УДК 519.711.3: 004.8.023

С. Г. Заблоцкий, Е. С. Семенкин, А. В. Швец

КОЭВОЛЮЦИОННЫЙ АСИМПТОТИЧЕСКИЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ФОРМИРОВАНИЯ ПРЕДЛОЖЕНИЙ ПО СЛОГОВОЙ МОДЕЛИ В СИСТЕМЕ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ СЛИТНОЙ РЕЧИ*

Последний этап работы системы автоматического распознавания слитной речи, основанной на слоговой модели, состоит в преобразовании полученной цепочки слогов в последовательность слов, образующих предложение. В данной работе предложен и исследован метод, основанный на специальном стохастическом алгоритме оптимизации, позволяющий определять за приемлемое время наиболее правдоподобное предложение, соответствующее входящему звуковому сигналу.

Ключевые слова: коэволюционный алгоритм, слоговая модель, распознавание речи, динамическое программирование.

Задача распознавания слитной речи состоит в том, чтобы составить наиболее правдоподобную последовательность слов, соответствующую полученному акустическому сигналу. Существуют системы обработки слитной английской речи, однако некоторые особенности русского языка не позволяют напрямую использовать разработанные системы. Русский язык обладает большим количеством различных словоформ, в связи с этим традиционные пословные модели языка, основанные на N-граммах (последовательностей, состоящих из n элементов) для слов, оказываются неприменимыми [1]. Размер используемого словаря многократно увеличивается, в то время как качество и скорость обработки информации существенно снижается.

Разработанные слоговые модели позволяют сократить размер словаря основных единиц распознавания на несколько порядков. Так, словарь, состоящий из 1,5 миллиона слов, может быть преобразован в словарь, содержащий около 12 тысяч слогов. Благодаря этому становится возможным разработать систему, позволяющую вести распознавание с высокой точностью и скоростью.

Распознавание проходит в несколько этапов, в течение которых используются акустические и лингвистические модели языка и применяются скрытые марковские модели, дискретное преобразование Фурье, построение частотных кепстральных коэффициентов, извлечение фонем языка [2]. Получаемые в процессе распознавания фонемы преобразуются в большие по размеру единицы распознавания на основе материала, который был получен при предварительной тренировке системы на текстовых корпусах. Таким образом, для слоговой модели речь преобразуется в слоги, при этом не отмечаются места, в которых должны проходить границы слов.

Полученная цепочка слогов должна быть преобразована в последовательность слов. Данный этап является одним из наиболее важных. Входящая цепочка может иметь более шести десятков слогов, при этом часть из них может содержать ошибки, возникшие при распознавании акустического сигнала. Согласно [3] считается, что на данный момент не существует правил и методов определения границ слов при использовании такой модели. Рассмотрение всех комбинаций слов за приемлемое время является невозможным и требует высоких вычислительных мощностей.

В данной работе предложен и исследован метод, позволяющий за приемлемое время определить границы слов, восстановить последовательность произнесенных слов и повысить качество распознавания. В том числе разработан коэволюционный асимптотический генетический алгоритм решения задач безусловной оптимизации, позволяющий определять наиболее правдоподобное предложение.

Метод оценки правдоподобия предложения по слоговой модели. Чтобы определить, какое из нескольких произвольно составленных из слогов предложений больше удовлетворяет искомой последовательности слов, предложен способ получения оценки правдоподобия предложения. Одним из инструментов, необходимых для получения оценки, является большой словарь всех словоформ русского языка. Так, выбран грамматический словарь Зализняка [4], содержащий около 1,7 миллиона слов. Стоит отметить, что передовые системы распознавания речи для английского языка используют словарь порядка 100 тысяч слов. Слоги, получаемые в результате преобразования речи, не соответствуют по написанию слогам, составляющим произнесенное слово. Например, слово «молоко» при отсутствии ошибок будет распознано как {ма, ла, ко}, так как безударная гласная «о» произносится как фонема «а». В связи с этим слова используемого словаря приведены к такому виду, как произносятся. Преобразование выбранного словаря к указанному виду осуществляется на основании правил произношения слов русского языка [5]. Для удобства описания далее в примерах используется правильное написание слов.

