УДК 004.3

И.Ю. Бондаренко, ассистент,
Д.Г. Савкова, студент,
Донецкий национальный технический университет, г. Донецк, Украина
bond005@yandex.ru, das.savkova@gmail.com

# Исследование системы Sphinx4 для решения задач однодикторного и дикторонезависимого распознавания речевых команд

*Данное исследование посвящено разработке системы распознавания речевых команд управления компьютерными системами на базе инструментальной среды Sphinx 4. Проанализирован математический аппарат скрытых Марковских моделей, реализованный в Sphinx 4 для организации процесса распознавания речи и обучения такому распознаванию. Проведены экспериментальные исследования системы распознавания речевых команд на базе Sphinx 4, направленные на оценку точности работы этой системы как в однодикторном, так и в дикторонезависимом режимах. Сделаны практические рекомендации по эффективности применения инструментальной среды Sphinx 4 для организации речевого интерфейса компьютерных систем.*

***Ключевые слова: автоматическое распознавание речи, скрытые Марковские модели, речевой интерфейс, Sphinx***

## *Введение*

Информационные технологии занимают уникальное положение в современном обществе. В отличие от других научно-технических достижений, средства вычислительной техники и информатики применяются практически во всех сферах интеллектуальной деятельности человека, способствуя прогрессу в технике и технологиях. В последнее время особое внимание как исследователей, так и конечных пользователей уделяется разработке и применению автоматизированных систем с речевым человеко-машинным интерфейсом. Устная речь является наиболее естественным и простым для человека способом общения, поэтому речевые технологии находят все большее распространение в роботехнике, управлении компьютерными устройствами, системах телекоммуникаций.

Особо актуальным речевой человеко-машинный интерфейс является в следующих двух случаях:

1) управление мобильными компьютерными устройствами (телефонами, смартфонами, нетбуками и т. д.);

2) управление персональным компьютером людьми с ограниченными физическими возможностями (прежде всего, с нарушениями опорно-двигательного и зрительного аппарата).

В первом случае использование традиционных средств тактильно-зрительного интерфейса (клавиатуры, мыши, дисплея) затруднено для пользователей из-за миниатюрности используемых компьютерных устройств, а во втором случае — и вовсе невозможно вследствие физических особенностей самих пользователей.

Ключевым элементом любого речевого интерфейса является система автоматического распознавания речи. Существует ряд методов построения таких систем. Наиболее популярными являются две группы методов:

1) методы распознавания речи, основанные на применении искусственных нейронных сетей;

2) методы распознавания речи, основанные на применении скрытых марковских моделей.

Авторы в течение многих лет ведут разработку систем автоматического распознавания как изолированных речевых команд, так и слитной речи, на базе нейросетевого подхода. Естественным образом возникает задача сравнить эти системы распознавания с другими системами, функционирующими на базе скрытых марковских моделей.

Исходя из вышеописанного, в данной работе была поставлена следующая цель: разработать и исследовать систему автоматического распознавания речевых команд для управления персональным компьютером на базе математического аппарата скрытых марковских моделей.

В качестве объекта исследования была выбрана инструментальная система Sphinx 4, разработанная американскими исследователями. Эта система предоставляет разработчикам удобный инструментарий для исследования

скрытых марковских моделей, а после определённой доработки может использоваться как система автоматического распознавания речевых команд управления компьютерными устройствами. Преимуществами Sphinx по сравнению с аналогичной инструментальной системой HTK являются:

1) открытая лицензия, по которой поставляется система Sphinx, что позволяет свободно использовать эту систему как в исследовательских, так и в коммерческих целях;

2) язык программирования Java, на котором написана система Sphinx, что идеально подходит для использования систем распознавания речи, построенных на базе Sphinx, в мобильных устройствах основанных на android от google, а также в сервлетах для последующего использования на устройствах типа CLDC.

Задачами данной работы являлись:

1) исследовать математический аппарат скрытых марковских моделей для автоматического распознавания речи;

2) на базе инструментальной системы Sphinx[1] разработать и обучить программную систему распознавания речевых команд управления компьютером, работающую как в однодикторном, так и в дикторонезависимом режимах;

3) на материалах собственного речевого корпуса длительностью свыше полутора часов оценить точность работы созданной системы распознавания речи.

## *Постановка задачи распознавания речевых команд*

В качестве математического аппарата, применяемого для распознавания речи в Sphinx4, применяются скрытые марковские модели [7]. Марковская модель – это вероятностный автомат с конечным числом состояний, изменяющий своё состояние один раз в единицу времени. При этом наблюдателю известны состояния и вероятности переходов между состояниями (матрица переходов). Таким образом, марковская модель описывает некоторый вероятностный процесс. Каждому наблюдаемому событию этого процесса соответствует одно из состояний модели.

Скрытая марковская модель, в отличие от обычной, описывает два вероятностных процесса – ненаблюдаемый (основной) и наблюдаемый (вспомогательный). О ходе основного процесса (например, процесса произнесения фонем устной речи) мы пытаемся судить по наблюдаемым событиям вспомогательного процесса (например, по изменению кратковременных спектральных характеристик звукового сигнала). В скрытой марковской модели задаётся не только множество состояний, но и алфавит символов наблюдения. Наблюдение является вероятностной функцией состояния. Для определения скрытой марковской модели необходимо задать следующие элементы:

1) $N$ – число состояний в модели. Хотя состояния и являются скрытыми от наблюдателя, им можно приписать физический смысл. Так, в распознавании речи под состояниями можно подразумевать фонемы слов или составные элементы фонем. Переход между состояниями осуществляется мгновенно.

2) $M$ – число различных символов наблюдения, которые могут порождаться моделью, т.е. размер дискретного алфавита. В распознавании речи в качестве такого дискретного алфавита может использоваться набор классов или кластеров, на которые разбивается множество возможных значений кратковременных спектральных характеристик звукового сигнала.

3) распределение вероятностей переходов между состояниями (или матрица переходных вероятностей) $A = \{a_{ij}\},\ i = \overline{1..N},\ j = \overline{1..N}$.

4) распределение вероятностей появления символов наблюдения в состоянии $j$, $B = \{b_j(k)\},\ j = \overline{1..N},\ k = \overline{1..M}$.

5) начальное распределение вероятностей состояний $\pi = \{\pi_i\},\ i = \overline{1..N}$.

В распознавании речи используются лево-правые скрытые марковские модели. Их последовательность состояний обладает свойством, которое выражается в том, что с увеличением времени индекс состояния $i$ также увеличивается или же остается неизменным. Т.е. состояния переходят слева направо, а наоборот сделать переход нельзя. Этот тип отлично подходит для описания процессов с прямым ходом времени, в частности, для распознавания речевых сигналов.

