

На правах рукописи



ПРУЦКОВ Александр Викторович

**МОДЕЛИ, МЕТОДЫ И ПРОГРАММЫ
АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ФОРМ СЛОВ
В ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ИНТЕРФЕЙСАХ**

**05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей**

**Автореферат
диссертации на соискание ученой степени
доктора технических наук**

Рязань 2015

Работа выполнена в Федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Рязанский государственный радиотехнический университет» (ФГБОУ ВПО «РГРТУ»).

Научный консультант:

Пылькин Александр Николаевич,
доктор технических наук, профессор, заслуженный работник высшей школы РФ, заведующий кафедрой вычислительной и прикладной математики ФГБОУ ВПО «РГРТУ», г. Рязань

Официальные оппоненты:

Зотов Игорь Валерьевич,
доктор технических наук, профессор, профессор кафедры информационных систем и технологий ФГБОУ ВПО «Юго-Западный государственный университет», г. Курск

Никульчев Евгений Витальевич,

доктор технических наук, профессор, проректор по научной работе, заведующий кафедрой программных систем НОУ ВО «Московский технологический институт», г. Москва

Хранилов Валерий Павлович,

доктор технических наук, доцент, заместитель директора Института радиоэлектроники и информационных технологий ФГБОУ ВПО «Нижегородский государственный технический университет им. Р.Е. Алексеева», г. Нижний Новгород

Ведущая организация:

ФГБОУ ВПО «Тамбовский государственный технический университет»

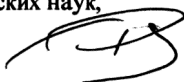
Защита диссертации состоится 23 декабря 2015 г. в 12 часов 00 минут на заседании диссертационного совета Д 212.211.01 на базе ФГБОУ ВПО «РГРТУ» по адресу: 390005, Рязань, ул. Гагарина, 59/1.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВПО «РГРТУ» или на сайте <http://rsgeu.ru>.

Автореферат разослан « ____ » сентября 2015 г.

Ученый секретарь

диссертационного совета
кандидат технических наук,
доцент



Пржегорлинский Виктор Николаевич

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования и степень ее разработанности.

Взаимодействие пользователя с ЭВМ на естественном языке предполагает автоматическую обработку текстов (АОТ) для понимания запроса пользователя и формирования ответа на них. При этом выдвигается требование, чтобы диалог мог вестись на нескольких языках. Это требование обусловлено двумя тенденциями современного мира, способствующими образованию многоязычной среды. Во-первых, появление межгосударственных союзов и объединений (например, Европейского Союза). Во-вторых, повышение статуса языков национальных меньшинств на государственном и региональных уровнях (Индия, ЮАР).

Для реализации многоязычных интерфейсов необходимо решать задачи машинного перевода, индексации и анализа текстов на нескольких языках. Однако существующие методы решения перечисленных задач ориентируются на один или несколько языков. Поэтому возникает необходимость разработки универсальных методов АОТ различных уровней, в том числе и морфологического уровня.

Данные тенденции, а также глобализация обостряют необходимость в специалистах со знанием нескольких языков. Поэтому актуальным становится разработка автоматизированных обучающих систем (АОС) по различным разделам языкознания, в том числе и морфологии естественных языков. Универсальные методы различных уровней АОТ могут быть использованы для построения перспективных систем проверки знаний, входящих в состав любой АОС, с динамической генерацией вариантов заданий.

Значительный вклад в развитие естественно-языковых интерфейсов и АОТ внесли такие ученые России, как Ю.Д. Апресян, Г.Г. Белоногов, В.М. Брябрин, О.С. Кулагина, А.А. Ляпунов, М.Г. Мальковский, Ю.Н. Марчук, И.А. Мельчук, А.С. Нариньяни, Р.Г. Пиотровский, Э.В. Попов, Д.А. Пospelов, В.А. Фомичев и др., а также зарубежные специалисты Т. Виноград (Т. Winograd), В.А. Вудс (W.A. Woods), К. Коскенниemi (K. Koskenniemi), М. Портер (M. Porter), Г. Хардегри (G. Hardegree), Н. Хомский (N. Chomsky), Р. Шенк (R. Schank) и др.

В большинстве научных работ перечисленных исследователей, связанных с разработкой методов морфологического уровня АОТ, рассматривается только один естественный язык или даже естественный язык с некоторыми ограничениями. В остальных работах, посвященных разработке многоязычных методов, возможность обработки других языков, кроме рассмотренных, теоретически не доказывается. Вопросы, связанные с обработкой числительных, в научных работах практически не обсуждаются, хотя эта проблема имеет важное значение в задачах машинного перевода, обработки речи и в обучении естественным языкам.

