru

❑

УДК 51-74:519.2:519.87

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ РАСЧЕТА СРЕДНЕГО ЧИСЛА ПЕРЕСПРОСОВ ПРИ КОМПЬЮТЕРНОМ РАСПОЗНАВАНИИ РЕЧИ

М. П. Фархадов[1], А. В. Жожикашвили[2]

[1] *Институт проблем управления, г. Москва*

[2] *Институт проблем передачи информации, г. Москва*

Предложена модель для расчета числа переспросов при компьютерном распознавании речи. Описаны два алгоритма поведения компьютера и дана их сравнительная оценка

ВВЕДЕНИЕ

Проектирование систем массового обслуживания с компьютерным распознаванием речи связано с решением ряда оптимизационных задач. Одна из них заключается в построении диалога, обеспечивающего минимизацию времени обслуживания вызова. В настоящей статье рассмотрена математическая модель, позволяющая выбрать алгоритм поведения компьютера в зависимости от качества распознавания речи.

1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Пусть имеется n слов $s_1, s_2, ..., s_n$. Компьютер пытается распознать произнесенное слово и предлагает свой вариант. Возможны три ситуации, когда компьютер:

— правильно распознает слово;

— ошибочно распознает слово;

— не может распознать слово (будем считать соответствующее сообщение компьютера словом s_0).

Введем условные обозначения:

l — математическое ожидание числа переспросов;

t_j — вероятность того, что компьютер воспринял слово как s_j;

p_i — вероятность того, что было произнесено слово s_i;

q_{ij} — вероятность того, что слово s_i распознано как s_j;

r_{ij} — вероятность того, что передавалось слово s_i при условии, что компьютер принял его за слово s_j.

Назовем эти вероятности "обратными". В частности, q_{ii} — вероятность того, что слово распознано правильно, а q_{i0} — вероятность того, что слово не было распознано вообще. Естественно, что $\sum\limits_{i=1}^{n} p_i = 1$ и $\sum\limits_{j=1}^{n} q_{ij} = 1$ для всех i.

Экспериментально можно определить вероятности p_i и q_{ij}. Здесь и далее $i, j = 0, 1, 2, ..., n$.

Для вычисления обратных вероятностей r_{ij} заметим, что вероятность того, что передавалось слово s_i, а компьютер понял его как s_j, равна p_iq_{ij}. Вероятность $t_j = \sum\limits_{k=1}^{n} p_k q_{kj}$. По формуле условных вероятностей $r_{ij} = p_i q_{ij} / \sum\limits_{k=1}^{n} p_k q_{kj}$.

Эти вероятности тоже можно определить статистически из опыта.

Рассмотрим следующий вариант действий компьютера. Пусть он распознал слово как s_j. Он переспрашивает говорящего, правильно ли распознано слово. Если ответ "да", процедура заканчивается. Если ответ "нет", компьютер перебирает все вероятности r_{1j}, r_{2j}, ..., r_{nj}, кроме r_{jj}, и выбирает среди них наибольшую. Пусть это будет r_{ij}. Компьютер предполагает, что было произнесено слово s_i, ибо при условии, что было получено слово s_j, у слова s_i самая большая вероятность. Он опять спрашивает говорящего, было ли произнесено слово s_i. Если и это не подтверждается, компьютер ищет вторую по значению вероятность и предполагает, что было произнесено второе слово и т. д. Назовем эту процедуру алгоритмом A_1.