Для более точного определения состава предложения применяется статистика встречаемости фраз. Каждое слово русского языка имеет большое разнообразие словоформ; так, многие словосочетания отличаются друг от друга только формами входящих в них слов.

*Данные исследования проводятся при поддержке ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» 2011-1.9-519-005.

Например, словосочетания «построил большой дом» и «построивший большой дом» отличаются только формой первого слова, начальные же формы всех слов совпадают. Объединение всех словоформ слова в один класс позволяет увеличить частоты отдельных N-грамм до приемлемого уровня, кроме того, это дает возможность использовать статистику при ошибочно распознанных окончаниях слов [6]. В качестве основных используемых N-грамм выбраны триграммы, так как по сравнению с двуграммами они содержат больше информации и в отличие от N-грамм более высокого порядка имеют большую частоту встречаемости. Целесообразно, чтобы оценка правдоподобия была тем выше, чем больше находится в предложении триграмм из базы N-грамм и чем выше их частота.

Необходимо, чтобы составленные слова являлись словами из словаря, именно в этом случае можно будет определить начальные формы слов и проверить наличие тройки слов в базе N-грамм. Присутствие же малейших ошибок распознавания приведет к тому, что некоторые слова будут отмечены как неизвестные, и предложение получит низкую оценку. Предложенный алгоритм способен обрабатывать ошибки распознавания. Существует три типа ошибок: удаление слогов, замена одних на другие или добавление лишних. Очевидно, что перебрать для данного предложения все варианты возможных операций замен, удалений и вставок невозможно за приемлемое время при большом числе слогов, даже если вводить условное ограничение на допустимое число операций. При выполнении алгоритма определяется, каким словам из словаря могут соответствовать составленные комбинации слогов. Один набор слогов может соответствовать нескольким словам, причем в большей или меньшей степени. Например, слово «бан» одинаково похоже на слова «банк» и «бант», но меньше похоже на слова «банан» и «бинт». Таким образом, для каждого слова предложения требуется получить список похожих слов и соответствующих им чисел, которые показывали бы, насколько рассматриваемое слово отвечает найденному слову из словаря. Для удобства далее будем называть такие числа коэффициентами подобия.

Для того чтобы узнать, насколько близким является одно слово по отношению к другому, возьмем за основу расстояние Левенштейна. Расстоянием Левенштейна (также редакционное расстояние) между двумя строками в теории информации и компьютерной лингвистике называют минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для преобразования одной строки в другую. Задачу поиска расстояния между словами эффективно решает алгоритм Вагнера–Фишера [7] с помощью метода динамического программирования.

Чтобы учесть схожесть слов по звучанию при вычислении расстояния, алгоритм Вагнера–Фишера был модифицирован введением различных нецелочисленных весов для удаления, вставки или замены элемента, зависящих от фонетических особенностей элементов. Во-первых, присваиваются немного большие веса операциям с первыми буквами слова, так как первую часть слова диктор обычно произносит более четко, чем последнюю. Таким образом, считаем, что изменение первых букв меняет слово сильнее, нежели изменение последних букв. Во-вторых, больший вес имеют операции с гласными, так как они длиннее по звучанию, более устойчивы к изменению и поэтому легче поддаются распознаванию, чем согласные. Но замена некоторых гласных другими, такая как «о – а», «е – и», будет иметь несколько меньший вес, поскольку эти гласные часто близки по произношению в русском языке. В-третьих, небольшой вес будут иметь операции над согласными, причем немного меньший – операции замены согласных некоторых пар, таких как «ж – ш», «б – п», и еще меньший – операции добавления и удаления сдвоенных согласных.

Посчитав таким способом расстояние d между словами, алгоритм может вычислить коэффициент подобия K между словами p_1 и p_2, по следующей формуле:

$$K(p_1, p_2) = \begin{cases} 1 - d \cdot N_1(n), & d \le N_2(n), \\ 0, & d > N_2(n), \end{cases}$$

где n – длина рассматриваемого слова p_1; N_1 и N_2 – некоторые переменные, зависящие от длины слова и сочетающиеся так, что $0 \le K(p_1, p_2) \le 1$. Таким образом, при расстоянии $d = 0$, т. е. при совпадении слов коэффициент подобия равен 1, при значительном расстоянии равен 0. Зависимость от n появляется из тех соображений, что изменения в слове небольшой длины выражены сильнее, чем такие же изменения в более длинном слове. Так, слово «ус» меньше похоже на «уж», чем слово с ошибкой «предлосенный» на слово «предложенный».