Алгоритмы решения задач описываются в рамках алгоритмических систем (моделей). Одной из алгоритмических моделей являются нормальные алгоритмы, предложенные А.А. Марковым в середине прошлого века. Однако их исследования продолжают, о чем свидетельствуют современные публикации. Исследованию нормальных алгоритмов Маркова посвящены работы Н.М. Нагорного, И.А. Цветкова, Г.С. Цейтина и др.

Объект исследования: методы морфологической обработки текстов и способы описания формообразования естественных языков.

Предмет исследования: модели и алгоритмы генерации и определения форм слов естественных языков различных групп и семейств с их последующей реализацией в виде программных систем.

Цель и задачи исследования. Целью диссертационной работы является разработка моделей, методов и алгоритмов, позволяющих описывать и реализовывать формообразование полной парадигмы слов и обрабатывать числительные, универсальных для естественных языков различных групп и семейств, с их последующей реализацией в программных системах естественного-языкового человеко-машинного взаимодействия, АОТ и автоматизации обучения.

Для достижения цели необходимо решить следующие задачи:

1) разработать модель формообразования и метод генерации и определения полной парадигмы слов естественных языков различных групп и семейств;

2) разработать обобщенную трехуровневую модель числительного и метод обработки количественных числительных естественных языков различных групп и семейств, упрощающий и упрощающий операции с ними;

3) исследовать нормальные алгоритмы Маркова, выявить их ограничения и предложить метод устранения этих ограничений;

4) реализовать предложенные модели и методы в программных системах морфологического анализа и синтеза форм слов и обработки числительных, а также системах проверки знаний с динамической генерацией вариантов заданий.

Методология и методы исследования. Для решения поставленных задач использовались элементы теорий графов, множеств, унификации, формальных грамматик и алгоритмов. Для практической проверки работоспособности предложенных методов использовалось разработанное программное обеспечение.

Основные научные результаты, полученные автором и выносимые на защиту.

1. Модель, позволяющая описывать формообразование полной парадигмы слов естественных языков различных групп и семейств.

2. Алгоритмы генерации и определения форм слов на основе предложенной модели формообразования и алгоритм, позволяющий автоматизировать построение цепочек преобразований на основе предложенной классификации образования форм слов.

3. Трехуровневая обобщенная модель числительного, являющаяся промежуточным этапом при обработке числительных и позволяющая сократить трудоемкость этих операций и упростить добавление новых языков.

4. Алгоритмы, позволяющие преобразовывать количественные числительные к предложенной модели числительного и обратно.

5. Метод устранения ограничения нормальных алгоритмов Маркова путём введения линейного вычислительного процесса и модификация нормальных алгоритмов Маркова, реализующая этот метод и позволяющая разрабатывать алгоритмы с линейной трудоемкостью.

6. Программные системы генерации и определения форм слов и обработки числительных, реализующие предложенные модели и алгоритмы и подтверждающие их адекватность и работоспособность.

Научная новизна. В диссертационной работе были получены следующие результаты.

1. Предложена модель, описывающая формообразование в виде цепочек преобразований, и в отличие от существующих моделей впервые доказана ее применимость к языкам различных групп и семейств. На основе данной модели разработаны алгоритмы, генерирующие и определяющие словоформы и составляющие вместе с моделью метод генерации и определения словоформ. Предложен алгоритм, автоматизирующий построение цепочек преобразований и упрощающий описание формообразование в терминах модели оператором-лингвистом.

2. Разработаны трехуровневая обобщенная модель числительного, являющаяся промежуточным этапом в операциях с количественными числительными различных языков и позволяющая сократить трудоемкость данных операций, а также алгоритмы преобразования числительных, записанные с помощью нормальных алгоритмов Маркова, составляющие метод обработки числительных. Введение промежуточного этапа упростило добавление новых языков в отличие от аналогичных подходов.

3. Выявлено ограничение нормальных алгоритмов Маркова, состоящее в невозможности реализовывать линейный вычислительный процесс, и предложен метод устранения данного ограничения. Разработана модификация нормальных алгоритмов Маркова (линейные нормальные алгоритмы), реализующая данный метод, и в ее рамках реализованы алгоритмы, сокращающие трудоемкость решения задач обращения и удвоения и число подстановок в схеме.

4. Разработано Интернет-приложение для обработки и перевода количественных числительных русского, английского, немецкого, испанского и финского языков через промежуточный этап в виде трехуровневой обобщенной модели. Интернет-приложением пользуются ежемесячно более 1 000 пользователей со всех постоянно обитаемых континентов, в том числе из более 200 университетов США, России, Канады и других стран. На основе предложенных методов генерации и определения форм слов и обработки числительных разработаны про-

граммные системы проверки знаний с динамической генерацией вариантов заданий и пояснений к неправильным ответам.

Теоретическая значимость работы заключается в разработке моделей и методов морфологической обработки текстов на естественных языках различных групп и семейств, исследовании нормальных алгоритмов Маркова и разработке их модификации, позволяющей устранить ограничения этой алгоритмической модели.