Любая система автоматического распознавания речи может функционировать в одном из двух режимов: режиме распознавания и режиме обучения. Использование скрытых марковских моделей для функционирования системы в режиме распознавания организовано следующим образом. В системе присутствует словарь распознаваемых элементов (например, речевых слов) размером $V$. Для каждого словарного элемента сформирована скрытая марковская модель $\lambda_v = \{A_v, B_v, \pi_v\},\ v = \overline{1..V}$. На вход поступает последовательность спектральных характеристик звукового сигнала,

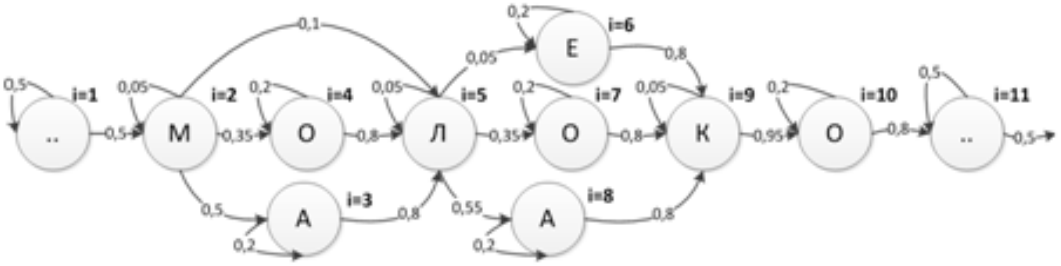


Рисунок 1 – СММ для «Молоко»

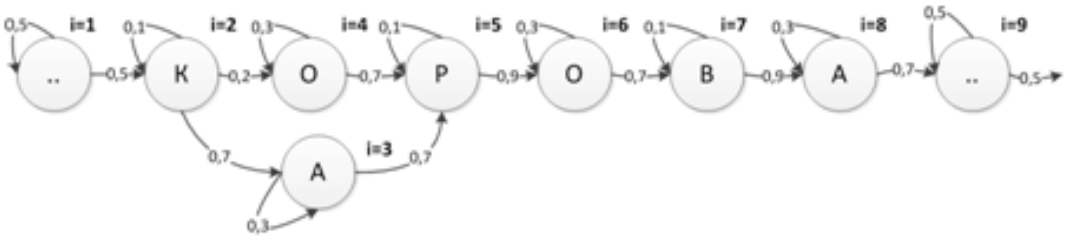


Рисунок 2 – СММ для «Корова»

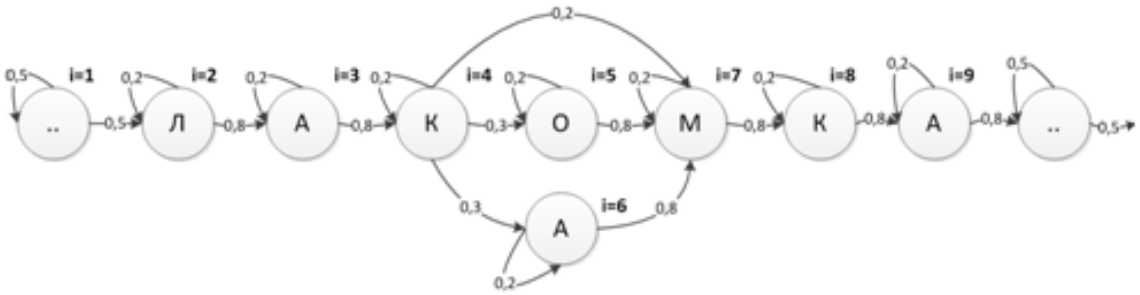


Рисунок 3 – СММ для «Лакомка»

рассматриваемая как последовательность наблюдений $O = O_1O_2..O_T$. Для каждого словарного элемента вычисляется $P(O \mid \lambda_v)$ – вероятность появления последовательности наблюдений $O$ для скрытой марковской модели $\lambda_v$, или, попросту говоря, вероятность того, что во входном звуковом сигнале присутствует *v*-й элемент словаря распознавания. Результат распознавания – номер распознанного словарного элемента $v_{res}$ – определяется следующим образом:

$$v_{res} = \arg\max_{1 \le v \le V}(P(O \mid \lambda_v)) \quad (1)$$

Проблема выбора последовательности состояний $Q = q_1q_2..q_T$, которая в значимом смысле будет оптимальной (например, наилучшим образом соответствует имеющейся последовательности наблюдений) решается с помощью алгоритма Витерби. Для того чтобы по заданной последовательности наблюдений $O = (O_1O_2...O_T)$ найти наилучшую последовательность состояний $Q=q_1q_2\dots q_T$ необходимо определить следующую величину: $\delta_t(i) = \max_{q_1,q_2..;q_{t-1}} P[q_1,q_2,\dots,q_t = i, O_1O_2..O_t \mid \lambda]$ что собой представляет максимальную вероятность того, что при заданных *t* первых наблюдениях последовательность состояний в соответствующие моменты времени заканчивается в состоянии *St.*

Рассмотрим, как работает алгоритм эффективного вычисления вероятности, на примере. Допустим, у нас есть 3 СММ для слов «молоко», «корова», «лакомка». Для слова «молоко» СММ имеет такие значения:

1) N=11
2) M=7 фонем(М, О, А, Л, Е, К, ..).
3) A (см. таблицу 1)
4) Распределение вероятностей появления символов наблюдения в состоянии j, B={bj(k)} (см. таблицу 2)
5) $\pi_1 = 1$, остальные $\pi$ равны 0.

Таблица 1 – Таблица вероятностей перехода между состояниями для СММ «Молоко»

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0,5 | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0,05 | 0,5 | 0,35 | 0,1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0,2 | 0 | 0,8 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0,2 | 0,8 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0,05 | 0,05 | 0,35 | 0,55 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0 | 0,8 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0,8 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0,8 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,05 | 0,95 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0,8 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,5 |

Таблица 2 – Таблица вероятностей появления символов для СММ «Молоко»

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| М | 0,5 | 0,05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| О | 0 | 0,35 | 0 | 0,2 | 0,35 | 0 | 0,2 | 0 | 0,95 | 0,2 | 0 |
| А | 0 | 0,5 | 0,2 | 0 | 0,55 | 0 | 0 | 0,2 | 0 | 0 | 0 |
| Л | 0 | 0,1 | 0,8 | 0,8 | 0,05 | 0 | 0 | 0 | 0 | 0 | 0 |
| Е | 0 | 0 | 0 | 0 | 0,05 | 0,2 | 0 | 0 | 0 | 0 | 0 |
| К | 0 | 0 | 0 | 0 | 0 | 0,8 | 0,8 | 0,8 | 0,05 | 0 | 0 |
| .. | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,8 | 0,5 |

Для того, чтобы по заданной последовательности наблюдений найти наилучшую последовательность состояний Q={q1, q2 … qt}, воспользуемся алгоритмом Витерби. Предположим, что у нас есть последовательность наблюдений .. .. М А А Л Л А К О О ..