Возвращаясь к задаче построения списка похожих слов, следует отметить, что из всех слов словаря в этот список включаются только те, которые удовлетворяют некоторому введенному ограничению на допустимый суммарный вес операций изменения слова W. Так, нет необходимости находить расстояния со всеми словами словаря. Как только словарное слово или его начальная часть отличается от рассматриваемого слова более чем на W, то его и все слова из словаря, имеющие схожую начальную часть, следует убрать из рассмотрения.

Ускорить процесс поиска позволяет хранение словаря в форме дерева, в английской литературе называемого trie, происходящего от части слова «retrieval», что означает поиск, извлечение [8]. Структура trie представляет собой граф, узлами которого являются части слов (в данной работе – фонемы). Вершиной дерева является пустой узел, который представляет собой начало всех слов из словаря. Конечными узлами (листьями) являются окончания слов. Основное преимущество этой структуры – возможность эффективного поиска всех слов, имеющих общее начало. Такая структура применяется в различных областях, например, может быть использована как вспомога-

тельный инструмент в системах автоматической коррекции слов в тексте [9].

При таком подходе, при нарушении ограничения на W, алгоритм перестанет идти вглубь дерева, перейдя на соседнюю ветвь или поднявшись на уровень вверх, тем самым отбросив определенное число слов из рассмотрения. Как только достигается конец слова за допустимое число операций, вычисляется коэффициент подобия, который вместе с найденным словом добавляется в список подобных слов для сравниваемого со словарем слова.

Полученный для каждого слова составленного предложения список похожих слов следует преобразовать в список, состоящий из начальных форм, соответствующих данным словам. При этом размер списка может немного увеличиться, так как одно слово может иметь несколько начальных форм. Таким, например, является слово «гладь», которое имеет две начальные формы: «гладь» и «гладить».

Для того чтобы найти наиболее правдоподобное предложение, следует выбрать по одному слову из каждого списка так, чтобы полученная последовательность слов доставляла максимум произведению триграмм. При этом учитываются коэффициенты подобия слов, т. е. добавляется некоторый штраф за изменение исходной последовательности слогов. Таким образом, оценка правдоподобия (Q) для отдельного предложения определяется по следующей формуле:

$$Q = \prod_{i=1}^{N-2} \left(q_i \cdot \prod_{j=1}^{3} p_i^j \right) \cdot \frac{\sum_{h=1}^{N} p_h}{N},$$

где q_i – частота i-й триграммы; p_i^j – коэффициенты подобия входящих в триграмму слов; N – количество слов; S – количество слогов; p_h – коэффициент подобия слова под номером h. Множитель под внешним знаком произведения заменяется на 1, если соответствующая триграмма отсутствует или произведение коэффициентов подобия меньше некоторого заданного ограничения.

Такой способ вычисления позволяет выявлять предложения с хорошей статистикой, но имеющие неизвестные слова. Таким образом, если диктор произнесет слово, не входящее в состав словаря, алгоритм представит его в виде набора существующих слов, либо укажет, что данное слово является неизвестным при том условии, что остальные слова дают предложению максимальную оценку.

Для сокращения времени поиска воспользуемся в очередной раз методом динамического программирования. На первом этапе для каждой пары, состоящей из двух предпоследних слов, выбирается слово из последнего списка так, чтобы значение оценки правдоподобия было как можно больше. После получения значения для каждой пары предпоследних слов происходит переход к рассмотрению следующей пары, находящейся ближе к началу предложения. Аналогичным образом происходит выбор третьего слова, но теперь уже учитываются накопленные значения от предыдущих пар, которые следует перемножить с вычисленным для данной тройки значением. Продолжив рекуррентный процесс, по достижении начала предложения получим требуемую оценку. Пример отыскания предложения с неполными списками для слов изображен на рисунке (в рамках – слова с начальными формами).