Практическая значимость работы. Разработанные методы и алгоритмы обработки форм слов и количественных числительных являются универсальными и позволяют решать соответствующие задачи для различных языков. Данные методы могут быть использованы как составные части многоязычных методов и систем АОТ, что позволит их упростить и уменьшить время разработки за счет универсальности предложенных методов. С помощью разработанной модификации нормальных алгоритмов Маркова были описаны некоторые алгоритмы данной работы. Результаты исследования могут быть использованы в учебных курсах по системам искусственного интеллекта и компьютерной лингвистике. Ежемесячно Интернет-приложение, разработанное на основе метода обработки числительных, используют более 1 000 человек из более 80 стран мира.

Достоверность научных положений, теоретических выводов и практических результатов диссертационной работы подтверждается соответствием результатов моделирования разработанных методов и алгоритмов правилам морфологии естественных языков, результатами работы программных средств, реализованных на основе предложенных методов, используемых во всем мире и имеющих государственную регистрацию в ФГУ «Федеральный институт промышленной собственности Федеральной службы по интеллектуальной собственности, патентам и товарным знакам» (ФГУ ФИПС, РОСПАТЕНТ), актами внедрения результатов диссертационной работы.

Реализация и внедрение результатов диссертационной работы. Диссертационная работа включает исследования, выполненные в Рязанском государственном радиотехническом университете (РГРТУ) в рамках госбюджетных НИР № 9-07Г «Разработка математических моделей, методов и алгоритмов обработки больших потоков информации в сложно организованных вычислительных структурах», № 8-08Г «Разработка теоретических основ автоматизации семантического анализа текстовых документов, проходящих правовую экспертизу», № 7-09Г «Разработка математических методов и алгоритмов передачи и обработки цифровой информации для поддержки интеллектуальных систем управления», № 6-11Г «Разработка теоретических основ управления качеством социально-образовательных отношений и необходимого правового и информационного обеспечения», № 1-12Г «Разработка и создание интеллектуального автоматизированного комплекса управления ресурсами в динамических информационно-телекоммуникационных системах», № 11-12Г «Разработка математических моделей, методов и алгоритмов обработки больших объемов

информации в сложно организованных системах искусственного интеллекта», в которых автор работы являлся исполнителем.

Метод генерации и определения форм слов и метод обработки количественных числительных были использованы в системе автоматизированного анализа коллективных договоров, разработанной в ведомственной лаборатории автоматизированного анализа коллективно-договорных актов образования, учредителем которой является Центральный Совет профсоюза работников народного образования и науки РФ, для сравнения их статей и, в случае их изменения, определения степени их различия. Использование данных методов позволило ускорить работу экспертов с данной системой по анализу статей коллективных трудовых договоров и эффективность системы в целом.

Модель формообразования, трехуровневая обобщенная модель числительного, метод генерации и определения форм слов и метод обработки числительных были реализованы в интерфейсе экспертной сети для переводчиков My-Polyglot.com, разрабатываемой компанией ТИСС (Белгородская область). Предложенные модели и методы позволили добавить в интерфейс полезные инструменты для перевода. Немаловажным преимуществом этих моделей и методов является их независимость от языка.

Метод генерации и определения форм слов использовался в агентстве недвижимости ООО «ИНКОМ-Сокол» (г. Москва) для ввода и анализа текста объявлений о купле, продаже и аренде объектов недвижимости. Применение разработанного метода позволило реализовать запросы к базе недвижимости не только с помощью выбора параметров, но и на естественном языке, что упростило работу с базой.

Системы проверки знаний, разработанные на основе методов генерации и определения форм слов и обработки числительных, используются в учебном процессе учебных заведений г. Рязани: филиала Московского государственного университета экономики, статистики и информатики, филиала НОУ ВПО «Московский институт экономики, менеджмента и права», МОУ «Средняя школа №45» и МОУ «Средняя общеобразовательная школа №49». Результаты диссертационной работы внедрены в учебный процесс РГРТУ для студентов специальности 230105 – «Программное обеспечение вычислительной техники и автоматизированных систем» в курсах «Математическая логика и теория алгоритмов», «Системы искусственного интеллекта» и «Проектирование систем искусственного интеллекта», для студентов специальности 080801 – «Прикладная информатика (в экономике)» в курсе «Интеллектуальные информационные системы».

Интернет-приложение обработки количественных числительных, которое доступно всем пользователям сети Интернет, используется в более 200 университетах России, США, Канады и других стран.