В результате расчетов мы получили вероятность $P^* = \max_{1 \le i \le N}(\delta_T(i)) = 1*10^{-6}$ и наиболее вероятную последовательность состояний .. .. .. М М О О Л А К О ..

Рассчитаем вероятность этой же последовательности наблюдений, но для СММ «Корова». Она имеет такие параметры:

1) N=9
2) M=6 фонем(К, О, А, Р, В, ..).
3) A (см. таблицу 3)
4) B (см. таблицу 4)
5) $\pi_1 = 1$, остальные $\pi$ равны 0.

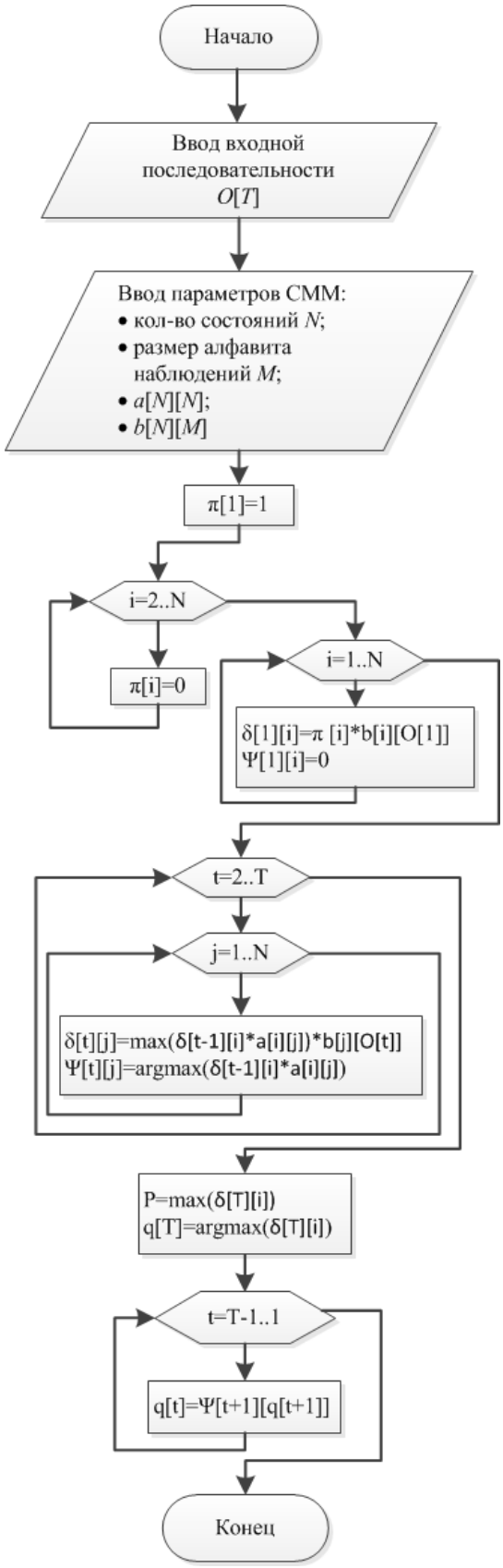


Рисунок 4 – Блок-схема вычисления вероятностей переходов между состояниями

Таблица 3 - Таблица вероятностей перехода между состояниями для СММ «Корова»

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0,5 | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0,1 | 0,7 | 0,2 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0,3 | 0 | 0,7 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0,3 | 0,7 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0,1 | 0,9 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0,3 | 0,7 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0,1 | 0,9 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,3 | 0,7 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,5 |

Таблица 4 - Таблица вероятностей появления символов для СММ «Корова»

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| К | 0,5 | 0,1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| О | 0 | 0,2 | 0 | 0,3 | 0,9 | 0,3 | 0 | 0 | 0 |
| А | 0 | 0,7 | 0,3 | 0 | 0 | 0 | 0,9 | 0,3 | 0 |
| Р | 0 | 0 | 0,7 | 0,7 | 0,1 | 0 | 0 | 0 | 0 |
| В | 0 | 0 | 0 | 0 | 0 | 0,7 | 0,1 | 0 | 0 |
| .. | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 0,7 | 0,5 |

$$P^* = \max_{1 \le i \le N}(\delta_T(i)) = 0$$

Вероятность равна 0, потому что в СММ «корова» нет переходов, которые есть в «молоке».

Рассчитаем вероятность для СММ «Лакомка».

1) N=10
2) M=6 фонем(Л, А, К, О, М, ..).
3) A (см. таблицу 5)
4) B (см. таблицу 6)
5) $\pi_1 = 1$, остальные $\pi$ равны 0.

Таблица 5 - Таблица вероятностей перехода между состояниями для СММ «Лакомка»

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0,5 | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0,2 | 0,8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0,2 | 0,8 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0,2 | 0,3 | 0,3 | 0,2 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0,8 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0,8 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0,8 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0,8 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0,8 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,5 |

Таблица 6 - Таблица вероятностей появления символов для СММ «Лакомка»

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Л | 0,5 | 0,2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| А | 0 | 0,8 | 0,2 | 0,3 | 0 | 0 | 0 | 0,8 | 0,2 | 0 |
| К | 0 | 0 | 0,8 | 0,2 | 0 | 0 | 0,8 | 0,2 | 0 | 0 |
| О | 0 | 0 | 0 | 0,3 | 0 | 0 | 0 | 0 | 0 | 0 |
| М | 0 | 0 | 0 | 0,2 | 0,8 | 0,8 | 0,2 | 0 | 0 | 0 |
| .. | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,8 | 0,5 |

На наблюдениях «.. ..» существовали вероятности и вероятные состояния. Но в СММ «Лакомка» нет перехода с «..» на «М», поэтому все последующие вероятности становятся равными 0. В итоге получаем $P^* = 0$

Следовательно, последовательность наблюдений .. .. М А А Л Л А К О О .. принадлежит к СММ «молоко».

Использование скрытых марковских моделей для функционирования системы в режиме обучения организовано следующим образом. Существует множество последовательностей наблюдений $O_L = \{\{O_1..O_{T_1}\},\{O_1..O_{T_2}\},...\{O_1..O_{T_L}\}\}$, сформированное на основе множества обучающих речевых сигналов *v*-го элемента словаря распознавания. Необходимо подстроить параметры скрытой марковской модели $\lambda_v = \{A_v, B_v, \pi_v\}$ так, чтобы максимизировать вероятности $P(O_\ell \mid \lambda_v)$, $\ell = \overline{1..L}$. Выполняя подстройку параметров, получаем скрытую марковскую модель *v*-го словарного элемента, наилучшим образом соответствующую обучающим последовательностям наблюдений. Применяя вышеописанную процедуру для всех элементов словаря распознавания $v = \overline{1..V}$, мы обучаем нашу систему автоматического распознавания речи.