Итак, полный алгоритм получения оценки правдоподобия составленного предложения описывается следующим образом.

1. Для каждого слова, входящего в предложение, проверить, был ли ранее составлен список подобных слов с коэффициентами подобия. Если такого списка нет:

– если слово находится в словаре, то добавить его в список без изменений с коэффициентом подобия 1;

– с помощью алгоритма Вагнера–Фишера, адаптированного к данной задаче, найти в словаре, представленном в виде trie, все слова, расположенные в пределах заданного расстояния.

2. С помощью метода динамического программирования выбрать те слова из списков, которые образуют предложение с наибольшей оценкой Q. Оценка полученного предложения является искомой оценкой.

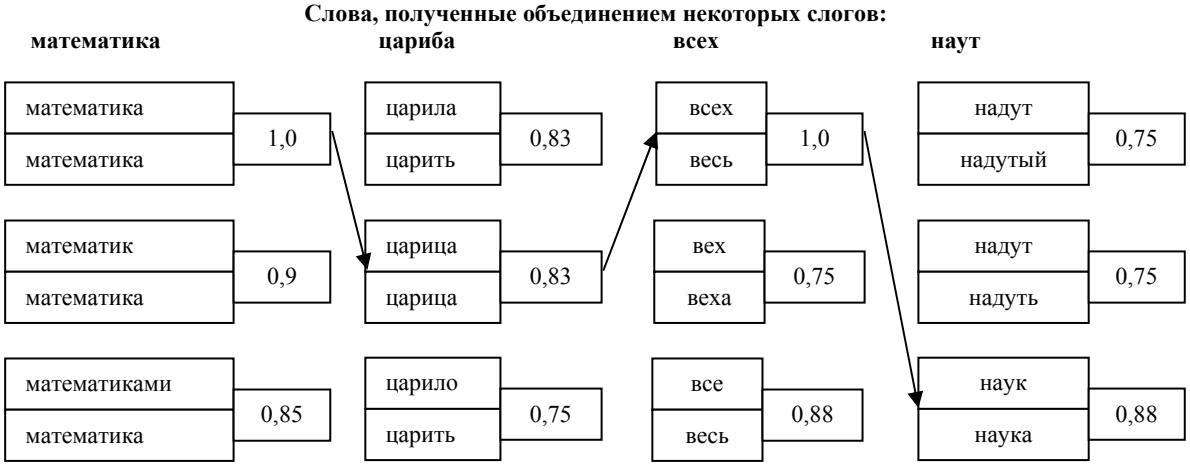

Иллюстрация процесса поиска наиболее вероятного предложения

Как говорилось выше, рассмотрение всех комбинаций слов для входящей цепочки слогов за приемлемое время является невозможным, поэтому предложен стохастический алгоритм оптимизации, который основывается на идеях генетических алгоритмов (ГА).

Коэволюционный асимптотический генетический алгоритм. Одной из основных проблем при применении генетических алгоритмов является настройка его параметров, от которой существенно зависит эффективность работы. Стандартный ГА имеет как минимум три метода селекции (пропорциональная, турнирная и ранговая), три метода рекомбинации (одноточечная, двухточечная и равномерная) и несколько уровней мутации, требующих дополнительной настройки. Количество возможных комбинаций достигает многих десятков. Полный перебор комбинаций настроек требует чрезмерно много времени и вычислительных ресурсов и не может быть применен на практике. Выбор настроек наугад также является плохой идеей, так как эффективность ГА на одной и той же задаче может варьироваться от нуля до 100 % в зависимости от выбора параметров.

Существует несколько подходов к решению этой проблемы. Например, прямое кодирование параметров ГА нижнего уровня, решающего задачу оптимизации, в хромосому мета-ГА более высокого уровня, решающего задачу настройки параметров [10]. Другой подход состоит в гибридизации ГА со специальным алгоритмом локального поиска в бинарном пространстве [11]. В некоторых приложениях это приводит к низкой чувствительности эффективности ГА к выбору параметров [12; 13], что позволяет избежать их настройки. Существуют и другие подходы.