Апробация результатов работы. Научные результаты работы докладывались и обсуждались на 9-10, 12-16-й Международных научно-технических конференциях «Проблемы передачи и обработки информации в сетях и системах телекоммуникаций» (г. Рязань, 2000-2001,

2004-2005, 2008, 2010 гг.), 7-й Всероссийской научной конференции студентов и аспирантов «Техническая кибернетика, радиоэлектроника и системы управления» (г. Таганрог, Ростовская область, 2004 г.), 9-16, 18-й Всероссийских научно-технических конференциях студентов, молодых ученых и специалистов «Новые информационные технологии в научных исследованиях и в образовании» (г. Рязань, 2004-2011, 2013 гг.), Международной научно-практической конференции «Электронные средства и системы управления» (г. Томск, 2004 г.), 30, 32-34-й Всероссийских научно-практических семинарах и научно-технических конференциях «Сети, системы связи и телекоммуникации», «Информационные и телекоммуникационные технологии. Подготовка специалистов для инфокоммуникационной среды» (г. Рязань, 2005, 2007-2009 гг.), Научной конференции, посвященной 80-летию Московского государственного университета печати (г. Москва, 2009 г.), Научно-практической конференции «Традиции и инновации в лингвистике и лингвообразовании» (г. Арзамас, Нижегородская область, 2011 г.), 27-й Международной научно-технической конференции «Математические методы и информационные технологии в экономике, социологии и образовании» (г. Пенза, 2011 г.), 11-й Международной научно-практической конференции «Интеллект и наука» (г. Железногорск, Красноярский край, 2011 г.), 2-й Всероссийской научно-методической конференции «Инновации и традиции науки и образования» (г. Сыктывкар, Республика Коми, 2011 г.), 3-й Всероссийской научно-методической конференции «Методы обучения и организация учебного процесса в вузе» (г. Рязань, 2013 г.), 12-й Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых «Техника XXI века глазами молодых ученых и специалистов» (г. Тула, 2013 г.), 5-6-й Межвузовских школах-семинарах «Задачи системного анализа, управления и обработки информации» (г. Москва, 2014-2015 гг.), International Conference on Computer Technologies in Physical and Engineering Applications 2014 (ICCTPEA-2014) (г. Санкт-Петербург).

Публикации. По теме диссертации опубликованы 88 печатных работ, из них 26 в соавторстве, включая 1 работу в базе данных Scopus. В их числе 23 статьи в изданиях из «Перечня ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней кандидата и доктора наук», 1 статья в издании на английском языке, 27 статей в научных журналах и межвузовских сборниках, 30 тезисов докладов Международных и Всероссийских конференций и семинаров, 7 свидетельств о государственной регистрации программы для ЭВМ в РОСПАТЕНТе.

Личный вклад автора. Все результаты диссертационной работы, в том числе постановка задач, разработка и исследование предложенных методов, основные научные результаты и выводы принадлежат лично автору. Программное обеспечение предложенных методов разработано под руководством и при непосредственном участии автора. Участие

соавторов заключалось в консультациях, совместной разработке программных средств и получении экспериментальных результатов на основе предложенных автором методов.

Структура и объем диссертационной работы. Диссертационная работа общим объемом 279 страниц состоит из введения, пяти глав, содержащих 47 рисунков и 17 таблиц, заключения, списка литературы из 193 наименований и приложений.

Соответствие паспорту специальности. Диссертационная работа представляется по специальности 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» и соответствует п. 7, поскольку в ней предложены модели и методы морфологической обработки текстов, являющиеся составной частью любого естественно-языкового человеко-машинного интерфейса.

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность темы, сформулированы цель и задачи диссертационной работы, показаны научная новизна и практическая ценность диссертационной работы.

В **главе 1** рассмотрены задачи и системы АОТ, как основного этапа реализации естественно-языковых интерфейсов, и сделан вывод о широком использовании в них этапов морфологического анализа и синтеза, что делает морфологическую обработку текста значимым этапом АОТ.

В **главе 2** предложен метод генерации и определения форм слов естественных языков различных семейств и групп.

На этапе морфологической обработки текстов решаются задачи генерации и определения форм слов. Генерация формы слова (синтез, продукция) – процесс получения формы с использованием в качестве начальных параметров основы и грамматического значения. Определение формы слова (анализ, распознавание, интерпретация) – процесс, обратный генерации. Определение заключается в нахождении по данной словоформе ее нормальной формы (основы) и грамматического значения.

В главе проведен анализ различных методов, предложенных отечественными и зарубежными авторами, выявлены их преимущества и недостатки. На основе данного анализа сформулировано требование к разрабатываемому методу генерации и определения форм слов, которому существующие методы не удовлетворяют. Метод должен быть универсальным по следующим критериям: универсальность генерации и определения форм слов естественных языков различных групп и семейств; универсальность структуры словарей, не требующей конвертации для решения задач определения или генерации; универсальность метода для всех видов формообразования словоформ, обработки всей парадигмы слова, в том числе и аналитических словоформ.