Рассчитаем среднее число переспросов, возникающих при таком алгоритме. Для упрощения формул и избежания двойных индексов, перенумеруем слова. Присвоим слову, которое было передано, по "мнению" компьютера, номер 1, слову, которое с наибольшей вероятностью могло быть ошибочно воспринято как слово s_1, — номер 2, следующему по значению вероятности слову — номер 3 и т. д. При такой нумерации имеем $r_{11} \geqslant r_{21} \geqslant ... \geqslant r_{n1}$ (первое из этих неравенств означает, что если компьютер распознал слово как s_1, то, скорее всего, и говорилось слово s_1. Компьютер, следовательно, будет предлагать говорящему для подтверждения слова в следующем порядке: s_1, s_2, s_3 и т. д. С вероятностью r_{11} будет один переспрос — s_1, ибо говорилось то самое слово s_1, которое компьютер "услышал". С вероятностью r_{21} будет два переспроса — s_1 и s_2, с вероятностью r_{31} будет три переспроса — s_1, s_2 и s_3 и т. д. Следовательно, математическое ожидание числа переспросов будет определяться по формуле $l = r_{11} + 2r_{21} + 3r_{31} + ... + nr_{n1}$.

Если компьютер не смог распознать слово, то в рамках алгоритма A_1 он ищет максимальную вероятность из r_{10}, r_{20}, ... r_{n0}, и, если это окажется, скажем, r_{i0}, предлагает слово s_i, как наиболее вероятное. Если говорящий не согласится, ищется следующая по значению вероятность и т. д. Если опять перенумеровать слова так, чтобы выполнялись неравенства $r_{10} \geqslant r_{20} \geqslant ... \geqslant r_{n0}$, то получим ту же формулу для математического ожидания среднего числа переспросов: $l = r_{10} + 2r_{20} + 3r_{30} + ... + nr_{n0}$.

Посчитанное среднее число переспросов будет своим для каждого распознанного слова (этого не видно из формулы, поскольку в нее не входит явно номер слова, услышанного компьютером, но от этого слова зависит перенумерация слов, а следовательно, и результат расчета). Интересно было бы усреднить это число по всем словам, но трудно представить, как записать полученную формулу, ибо для каждого слова будет своя нумерация.

Попробуем переписать все формулы, не пользуясь явной перенумерацией слов. Пусть компьютер услышал слово s_j. Упорядочим вероятности r_{1j}, r_{2j}, ... r_{nj} в порядке убывания. В результате получим последовательность $r_{m_{1j}j} \geqslant r_{m_{2j}j} \geqslant ... \geqslant r_{m_{nj}j}$. Таким образом, m_{1j} — номер слова, которое наиболее вероятно передавалось при условии, что компьютер услышал слово номер j (как уже говорилось, обычно $m_{1j} = j$, ибо, если компьютер услышал слово s_j, то, скорее всего, оно и было произнесено, впрочем, это не принципиально для дальнейших рассуждений); m_{2j} — номер слова, которому соответствует вторая по значению вероятность, опять же при условии, что компьютер услышал слово s_j, и т. д. Формула для среднего числа переспросов принимает вид:

$$l_j = r_{m_{1j}j} + 2r_{m_{2j}j} + 3r_{m_{3j}j} + ... + nr_{m_{nj}j} = \sum_{i=1}^{n} i r_{m_{ij}j}.$$

Теперь можно выписать формулу для среднего числа переспросов, учитывая, что вероятность t_j того, что компьютер воспринял слово как s_j, равна $\sum\limits_{i=1}^{n} p_i q_{ij}$:

$$l = \sum_{j=1}^{n} t_j l_{ij} = \sum_{j=1}^{n} \Bigl(\sum_{i=1}^{n} p_i q_{ij} \sum_{i=1}^{n} i r_{m_{ij}j} \Bigr).$$

ПРИМЕР РАСЧЕТА СРЕДНЕГО ЧИСЛА ПЕРЕСПРОСОВ

Рассмотрим случай, когда компьютер должен определить произносимые пользователем слова, обозначающие цифры от 0 до 9, т. е. 10 слов: "ноль", "один", ..., "девять". Считаем вероятности p_i равными между собой, т. е. $p_0 = p_1 = \ = p_9 = 0{,}1$. Значения вероятности q_{ij} — того, что слово i при распознавании было принято за слово j, были определены экспериментально и приведены в табл. 1.