### *Архитертура системы Sphinx*

В качестве инструментальной среды для разработки системы распознавания речи, основанной на скрытых марковских моделях, использовалась система Sphinx4. Эта система предоставляет разработчику развитую библиотеку классов, реализующих отдельные блоки типовой системы распознавания речи.

Рассмотрим объектно-ориентированную архитектуру данной библиотеки. Диаграмма классов представлена на рисунке 5.

Класс SearchManager (Модуль поиска) управляет процессом распознавания, используя следующие классы: SentenceHMMState (состояния CMM), AcousticScorer (блок выделения), ActiveList (список лексем) и Token (единица распознавания). Метод Recognize() класса SearchManager выполняет непосредственно распознавание речевого сигнала и возвращает результаты распознавания (объект класса Result), а метод StartRecognize() инициализирует процесс распознавания (должен вызываться перед вызовом метода Recognize()). StopRecognition() выполняет очистку после распознавания, должен вызываться после Recognize().

Класс SentenceHMMState (состояние CMM для предложения) – абстрактный класс, представляющий отдельное состояние графа SentenceHMM – CMM предложения в целом. Также этот класс является суперклассом для многих других классов состояний:

1) GrammarState (текущее состояние конечного автомата описывающего грамматику);

2) WordState (состояние CMM, характеризующее слово в предложении);

3) PronunciationState (транскрипция, соответствующая WordState).

Класс Grammar (грамматика) представляет собой дерево. Атрибут InitialNode – корень дерева. Атрибут Dictionary – словарь слов, которые может содержать грамматика. Соответственно, метод getInitialNode() – возвращает корень дерева, метод getNumNodes() – общее количество узлов, а метод getGrammarNodes() – возвращает набор узлов, присущих данной грамматике. Защищенный метод createGrammarNode() позволяет создавать новый узел грамматики в объекте. Класс GrammarNode содержит следующие методы: getWord() (возвращает слово, соответствующее текущему узлу грамматики), isEmpty() (проверяет существует ли слово, соответствующее данному узлу), add(GrammarNode node, float Probability) (добавляет новый переход из текущего узла в node с вероятностью Probability). GrammarState содержит узел (атрибут node: GrammarNode) текущего состояния грамматики. Метод GetGrammarNode() возвращает текущий узел.

Класс WordState хранит состояние CMM, характеризующее слово в предложении. Метод getWord() возвращает это слово.

Класс PronunciationState –произношение слова в SentenceHMM. Метод getPronuntiation() возвращает произношение текущего слова. Метод isWordStart() определяет было ли обнаружено начало слова.

Класс SentenceHMM состоит из множества потомков абстрактного класса SentenceHMMState, перечисленных выше, и матрицы вероятностей переходов между состояниями, унаследованной от класса HMM. Каждому состоянию сопоставлена последовательность кластеров. HMM представляет собой совокупность состояний CMM – объектов класса HMMState.

Лингвистический модуль(класс Linguist) на основе базы знаний создает все необходимые объекты-потомки класса SentenceHMMState. В эту базу знаний входят:

1) словарь (Dictionary)

2) акустическая модель (AcousticModel)

3) грамматика (Grammar)

4) языковая модель (LanguageModel)

Класс Dictionary содержит доступные слова для распознавания, а также список слов-заполнителей (fillers), которые нужно будет пропускать без какой-либо обработки (например, звуки дыхания, междометия, слова-паразиты). Статические атрибуты класса PROP_DICTIONARY и PROP_FILLER_DICTIONARY хранят пути к файлам словаря (файл с расширением .dic) и файлам с заполнителями (.filler). Метод GetFillerWords() возвращает набор слов-заполнителей. Все методы словаря возвращают слово (Word). Слово имеет 3 атрибута: произношение (pronunciation), написание (spelling) и флажок (isFiller) является ли это слово заполнителем.

Класс AcousticModel содержит CMM слов или фонем, полученные в процессе обучения системы. Для поиска оптимальной CMM в классе есть следующие методы: lookupNearestHMM() (возвращает CMM, которая наиболее соответствует предоставленным данным) и getHMMIterator() (итератор для прохождения по всем CMM).

Языковая модель содержит вероятности появления слов. GetProbability(WordSequence *wordSequence*) возвращает вероятность появления последовательности слов, переданной в качестве аргумента. GetVocabulary возвращает набор слов в языковой модели (набор неизменный).

Класс Microphone захватывает речевой сигнал с микрофона и передает его блоку предварительной обработки сигнала (блоку препроцессинга). Этот блок реализован классом FrontEnd. Метод SetDataSource() класса FrontEnd позволяет установить источник звуковых данных для модуля препроцессинга (микрофон или файл).