Так, в [14] была предложена адаптация стратегии ГА за счет конкурирующих подпопуляций. Каждая подпопуляция имеет собственную стратегию (комбинацию параметров алгоритмов). Перераспределение ресурсов обеспечивает доминирование подпопуляции с наиболее подходящей для решаемой задачи стратегией поиска. Этот подход, а также предложенный [15; 16], может рассматриваться как один из примеров коэволюционных генетических алгоритмов, отличающихся идеями организации взаимодействия стандартных генетических алгоритмов.

В [17] рассматривается несколько иной подход, использующий как конкуренцию, так и сотрудничество индивидуальных ГА, имеющих каждый свою комбинацию параметров. Сотрудничество индивидуальных ГА достигается за счет миграции лучших решений во все подпопуляции, что обеспечивает рост эффективности коэволюционного алгоритма за счет положительного эффекта взаимодействия подпопуляций. Проблемой применения коэволюционных алгоритмов является неопределенность выбора кандидатов для включения в коэволюцию. Число таких кандидатов не может превышать полтора-два десятка алгоритмов, в то время как выбирать их надо из сотен.

Еще одним полезным подходом к решению проблемы выбора эффективных настроек ГА является сокращение количества настраиваемых параметров. В [18] был предложен так называемый вероятностный ГА, в котором операция скрещивания была заменена на генерирование потомков в соответствии с распределением вероятностей их компонент. Распределение вероятностей изменялось таким образом, чтобы возрастала вероятность правильного значения гена на конкретном месте в хромосоме. Очевидно, что выбор типа рекомбинации в таком алгоритме не требуется, что сокращает количество возможных комбинаций параметров в три раза.

Еще меньше настраиваемых параметров содержит асимптотический вероятностный генетический алгоритм [19], в котором применяется адаптивная мутация, не требующая предварительной настройки. Дополнительным преимуществом асимптотического алгоритма является способ выполнения генетических операторов, который сводится к изменению распределения вероятностей компонент, что значительно сокращает время на проработку алгоритма. Фактически асимптотический вероятностный ГА имеет всего несколько комбинаций настроек параметров.

В данной работе применен разработанный авторами коэволюционный подход к организации работы асимптотического вероятностного ГА (КАГА), благодаря которому проблема выбора типов применяемых генетических операторов отсутствует.

Данный алгоритм сначала был протестирован на стандартном наборе тестовых функций безусловной оптимизации (например, функций Растригина, Розенброка, Катковника, Дейонга, Экли и др.). Результаты тестирования были сравнены с результатами работы индивидуальных стандартных генетических алгоритмов при равных вычислительных ресурсах (произведение количества индивидов в популяции на число поколений). Усреднение проводилось по 100 запускам каждого алгоритма.

Исследования показали, что на большинстве задач предложенный подход по надежности не уступает наилучшему индивидуальному стандартному ГА (табл. 1) и превосходит его по скорости выполнения. Здесь под надежностью понимается отношение числа запусков, в которых с заданной точностью найден известный оптимум, к общему числу запусков. Скорость – средний номер поколения, на котором впервые в прогоне найден оптимум.

Таким образом, коэволюционный асимптотический генетический алгоритм показывает высокую надежность и эффективность на большинстве тестовых задач и не уступает стандартному генетическому алгоритму. Главное его преимущество заключается в автоматической настройке практически всех параметров, что значительно освобождает работу алгоритма от человеческого фактора и повышает эффективность поиска решения.

Применение асимптотического генетического алгоритма в системе распознавания речи. Для задачи распознавания речи индивид представляет собой бинарную строку, размер которой зависит от количества слогов предложения. Индивид содержит информацию о том, какие слоги следует объединить и где находятся границы слов; так, каждая единица или ноль, стоящие на i-м месте, будут означать, что слоги s_i и s_{I+1} необходимо объединить или отдать разным словам соответственно. Таким образом, длина индивида на единицу меньше количества слогов.

Алгоритм был освобожден от повторного выполнения операций для одинаковых входных данных. Например, был создан ассоциативный массив, ключом которого является предложение, а значением его оценка. Аналогичный массив был создан для триграмм, встречавшихся в предложении, что дало значительный выигрыш по времени, поскольку поиск триграмм по большой базе производится довольно долго, а схожие триграммы присутствуют во многих предложениях, составляемых генетическим алгоритмом.