Для расчета определяем вероятности $r_{ij} = p_i q_{ij} / \sum_{k=1}^{n} p_k q_{kj}$. Полученные значения приведены в табл. 2.

Теперь можно посчитать среднее число переспросов. Результаты расчетов собраны в табл. 3 (алгоритм A_1).

верхней строке содержится предположение, сделанное компьютером о сообщенном ему слове, в нижней — среднее число переспросов, т. е. если компьютер распознал переданное слово как s_i, то соответствующее этому слову число переспросов указано во второй строке.

Для сравнения приведем тот же расчет по более простому алгоритму A_2. Он состоит в следующем: распознав число, компьютер спрашивает, правильно ли он его распознал, и, если нет, просит произнести то же число снова. В этом случае среднее число переспросов при условии, что говорилось слово s_i, определяется по формуле $1/q_{ii}$. Получаем, следовательно, табл. 4.

В отличие от табл. 3, в верхней строке здесь записаны не предположения компьютером о переданных словах, а реально переданные слова.

Среднее число переспросов по алгоритму A_1 составит $l = 1{,}08$. Соответственно, по алгоритму A_2 величина $l = 1{,}072$. Также он действует, если слово не было распознано.

Схема диалога показана на рисунке. В процессе работы (или предварительного тестирования) на сервере формируются вероятностные матрицы, которые содер-

Таблица 1

Вероятности q_{ij}

i	j									
	0	1	2	3	4	5	6	7	8	9
0	0,98	0,02	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0,02	0	0,98	0	0	0	0	0	0	0
3	0	0,02	0	0,86	0	0	0	0	0	0,12
4	0	0	0	0	0,94	0	0	0	0	0,06
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0,18	0,76	0,04	0	0,02
7	0	0,04	0	0	0	0	0	0,96	0	0
8	0	0,04	0	0	0	0	0	0	0,94	0
9	0	0	0	0	0	0	0	0,04	0	0,96

Таблица 2

Обратные вероятности r_{ij}

i	j									
	0	1	2	3	4	5	6	7	8	9
0	0,98	0,02	0	0	0	0	0	0	0	0
1	0	0,88	0	0	0	0	0	0	0	0
2	0,02	0	1	0	0	0	0	0	0	0
3	0	0,02	0	1	0	0	0	0	0	0,1
4	0	0	0	0	1	0	0	0	0	0,05
5	0	0	0	0	0	0,85	0	0	0	0
6	0	0	0	0	0	0,15	1	0,04	0	0,02
7	0	0,03	0	0	0	0	0	0,92	0	0
8	0	0,05	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0,04	0	0,83

Таблица 3

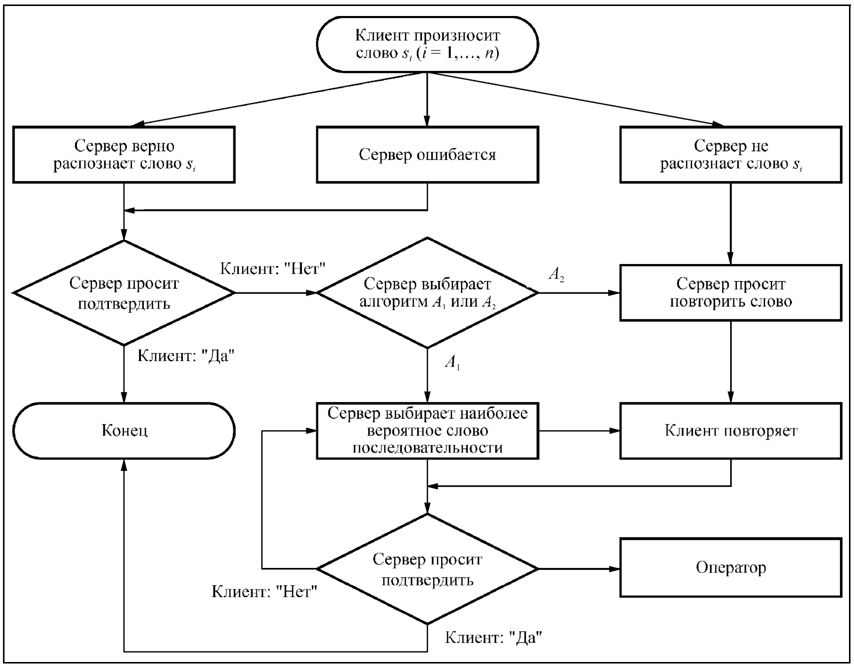
Среднее число переспросов по алгоритму A_1