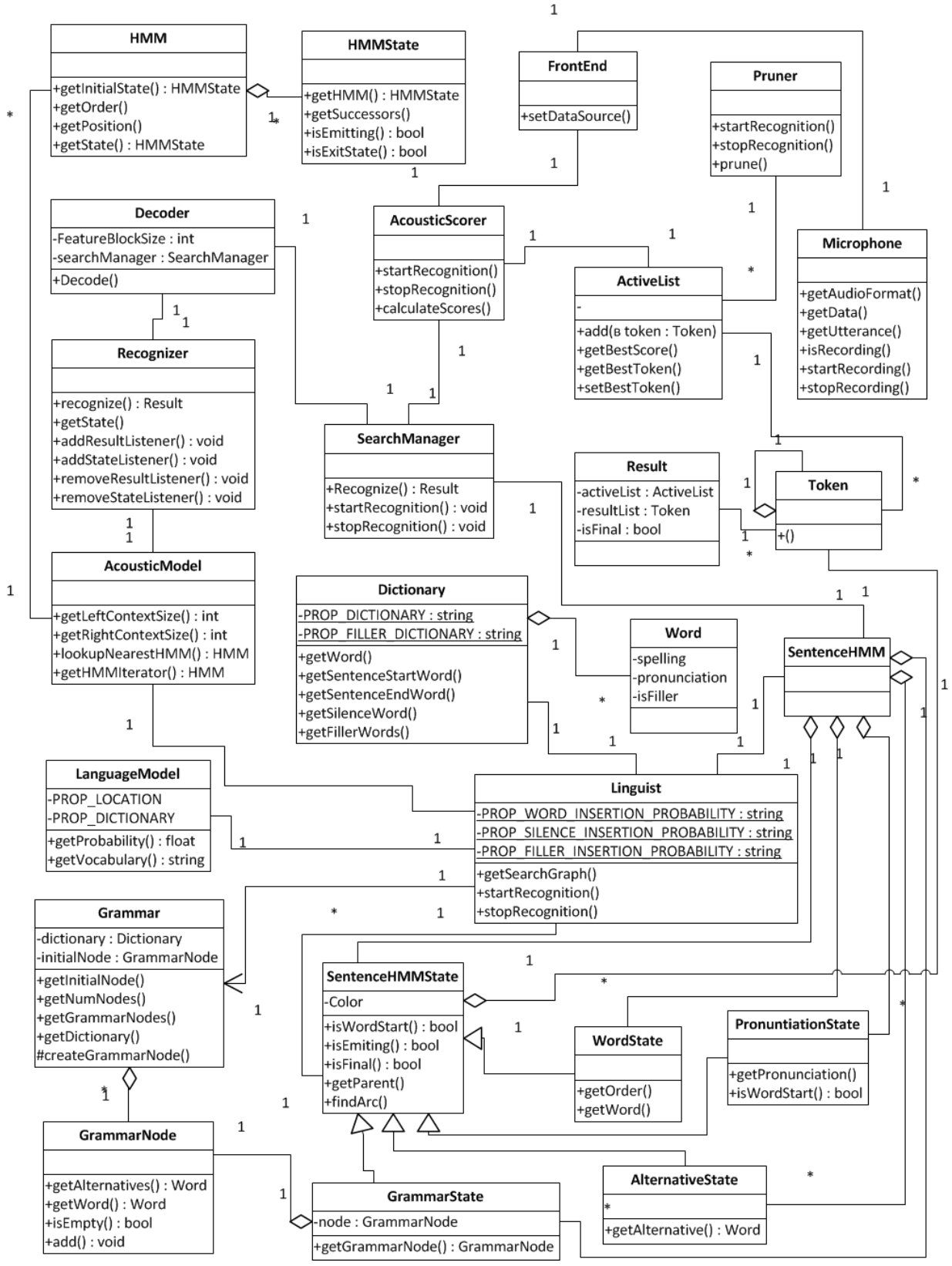


Рисунок 5 – Упрощенная диаграмма классов системы Sphinx 4

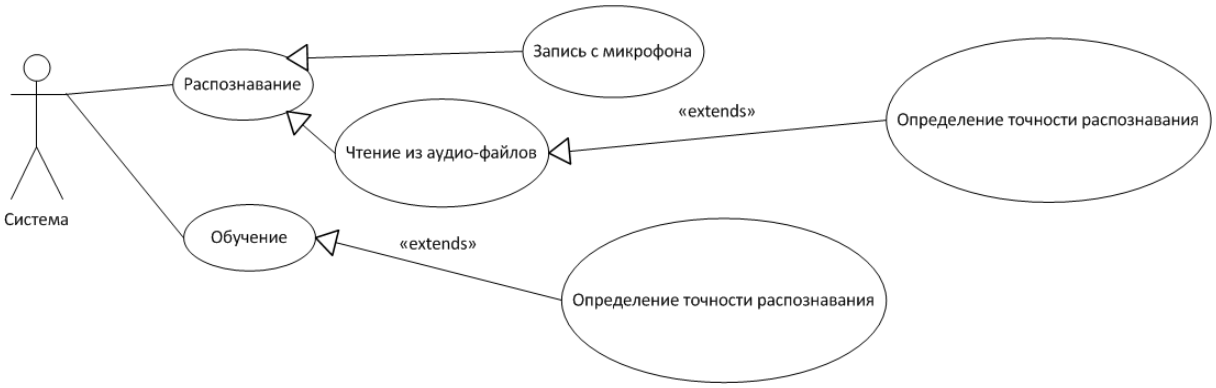


Рисунок 6 – Диаграмма вариантов использования системы Sphinx

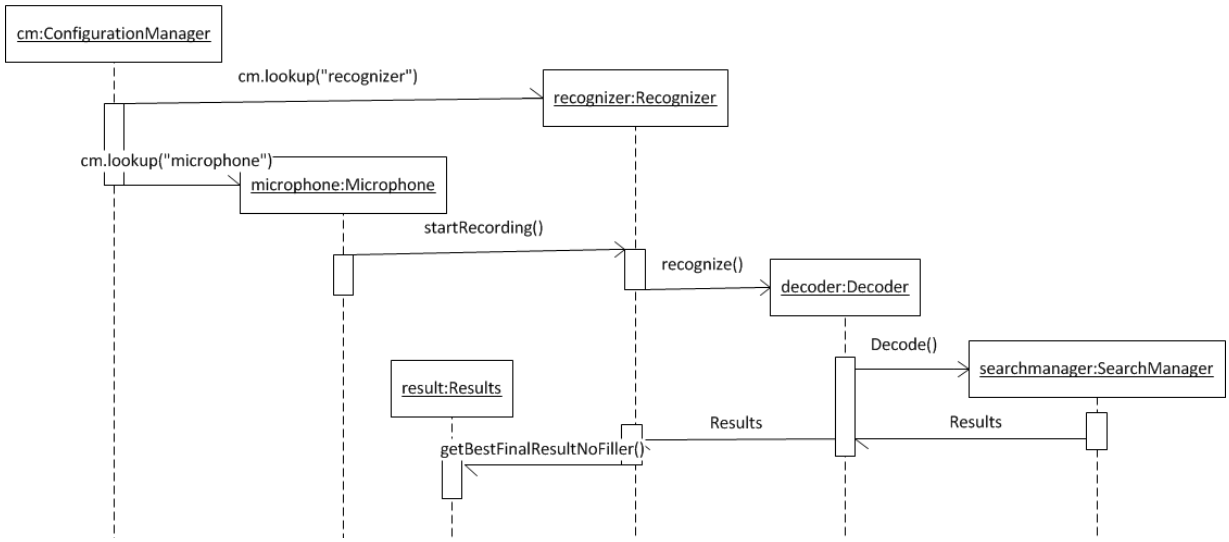


Рисунок 7 – Диаграмма последовательности сценария распознавания команд с микрофона