Предлагается модель формообразования и метод генерации и определения форм слов, который удовлетворяет перечисленным кри-

териям и включает также алгоритмы генерации и определения, разработанные на основе этой модели.

Модель формобразования естественного языка заключается в том, что получение любой словоформы с данным грамматическим значением можно представить в виде последовательности конечного числа преобразований над основой.

Можно выделить два основных типа преобразований строк: 1) добавление подстроки P к строке слева (префикс) (обозначается $P+$) или справа (постфикс) ($+P$) без изменения самой строки; 2) замена в строке первого слева вхождения подстроки H на подстроку P ($H \rightarrow P$).

Каждое преобразование имеет обратное к нему, то есть совершающее обратное действие. Основным типам преобразований соответствуют следующие типы обратных преобразований: 1) отделение подстроки P от строки слева ($P-$) или справа ($-P$); 2) обратная замена первой слева подстроки P на подстроку H ($H \leftarrow P$).

Данная модель является открытой и может быть расширена другими типами преобразований.

Пусть преобразование Q – прямое, а преобразование Q' – обратное данному. Цепочка преобразований (прямая цепочка преобразований, комбинированное преобразование) – это конечная последовательность преобразований:

$$R = (Q_1, Q_2, \dots, Q_n).$$

Обратная цепочка преобразований R' представляет собой обратную последовательность преобразований, обратных данным:

$$R' = (Q'_n, Q'_{n-1}, \dots, Q'_1).$$

Прямая цепочка преобразует основу слова S в форму слова F , а обратная – форму F в основу S .

Преобразование и цепочка преобразований должны обладать следующими свойствами: 1) однозначностью результата: преобразование или цепочка всегда приводят к одному и тому же результату; 2) обратимостью действия: применение к строке прямого, а затем обратного преобразования или цепочки не изменяет ее.

Обладание данными свойствами является обязательными условиями для правильной работы алгоритмов предлагаемого метода.

Применить преобразование Q к форме F означает выполнить преобразование Q над формой F , чтобы получить форму Φ . Применить цепочку преобразований R к форме F означает применить последовательно слева направо каждое преобразование Q , составляющее цепочку R , к форме F . Если одно преобразование цепочки R является неприменимым к форме F , то вся цепочка R считается неприменимой к форме F и процесс применения преобразований цепочки R к форме F прекращается. Если преобразование или цепочка преобразований применимы, то их результатом является преобразованная форма. Если преобразование или цепочка преобразований неприменимы к форме, то результат не определен.

Пример 1. Получение формы «*has been waiting*» глагола английского языка «*wait*» описывается прямой и обратной цепочками:

$$R = (Q_1, Q_2, Q_3) = (+ing; been_+; has_+);$$

$$(wait \Rightarrow waiting \Rightarrow been\ waiting \Rightarrow has\ been\ waiting);$$

$$R' = (Q'_3, Q'_2, Q'_1) = (has_ -; been_ -; -ing.);$$

$$(has\ been\ waiting \Rightarrow been\ waiting \Rightarrow waiting \Rightarrow wait). \square$$

Пример 2. Получение формы «*macht auf*» глагола немецкого языка «*auf|machen*» описывается прямой и обратной цепочками:

$$R = (auf\ -; -en; +t; +\ auf);$$

$$(aufmachen \Rightarrow machen \Rightarrow mach \Rightarrow macht \Rightarrow macht\ auf);$$

$$R' = (-\ auf; -t; +en; auf\ +).$$

$$(macht\ auf \Rightarrow macht \Rightarrow machen \Rightarrow aufmachen). \square$$

Данные цепочки из примеров применимы и для других слов, имеющих тот же тип формообразования.

Формы слова могут образовываться с помощью редупликации. Редупликация состоит в полном или частичном повторении основы или слова. Обозначим полную редупликацию основы (далее редупликация) через операцию $\times 2$. Обратная ей операция $/2$ будет состоять в разделении словоформы пополам и отбрасывании одной из половин. Редупликация обладает свойствами однозначности результата и обратимости действия и может использоваться в цепочках преобразований.

Пример 3. В малайском языке существительное *orang* – «человек» имеет форму множественного числа *orangorang* – «люди».

Образование этой словоформы можно описать следующими прямой и обратной цепочками:

$$R = (\times 2); \quad R' = (/2).$$

Используем эти цепочки:

$$orang (\times 2) = orangorang; \quad orangorang (/2) = orang. \square$$

Возможность реализации редупликации в модели формообразования показывает универсальность модели и ее преимущество над другими способами описания формообразования.

Опишем модель формообразования алгебраически. Пусть $E = \{\Psi, \Omega\}$ – алгебра, где Ψ – множество слов естественного языка, Ω – множество операций над словами.

Множество Ψ включает $\Psi = S_M \cup F_M \cup \{\emptyset\}$, где S_M – множество нормальных форм слов; F_M – множество остальных форм слов; $S_M \cap F_M = \emptyset$; \emptyset – пустое слово.