Эффективность алгоритма была проверена на тестовом наборе предложений. Предварительно ряды распознанных слогов были поделены на 2 класса: полученные в результате распознавания подготовленной, четкой речи, не содержащие ошибок, и речи спонтанной, содержащие до 15 % ошибок. Каждый класс, в свою очередь, разделен на три группы, различающиеся длиной входящих в них цепочек. Усреднение проводилось для каждой из групп, содержащих по 20 различных цепочек слогов.

Результаты данного исследования показали, что для подготовленной речи алгоритм представляет входящую последовательность слогов в виде предложения с незначительным содержанием ошибочных слов (табл. 2). При этом КАГА работает устойчиво, о чем можно судить по среднеквадратичному отклонению от среднего номера поколения, на котором алгоритм находит решение. Для спонтанной речи получаемые предложения также содержат в среднем незначительное количество ошибок относительно изначальной погрешности в ряде слогов (табл. 3). Однако часто в отдельных составленных предложениях погрешность уменьшается или ошибочные слова вовсе отсутствуют, т. е. алгоритм способен повышать качество распознавания.

Таблица 1

Результаты тестирования. Сравнение КАГА с индивидуальными алгоритмами

Функция	Надежность лучшего (на данной функции) индивидуального ГА	Скорость ГА	Надежность КАГА	Скорость КАГА
AdditivePotential	1	2 160	1	1 062
DeJong2	1	3 040	1	2 226
GaussianQuartic	1	2 990	1	1 861
Griewank	0,98	1 428	0,6	3 813
Ackley	1	2 745	1	2 014
Ackley2	1	2 775	1	1 850
Griewank2	1	2 130	1	2 146
Himmelblau	0,72	6 764	0,4	2 237
HyperEllipsoid	1	2 595	1	2 180
HyperEllipsoid2	1	3 875	1	2 073
Rastrigin	1	3 490	1	3 881
Katkovnik	1	3 310	1	2 278
Multiextremal3»	1	3 415	1	1 747
Multiextremal4	1	4 330	1	1 838
MultiplicativePotential»	1	1 845	1	1 335
Rastrigin2	1	2 415	1	1 964
Rastrigin3	1	3 425	0,98	4 259
RastriginWithTurning	1	2 970	1	1 948
Rosenbrock	0,36	6 666	0,44	6 502
SphereModel	1	2 880	1	1 861
Средние значения	0,953	3 262	0,921	2 454

Таблица 2

Результаты тестирования. Предложения подготовленной речи

Входящие цепочки содержат 0 % ошибок					
Длина предложения	Ошибка распознавания, %	Время поиска, с	Количество поколений	Среднеквадратичное отклонение поколений	Количество рассмотренных индивидов
9…20	7	3	2,4	0,5	860
21…50	4	8	5,3	2,9	3 058
51…80	5	41	7,5	3	11 307

Таблица 3

Результаты тестирования. Предложения спонтанной речи

Входящие цепочки содержат 8...15 % ошибок					
Длина предложения	Ошибка распознавания, %	Время поиска, с	Количество поколений	Среднеквадратичное отклонение поколений	Количество рассмотренных индивидов
9…20	20	6	2,6	0,7	1 044
21…50	11	11	7,5	4,9	4 121
51…80	9	47	11	5,2	11 740

Для предложений небольшой длины правильность найденных решений была проверена с помощью алгоритма полного перебора. В отличие от полного перебора предложенный алгоритм находит решение, рассматривая значительно меньшее количество комбинаций слогов, выигрывая таким образом по времени выполнения.

Также проверена способность алгоритма искать решение для предложения, изначально содержащего неизвестные слова, не входящие в состав словаря. В одних случаях алгоритм выделяет неизвестные слова, в других – представляет их в виде ряда более коротких слов, что также является приемлемым результатом для таких предложений.