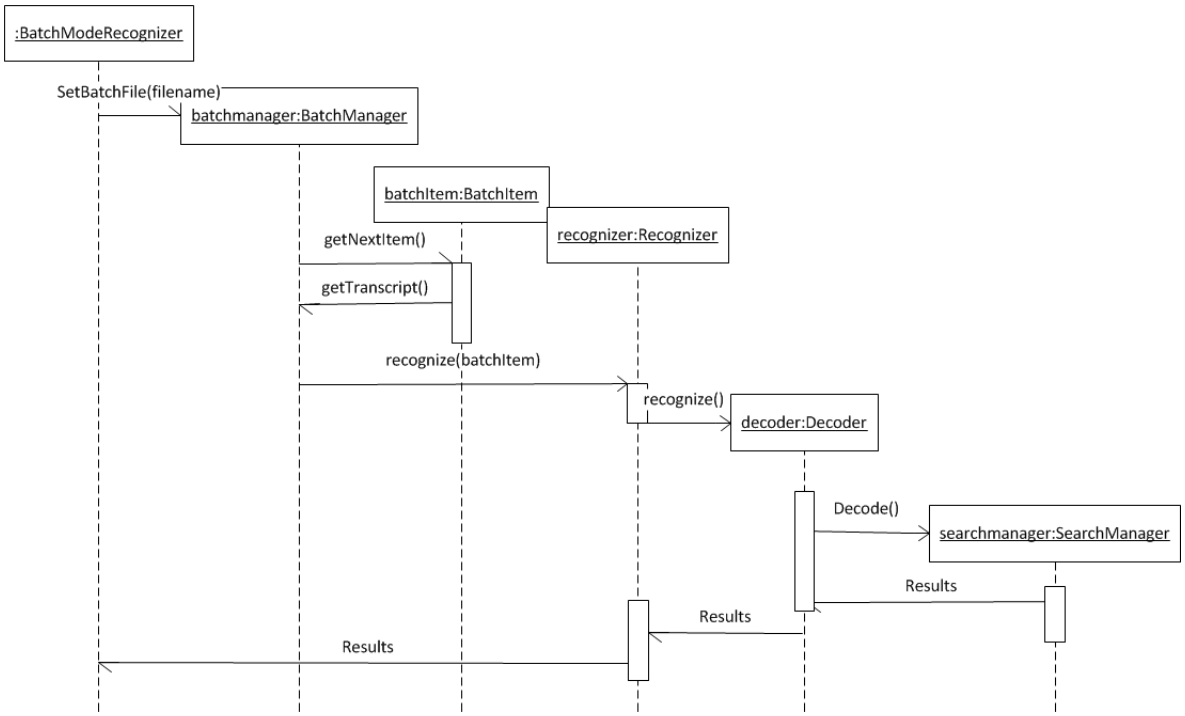


Рисунок 8 – Диаграмма последовательности сценария распознавания команд из аудиофайла

Класс AcousticScorer (блок выделения) получает данные из FrontEnd, выделяет единицы распознавания (объекты класса Token) и составляет список из них (класс ActiveList). Осуществляется проход по списку и каждой распознанной единице речи сопоставляется состояние CMM.

Объекты класса Token также используются в формировании результатов распознавания (класс Result). Класс Token хранит указатель на состояние (SentenceHMMState), сопоставленное ему блоком выделения, и указатель на предыдущий объект Token.

Блок сегментации (класс Pruner) сегментирует ActiveList (последовательность состояний) на единицы распознавания.

Класс SearchManager на основе данных из классов AcousticScorer, Decoder и SentenceHMM генерирует результаты (объект класса Result) распознавания. Подробнее их взаимодействие рассматривается на диаграммах последовательности (рис. 7 и рис. 8)

На рисунке 6 показана диаграмма вариантов использования инструментальной среды Sphinx. На ней показаны функциональные возможности, используемые системой, в частности: обучение и распознавание. Распознавание может выполняться как с микрофона, так и с аудиофайлов. Функциональные возможности прецедента «Чтение из аудиофайлов» дополняются прецедентом «Определение точности распознавания». В ходе распознавания речевого потока из аудиофайлов может понадобиться провести анализ точности распознавания, т.е. посчитать WER. Аналогичным образом функциональные возможности расширяются и у прецедента «Обучение». Обучением системы на распознавание занимается модуль системы Sphinx – SphinxTrain. На вход ему подается набор обучающих речевых сигналов и их транскрипций, а также база знаний: словарь, перечень используемых фонем и языковая модель.

Рассмотрим последовательности действий при распознавании речи с микрофона и при определении точности распознавания аудиофайлов.

На рисунке 7 приведена диаграмма последовательности, иллюстрирующая сценарий распознавания речевых команд с микрофона. Предположим, что этот сценарий начинается с момента, когда создан объект класса ConfigurationManager (конфигурационный менеджер). Объект cm с помощью метода lookup выполняет поиск настроек (конфигураций) в файле настроек xml. Как следует из диаграммы, lookup("recognizer") и lookup("microphone") представляют собой асинхронные операции, поскольку объект cm не нуждается в ответах от только что созданных объектов класса Microphone и Recognizer. Операция startRecording() объекта microphone представляет собой метод класса Microphone, начинающий запись аудио сигнала. Это синхронная операция, т.к. в случае невозможности произвести запись, microphone получает сведения об ошибке и завершается работа программы. Объект класса Recognizer в случае удачного начала записи звука с микрофона посылает сообщение объекту класса Decoder. Объект decoder передает обработанные данные объекту класса SearchManager, а он, в свою очередь, генерируется результаты и передает их объекту decoder .

На рисунке 8 показана диаграмма последовательности сценария распознавания команд из аудиофайла. Вместо конфигурационного менеджера и микрофона создается объект класса BatchModeRecognizer. Объект batchManager свяжется с файлом .batch и с помощью класса BatchItem и метода getNextItem() получит строку с правильными ответами для текущего аудиофайла. Класс BatchItem имеет всего 2 атрибута: название файла, который будет распознаваться, и транскрипция этого файла. Операция recognize(batchItem) передает классу Recognizer текущий объект batchItem, далее данные передаются объекту decoder. И в конце, как и на предыдущей диаграмме, объект searchmanager выдаст результаты распознавания, а batchmoderecognizer на основе этих результатов посчитает ошибку распознавания (Word Error Rate, или процент неправильно распознанных слов – см. следующий раздел) и запишет полученную информацию в журнал.

### *Эксперименты по распознаванию речевых команд управления компьютером*

При проведении экспериментальных исследований использовалась инструментальная среда Sphinx 4, предназначенная для автоматизации проектирования систем распознавания речи на базе скрытых марковских моделей. С помощью Sphinx 4 была построена система распознавания речевых команд управления персональным компьютером со словарём объёмом 100 слов.

Для обучения данной системы дикторонезависимому распознаванию была использована свободная речевая база VoxForge [8]. Общая продолжительность аудиоматериала

базы 9,5 часов. В записи принимали участие 18 дикторов.

Для обучения системы речевого управления однодикторному распознаванию команд была сформирована собственная речевая база. Продолжительность аудиоматериала в ней составила 1,31 часа. Все записи выполнял один диктор.