i	0	1	2	3	4	5	6	7	8	9
l	1,02	1,25	1	1	1	1,15	1	1,12	1	1,26

Таблица 4

Среднее число переспросов по алгоритму A_2

i	0	1	2	3	4	5	6	7	8	9
l	1,02	1	1,02	1,16	1,06	1	1,32	1,04	1,06	1,04



Речевой диалог “клиент—сервер” при динамическом управлении поведением компьютера

жат вероятности p_i появления всех слов, вероятности p_{ij} всех переходов, включая p_{ii} (правильное распознавание) и вероятности нераспознавания p_{i0}, а также обратные вероятности r_{ij}. С помощью этих матриц сервер автоматически решает, какой из двух описанных алгоритмов A_1 или A_2 будет работать эффективней, и в процессе диалога использует этот алгоритм.

Если матрица вероятностей содержит слова с высокой вероятностью переходов p_{ij}, то в соответствии с алгоритмом A_1 эти слова последовательно произносятся сервером до тех пор, пока клиент не ответит “верно” или такие слова будут исчерпаны. В последнем случае, когда остаются слова с примерно равной и, следовательно, невысокой вероятностью переходов, сервер продолжит поиск по алгоритму A_2. Следует ожидать, что весь процесс потребует не более двух-трех переспросов. Например, выше на численном примере суммарное число переспросов оказалось равно ~2,15.

Чтобы правильно настраивать распознающую систему, необходимо собирать статистику поведения системы в процессе ее эксплуатации. В результате получим матрицы вероятностей, подобные приведенным в табл. 1 и 2. На практике такие матрицы обычно содержат так называемые “плохие” слова, т. е. слова, которые с большой вероятностью неправильно распознаются. Если система правильно спроектирована, число таких слов обычно не превышает 2—3 % от общего числа слов в системном словаре. Если в процессе диалога с клиентом система не распознает “плохое” слово с первого раза, она автоматически отыщет с помощью вероятностной матрицы это “плохое” слово и предложит его клиенту. Если слово не будет распознано и со второго раза, система обратится за помощью к оператору.

Описанная выше процедура соответствует алгоритму A_1. Разумеется, речь идет о диалоге, который не допускает длительных задержек. Если же процесс не налагает жестких ограничений на длительность задержек, можно повторить поиск слова несколько раз.

Другая ситуация возникает, если матрица вероятностей не содержит вероятностей появления “плохих” слов. В этом случае все слова с примерно одинаковой и небольшой вероятностью могут быть неправильно распознаны. В этой ситуации система не может отыскать наиболее вероятное слово, и алгоритм A_1 не работает. В соответствии с алгоритмом A_2 система просит клиента повторить произнесенное слово и с высокой вероятностью правильно его распознает.

Однако полностью исключить необходимость помощи со стороны оператора удается не всегда, ибо это потребовало бы чрезмерно большого времени.

ЗАКЛЮЧЕНИЕ

Применение математических моделей весьма полезно на этапе проектирования речевого человеко-машинного интерфейса. Расчеты, выполненные на основе таких моделей, позволят выбирать из многочисленных сценариев наиболее приемлемые и ограничивать тем самым число вариантов поведения компьютера, которые подлежат последующему трудоемкому тестированию.

☎ *(495) 334-87-10*
E-mail: mais@ipu.ru ❑