Пусть $x \in X$, $y \in Y$, $Z = X \times Y$. Обозначим через $\text{Im}(x, Y)$ – образ x в Y при соответствии Z и через $\text{pIm}(y, X)$ – прообраз y в X при соответствии Z .

Соответствие $L \subset S_M \times F_M$ обладает следующими свойствами:

$$1) |\text{Im}(S, F_M)| \geq 0, \text{ где } S \in S_M;$$

$$2) |\text{pIm}(F, S_M)| = 1, \text{ где } F \in F_M;$$

$$3) \forall S_i \forall S_j (\text{Im}(S_i, F_M) \cap \text{Im}(S_j, F_M) = \emptyset), \quad \text{где } S_i, S_j \in S_M;$$

$$i, j = 1, 2, \dots, N_S; i \neq j; N_S = |S_M|.$$

Множество Ω объединяет множество прямых операций Θ_M и множество обратных операций Θ'_M : $\Omega = \Theta_M \cup \Theta'_M$, причем $\Theta_M \cap \Theta'_M = \emptyset$; $|\Theta_M| = |\Theta'_M| = N_\Theta$.

Пусть множество $T = \{(\Theta_i, \Theta'_i) \mid i = 1, 2, \dots, N_\Theta\} \subset \Theta_M \times \Theta'_M$ – взаимно однозначное соответствие.

Операции множества Ω в дальнейшем будем называть преобразованиями. Тогда множество Θ_M включает основные типы преобразований: $\Theta_M = \{P^+; +P; H \rightarrow P\}$, а множество Θ'_M включает обратные преобразования: $\Theta'_M = \{P^-; -P; H \leftarrow P\}$.

Множество Θ_M может быть расширено другими преобразованиями, специфичными для естественного языка.

Пусть $Q \in \Omega$ – преобразование, а $Q' \in \Omega$ – преобразование, обратное данному. Пара преобразований Q и Q' обладает следующим свойством:

$$[(Q, Q') \in T] \otimes [(Q', Q) \in T] = 1, \quad (2.1)$$

где \otimes – логическая операция сложение по модулю 2 (исключающее или); 1 – логическая константа «истина».

Обозначим цепочку преобразований как $R = (Q_1, Q_2, \dots, Q_n) \in R_M$, где $Q_1, Q_2, \dots, Q_n \in \Omega$; $n \geq 1$; R_M – множество цепочек.

Обозначим обратную цепочку преобразований R' как $R' = (Q'_n, Q'_{n-1}, \dots, Q'_1) \in R_M$, где $Q'_1, Q'_2, \dots, Q'_n \in \Omega$.

Все пары преобразований Q_i и Q'_i , где $i = 1, 2, \dots, n$, в прямой и обратной цепочках обладают свойством (2.1).

Пусть $\mathfrak{R} \in R_M$. Обозначим применение к форме $A \in \Psi$ цепочки \mathfrak{R} как $A(\mathfrak{R})$. Результат этой операции $B \in \Psi$ определяется следующим выражением:

$$B = \begin{cases} A(\mathfrak{R}), & \text{если } (\mathfrak{R} \text{ применима к } A) \text{ и } (A(\mathfrak{R}) \in \Psi), \\ \emptyset, & \text{в остальных случаях.} \end{cases}$$

Цепочка \mathfrak{R} применима к форме A , если возможно применение всех преобразований цепочки к форме. Например, невозможно отделить постфикс из-за его отсутствия в форме A или невозможно произвести замену подстрок из-за отсутствия исходной подстроки в форме A .

Назовем нулевой цепочкой $R_0 \in R_M$ цепочку, не изменяющую форму A : $A = A(R_0)$.

Нулевая цепочка может быть реализована разными способами, например: $R_0 = (+ \emptyset)$ или $R_0 = (+B; -B)$. Нулевая цепочка используется с неизменяемыми нормальными формами.

Таким образом, множество R_M включает прямые цепочки, в том числе и нулевые, и обратные им.

Прямая цепочка связывает форму S с формой $F \in \text{Im}(S, F_M)$ при L : $\forall F \exists S \exists R (F = S(R))$.

Обратная цепочка связывает форму F с формой $S \in \text{pIm}(F, S_M)$ при L : $\forall F \exists S \exists R' (S = F(R'))$.

Для решения задачи генерации форм слов необходимо найти такую цепочку R , что $A = S(R)$, где $S \in S_M$; ($A \in \text{Im}(S, F_M)$ при L) или ($A = S$).

Решение задачи определения форм слов включает нахождение цепочки R' такой, что $S = A(R')$, где $S \in \text{rIm}(A, S_M)$ при L) или ($S = A$).