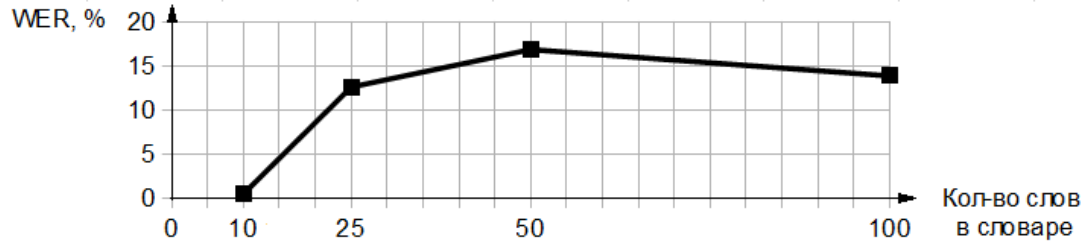


Рисунок 9 - График зависимости точности дикторонезависимого распознавания речевых команд от объема словаря на материале тестовой шестидикторной речевой базы

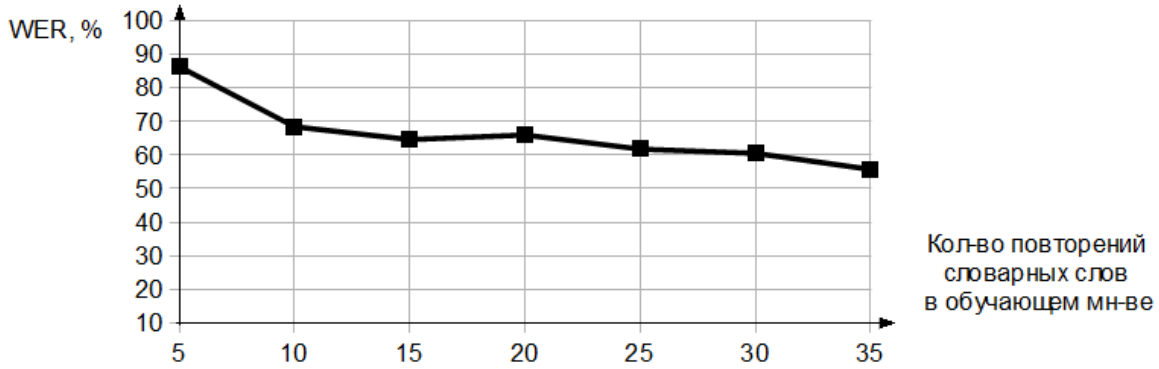


Рисунок 10 - График зависимости точности однодикторного распознавания речевых команд (на базе словаря 100 слов) от объема обучающего множества

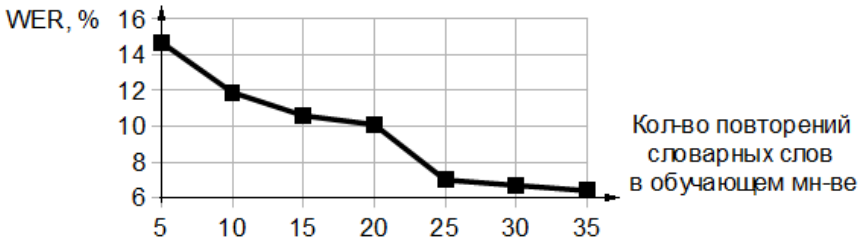


Рисунок 11 - График зависимости точности распознавания речевых команд от объема словаря на материале тестовой шестидикторной речевой базы при условии, что система распознавания обучалась в однодикторном режиме (на одного диктора)

Для проверки эффективности распознавания было создано 4 словаря: на 10 слов, 25, 50 и 100. Чтобы слова из словаря объединить в различные словосочетания, с помощью которых оператору удобно было бы отдавать команды компьютеру, был создан набор контекстно-свободных грамматик (пример такой грамматики для словаря из 25 слов показан на рис.2).

В качестве источника аудиоматериала, на котором оценивалась точность работы обученной системы распознавания, использовалась собственная шестидикторная речевая база данных. В её формировании приняли участие трое дикторов-мужчин и трое дикторов-женщин. В их задачи входило чтение всех предложений грамматики для 4 вышеописанных словарей. Запись производилась в несжатом формате raw с частотой 16000 Гц и разрядностью звука 16 бит в соответствии с требованиями Sphinx4.

Для численной оценки точности распознавания использовался стандартный критерий WER (Word Error Rate) [9]. Он вычисляется по следующей формуле:

$$WER = \frac{S + D + I}{N} \cdot 100\%, \quad (2)$$

где S – количество замен (substitutions), D – количество удалений (deletions), I – количество вставок (insertions), которые необходимо применить к цепочке распознанных слов, чтобы она совпала с цепочкой правильных слов, а N – общее количество правильных слов.

Было проведено три эксперимента, направленных на оценивание точности распознавания речевых команд системой автоматического распознавания, разработанной с использованием инструментальной среды Sphinx 4 на базе математического аппарата скрытых марковских моделей.

Целью первого эксперимента было определение точности дикторонезависимого распознавания речевых команд. Система распознавания обучалась на материале речевой базы VoxForge, а тестировалась на материале собственной шестидикторной речевой базы. Результаты распознавания представлены на рис.9. и в таблице 7.

Таблица 7 – Таблица WER для каждого словаря и диктора

| | Количество слов в словаре | | | |
|---|---|---|---|---|
| | 10 | 25 | 50 | 100 |
| Женщина 1 | 2,78 | 18,37 | 21,28 | 15,85 |
| Женщина 2 | 0 | 6,12 | 12,76 | 7,65 |
| Женщина 3 | 0 | 26,53 | 23,4 | 26,78 |
| Мужчина 1 | 0 | 8,16 | 11,7 | 9,29 |
| Мужчина 2 | 0 | 6,12 | 14,89 | 10,93 |
| Мужчина 3 | 0 | 10,2 | 17,02 | 12,57 |

Целью второго эксперимента было определение точности однодикторного распознавания речевых команд. Система распознавания обучалась на материале собственной однодикторной речевой базы, а тестировалась на дополнительном аудиоматериале длительностью около восьми минут, записанном тем же диктором. Результаты распознавания представлены на рис.10.

Целью третьего эксперимента было определение того, насколько система распознавания снижает точность своей работы, если её обучать как однодикторную, а использовать в дикторонезависимом режиме. Система распознавания обучалась на материале однодикторной речевой базы, а тестировалась на материале шестидикторной речевой базы. Результаты распознавания представлены на рис.11 и в таблице 8.

Таблица 8 – Таблица WER для каждого диктора по объему обучающего множества

| | Объем обучающего множества | | | | | | |
|---|---|---|---|---|---|---|---|
| | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
| Женщина 1 | 72,13 | 44,26 | 40,44 | 40,44 | 37,71 | 31,69 | 25,68 |
| Женщина 2 | 74,32 | 44,26 | 30,06 | 32,24 | 25,14 | 25,68 | 16,94 |
| Женщина 3 | 94,54 | 79.24 | 77,6 | 79,78 | 78,69 | 75,96 | 71,59 |
| Мужчина 1 | 94.54 | 80,33 | 79,78 | 80,33 | 74,32 | 72,13 | 68,85 |
| Мужчина 2 | 87,43 | 73,77 | 70,49 | 72,68 | 67,76 | 69,4 | 65,03 |
| Мужчина 3 | 94,54 | 88,53 | 89,07 | 90,16 | 86,89 | 87,98 | 86,16 |

### *Заключение*

В данной работе была создана и исследована система автоматического распознавания речевых команд для управления персональным компьютером на базе математического аппарата скрытых марковских моделей. В ходе выполнения работы были получены следующие результаты:

1) исследован математический аппарат скрытых марковских моделей применительно к решению задачи автоматического распознавания речи;

2) на базе инструментальной системы Sphinx разработана и обучена программная система распознавания речевых команд управления компьютером, работающая в двух режимах: в однодикторном и дикторонезависимом;

3) на материалах собственного речевого корпуса длительностью свыше полутора часов (1,31 часа обучающего аудиоматериала и 0,2 часа тестового материала) оценена точность работы созданной системы распознавания речи.