Таким образом, предложен алгоритм, выполняющий задачи последнего этапа в системах распознавания речи для модели слогового представления слов русского языка. Кроме того, разработан коэволюционный асимптотический генетический алгоритм решения сложных задач безусловной оптимизации, применяемый к решению поставленной задачи. Сейчас разработанная система может быть применена для преобразования в текст речи, произнесенной на заседаниях, конференциях, для документирования архивных звуковых записей, т. е. в областях, не требующих чрезвычайно быстрого отклика системы. В дальнейшем планируется модернизировать алгоритмы, внести дополнительные преобразования, чтобы оптимизировать приложение по времени выполнения. Работая в режиме реального времени, система могла бы принести пользу в самых различных сферах деятельности, в том числе могла бы оказать помощь людям с ограниченными возможностями.

Библиографические ссылки

1. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner / V. Siivola, T. Hirsimäki, M. Creutz, M. Kurimo // Proc. of the 8th European Conf. on Speech Communication and Technology (Eurospeech). Geneva, Switzerland, 2003. P. 2293–2296.

2. Rabiner L., Juang B.-H. Fundamentals of Speech Recognition. M. : Prentice Hall, 1993.

3. Карпов А. А. Модели и программная реализация распознавания русской речи на основе морфемного анализа : дис. … канд. техн. наук. СПб., 2007.

4. Зализняк А. А. Грамматический словарь русского языка: словоизменение. 4-е изд., испр. и доп. М. : Рус. слов., 2003.

5. Кипяткова И. С., Карпов А. А. Модуль фонематического транскрибирования для системы распознавания разговорной русской речи // Искусств. интеллект. 2008. № 4. P. 747–757.

6. Холоденко А. Б. О построении статистических языковых моделей для систем распознавания русской речи // Интеллектуал. системы. 2001. Т. 6. Вып. 1–4. С. 381–394.

7. Wagner R. A., Fischer M. J. The string-to-string correction problem // J. ACM. 1974. Vol. 21. № 1. P. 168–173.

8. Knuth D. E. The art of computer programming. Vol. 3. Sorting and Searching. Reading, Mass. : Addison–Wesley, 1973.

9. Kukich K. Techniques for automatically correcting words in text // ACM Computing Surveys. 1992. Vol. 24. № 4. P. 377–439.

10. Eiben A., Hinterding R., Michalewicz Z. Parameter control in evolutionary algorithms // IEEE Trans. Evolutionary Algorithms. 1999. Vol. 3. P. 124–141.

11. Antamoshkin A., Semenkin E. Local Search Efficiency when Optimizing Unimodal Pseudoboolean Functions // Informatica. 1998. Vol. 9. № 3. P. 279–296.

12. Бежитский С. С., Семенкин Е. С., Семенкина О. Э. Гибридный эволюционный алгоритм для выбора эффективных вариантов систем управления космическими аппаратами // Автоматизация и соврем. технологии. 2005. № 11.

13. Yakimov Y., Semenkin E., Yakimov I. Hybrid genetic algorithm for a full-profile analysis of XRD powder patterns // Acta Cryst. A64. C226. 2008.

14. Schlierkamp-Voosen D., Mühlenbein H. Strategy adaptation by competing subpopulations // Parallel Problem Solving from Nature III. Springer-Verlag, 1994.

15. Potter M. A., De Jong K. A. Cooperative coevolution: an architecture for evolving coadapted subcomponents // Trans. Evolutionary Computation. 2000. Vol. 8. P. 1–29.

16. Rosin C., Belew R. New Methods for Competitive Coevolution // Trans. Evolutionary Computation. 1997.

17. Sergienko R., Semenkin E. Competitive Cooperation for Strategy Adaptation in Coevolutionary Genetic Algorithm for Constrained Optimization // IEEE World Congress on Computational Intelligence (WCCI'2010). Barcelona, Spain, 2010. P. 1626–1631.

18. Семенкин Е. С., Сопов Е. А. Вероятностные эволюционные алгоритмы для оптимизации сложных систем // Сб. тр. междунар. конф. «Intelligent systems» (AIS’05) and «Intelligent CAD» (CAD-2005) : в 3 т. Т. 1. М. : Физматлит, 2005.

19. Galushin P. V., Semenkin E. S. Asymptotic probabilistic genetic algorithm // Vestnik. Sci. J. of Siberian State Aerospace Univ. named after academician M. F. Reshetnev. Vol. 5 (26). 2009. P. 45–49.