Свойство обратимости действия можно записать так: $A = (A(R))(R')$.

Равенство $B = A(\mathfrak{R}) = \emptyset$ может быть верно в следующих случаях:

- \mathfrak{R} – прямая цепочка, $A \in S_M$, но $B \notin \text{Im}(A, F_M)$ при L ;

- \mathfrak{R} – обратная цепочка, $A \in F_M$, но $B \notin \text{rIm}(A, S_M)$ при L ;

- цепочка \mathfrak{R} не обладают свойствами однозначности результата или обратимости действия.

Соотношение между мощностями множеств, как правило, имеет вид: $|R_M|/2 < |S_M| \ll |F_M|$.

На основе модели формообразования предложены алгоритмы решения задач генерации и определения форм слов.

Условные обозначения операций в алгоритмах: $Y = (X)$ – получить значение Y по значению X ; $B = A(R)$ – применить к форме A цепочку преобразований R и получить форму B .

Данные организованы следующим образом: одной основе S соответствует один тип основы T ; одному типу основы T соответствует несколько основ S ; одному сочетанию типа основы T и грамматическому значению G соответствует одна цепочка преобразований R ; одна цепочка преобразований R соответствует одному сочетанию типа основы T и грамматическому значению G .

Задачей алгоритма генерации форм слов (рисунок 1) является получение словоформы F , соответствующей основе S и грамматическому значению G . Результат генерации форм слов всегда однозначен.

Алгоритм генерации состоит из двух этапов. На первом этапе производится поиск необходимой цепочки преобразований R по входным данным: основе S и грамматическому значению G (блоки 1-2). Второй этап заключается в применении найденной цепочки R к основе S для получения требуемой формы F (блок 3).

Алгоритм определения форм слов (рисунок 2) определяет грамматическое значение G и основу S , соответствующие исходной форме F .

Алгоритм определения форм слов заключается в переборе всех n цепочек преобразований в цикле (блоки 1-6). На каждой итерации извлекаются очередная цепочка R_i и соответствующие ей тип основы T_i и грамматическое значение G_i (блок 2).

Цепочка R_i' , обратная цепочке R_i , применяется к входной словоформе F для получения основы S_i (блок 3). Если найденной основе S_i соответствует тот тип основы T_i , что и очередной цепочке (блок 4), то результат, состоящий из полученной основы S_i и грамматического значения G_i , добавляется к полученным результатам (блок 5).

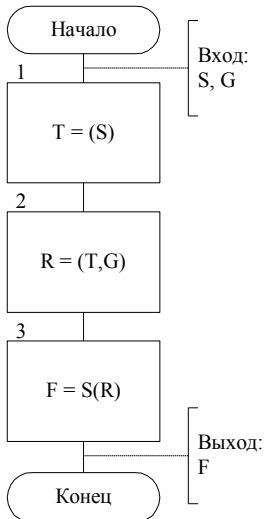


Рисунок 1. Алгоритм генерации форм слов

Решение задач искусственного интеллекта, например, задач генерации и определения форм слов можно представить в виде графа в пространстве состояний. Предлагается представить решение задач генерации и определения как поиск в ориентированном графе в пространстве словоформ.

Построение графа в пространстве словоформ основывается на следующем соответствии терминологии теории графов и метода генерации и определения форм слов: вершины графа – это словоформы, при этом корневая вершина – основа, промежуточные вершины – промежуточные словоформы, а конечные вершины – словоформы парадигмы; дуги графа – это преобразования, которые необходимо произвести над одной формой, чтобы получить другую; путь (маршрут) в графе – это цепочка преобразований; путь от корневой вершины к конечной – прямая цепочка преобразований, путь от конечной вершины к корневой – обратная цепочка преобразований; пути,

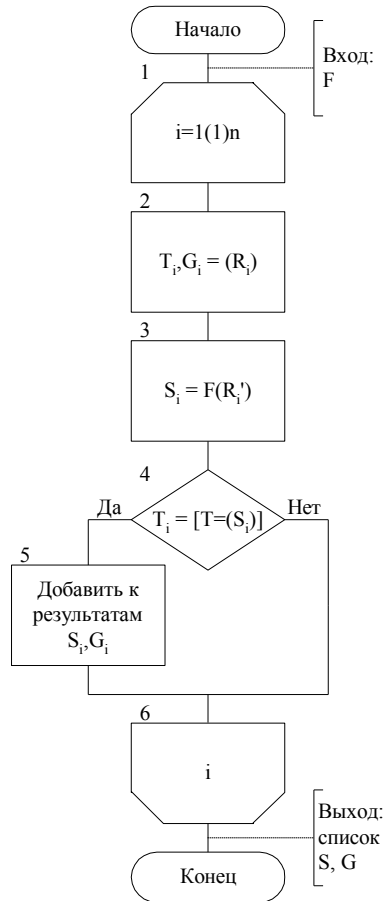


Рисунок 2. Алгоритм определения форм слов

имеющие одну и ту же корневую вершину, составляют граф, и корневая вершина становится общей для них.