Анализ результатов экспериментальных исследований показывает следующее.

1. Чем больше словарь системы распознавания, тем хуже точность работы этой системы в дикторонезависимом режиме. Известно, что речевой интерфейс становится удобен пользователям лишь в том случае, если ошибка распознавания речевых команд при использовании этого интерфейса не превышает 5%. Для системы дикторонезависимого распознавания речевых команд, разработанной на базе скрытых марковских моделей, этот порог превышается уже на словаре объёмом 25

слов. Таким образом, применение такой системы для полноценного речевого управления компьютером без предварительной подстройки под диктора не представляется оправданным: высокая точность достигается лишь при малом словаре, размер которого затрудняет речевой диалог, а на словаре более приемлемых размеров снижается точность распознавания.

2. При использовании системы речевого управления компьютером в однодикторном режиме оператору необходимо собрать достаточно большой объём аудиоматериала, чтобы достичь точности распознавания, близкой к 5 %. Кроме того, на результат обучения и, как следствие, на точность распознавания влияет не только объем материала, но и его качество. Чем понятнее будут произнесены слова, тем ниже будет ошибка. Также немаловажен порядок слов. Его обязательно нужно менять, иначе усилия, приложенные к записи большого объема материала, будут безрезультатными. Каждая запись должна быть дольше 5 секунд, но меньше 30. Между словами должны быть соблюдены паузы. Слова произносятся в соответствии с транскрипцией в словаре.

3. Категорически не рекомендуется использовать систему распознавания речи, настроенную на использование в однодикторном режиме, т.е. на одного оператора, для распознавания речевых команд других операторов.

Таким образом, можно сделать следующие выводы:

1) применение инструментария Sphinx для создания системы дикторонезависимого речевого управления компьютерными устройствами нецелесообразно, а для создания системы с подстройкой под диктора – затруднено ввиду громоздкой процедуры обучения, требующей со стороны оператора значительных усилий по формированию обучающего аудиоматериала;

2) аппарат скрытых марковских моделей неплохо подходит для описания процессов с прямым ходом времени, таких как порождение речи, но применение только этого аппарата для распознавания речевых команд без привлечения других мощных методов распознавания (нейронных сетей, нечёткой логики и т.п.) не всегда эффективно.

Дальнейшие исследования будут направлены на усовершенствование системы Sphinx в двух направлениях:

– разработка инвариантной системы фонетических признаков речевого сигнала, вычисляемой с помощью искусственных нейронных сетей и используемой при формировании входных сигналов для скрытых марковских моделей;

– исследование алгоритмов распознавания, альтернативных скрытым марковским моделям.

Целью такого усовершенствования будет являться повышение точности дикторонезависимого распознавания речи.

### *Список литературы*

1. Ронжин А.Л. Система автоматического распознавания русской речи SIRIUS / А.Л. Ронжин, А.А. Карпов, И.В. Ли. - Спб.: СПИИРАН, 2006. - 12 с.

2. Федяев О.И. Сегментация речевого сигнала на основе bagging-коллектива нейросетевых детекторов фонем / О.И. Федяев, И.Ю. Бондаренко // Материалы 8-й Международной научно-практической конференции [«Математическое и программное обеспечение интеллектуальных систем»] (MPZIS-2010). – Днепропетровск: ДНУ, 2010. – С. 238-239.

3. Аграновский А.В. Теоретические аспекты алгоритмов образотки и классификации / А.В. Аграновский, Д.А. Леднов. - М.: Радио и связь, 2004. - 164 с.

4. Федяев О.И. Организация системы автоматического распознавания речи на основе коллектива распознающих автоматов / О.И. Федяев, И.Ю. Бондаренко // Материалы 4-й международной научно-технической конференции ["Моделирование и компьютерная графика - 2011"]. – Донецк, 2011. - С. 309-316.

5. CMU Sphinx Open Source Toolkit For Speech Recognition Evaluation [Electronic resourse] / [Интернет-ресурс]. - Режим доступа: http://cmusphinx.sourceforge.net/ [проверено 1.04.2012]. - Загл. с экрана.

6. What is HTK? [Electronic resourse] / [Интернет-ресурс]. - Режим доступа: http://htk.eng.cam.ac.uk/ [проверено 1.04.2012]. - Загл. с экрана.

7. Рабинер Л.Р. Скрытые марковские модели и их применение в избранных приложениях при распознавании речи / Л.Р. Рабинер // ТИИЭР. - 1984. - Т.72, № 2. - С. 86-120.

8. Welcome – Russian Evaluation [Electronic resourse] / [Интернет-ресурс]. - Режим доступа: http://www.voxforge.org/ru [проверено 1.04.2012]. - Загл. с экрана.

9. Word Error Rate [Electronic resourse] / [Интернет-ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/Word_error_rate [проверено 1.04.2012]. - Загл. с экрана.

*Надійшла до редакції 30.09.2012*

**І.Ю. БОНДАРЕНКО, Д.Г. САВКОВА**
Донецький національний технічний університет

**Дослідження системи Sphinx4 для розв'язання задач однодикторного та дикторонезалежного розпізнавання мовних команд**

Дане дослідження присвячене розробці системи розпізнавання мовних команд керування комп'ютерними системами на базі інструментального середовища Sphinx 4. Проаналізовано математичний апарат прихованих марковських моделей, реалізований в Sphinx 4 для організації процесу розпізнавання мови і навчання такому розпізнанню. Проведено експериментальні дослідження системи розпізнавання мовних команд на базі Sphinx 4, спрямовані на оцінку точності роботи цієї системи як в однодикторному, так і в дикторонезалежному режимах. Зроблені практичні рекомендації по ефективності застосування інструментального середовища Sphinx 4 для організації мовного інтерфейсу комп'ютерних систем.

***Ключові слова: автоматичне розпізнавання мови, приховані марківські моделі, мовний інтерфейс, Sphinx***

**I.Y. BONDARENKO, D.G. SAVKOVA**
Donetsk National Technical University

**Research of Sphinx4 System for Solving the Tasks of Single-Speaker and Speaker-Independent Speech Recognition**

The paper discusses the system of computer administration speech commands recognition based on instrumental environment Sphinx 4. The hidden Markov's models implemented in Sphinx to organize the speech recognition process are analyzed. The experimental research of the speech command recognition system based on Sphinx 4 is conducted. The research is aimed at estimating the system performance in both single-speaker and speaker-independent modes. We also provide some practical recommendations as for Sphinx 4 usage for the computer systems speech interface organization.

***Keywords: automatic speech recognition, hidden Markov's model, speech interface, Sphinx***