В пространстве словоформ находится несколько графов, соответствующих определенным типам основ T (рисунок 3).

В зависимости от задачи генерации или определения дуги в графе ориентированы в противоположные стороны. Для задачи генерации дуги ориентированы от корневой вершины к конечным, для задачи определения – в обратном направлении.

Задача генерации форм слов заключается в выборе графа, соответствующего типу основы T , и пути в данном графе, соответствующего грамматическому значению G .

Задача определения сводится к перебору графов, соответствующих различным типам основ T , и поиску в них путей от конечных вершин к корневым. Если достигнута корневая вершина и соответствующая ей основа S присутствует в списке основ, то данная основа S и грамматическое значение G , соответствующие пути в графе, являются решением задачи определения.

Данное представление задачи генерации и определения в пространстве словоформ позволяет просто, понятно и наглядно представить метод и алгоритмы генерации и определения форм слов.

Решение задач генерации и определения форм слов можно представить с помощью теории унификации и теории множеств. Опишем объекты данной предметной области и их составляющие отношениями.

Пусть имеются следующие исходные множества:

$N_Z = \{n_1, n_2, \dots, n_{Mn}\}$ – нормальные формы слов;

$H_Z = \{h_1, h_2, \dots, h_{Mh}\}$ – одноместные функции $h(q)$, реализующие цепочки преобразований; где q – строка;

$P_Z = \{p_1, p_2, \dots, p_{Mp}\}$ – постоянные грамматические значения;

$G_Z = \{g_1, g_2, \dots, g_{Mg}\}$ – переменные грамматические значения;

$C_Z = \{c_1, c_2, \dots, c_{Mc}\}$ – семантические значения;

$T_Z = \{t_1, t_2, \dots, t_{Mt}\}$ – типы основ;

$V_Z = \{v_1, v_2, \dots, v_{Mv}\}$ – вид словоформ.

Здесь C, G, H, N, P, T, V – переменные, значениями которых могут быть элементы соответствующих множеств; c, g, h, n, p, t, v – элементы соответствующих множеств.

Цепочки преобразований

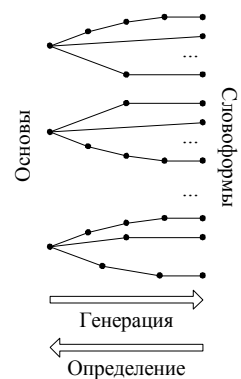


Рисунок 3. Представление задач генерации и определения в виде графов в пространстве словоформ

Определим следующие отношения исходных множеств:

$S \subset N_Z \times P_Z \times C_Z \times T_Z$ – основы;

$R \subset H_Z \times T_Z \times G_Z$ – цепочки преобразований.

Определим производные отношения:

$F \subset V_Z \times N_Z \times G_Z$ – словоформы; элементы множества можно получить из элементов отношений S и R ;

$R' \subset H'_Z \times T_Z \times G_Z$ – обратные цепочки преобразований; элементы множества можно получить из элементов отношения R заменой функции h обратной по действию функцией h' .

Элементы отношений обозначим $s(n, p, c, t)$, $r(h, t, g)$, $f(v, n, g)$ и $r'(h', t, g)$ соответственно.

Элементы отношений S и R определены, а элементы отношений F и H' могут быть получены по мере надобности.

Введем операции:

1) $\alpha = \text{Unif}(y, X)$ – унификация элемента y с элементами множества X с получением унификатора α ; если унификация успешна, то $\alpha \neq \varepsilon$, где ε – пустая подстановка;

2) $\lambda = \alpha\beta$ – композиция подстановок $\alpha = \{K_1/k_1, K_2/k_2, \dots, K_{M_a}/k_{M_a}\}$ и $\beta = \{L_1/l_1, L_2/l_2, \dots, L_{M_b}/l_{M_b}\}$, где $K_1, K_2, \dots, K_{M_a}, L_1, L_2, \dots, L_{M_b}$ – переменные; $k_1, k_2, \dots, k_{M_a}, l_1, l_2, \dots, l_{M_b}$ – константы или функции;

3) $A = B\lambda$ – применение подстановки λ к выражению B .

Опишем алгоритмы генерации и определения форм слов с помощью данных отношений и теории унификации.

В качестве исходных данных для генерации форм слов используются основа $s(n, p, c, t)$ и значение g переменной G . Необходимо получить словоформу $f(v, n, g)$.

Алгоритм генерации форм слов с помощью унификации состоит из следующих шагов.

1. $s(n, p, c, t) \Rightarrow \{T/t\}, \{N/n\}; r(H, t, g) = r(H, T, G) \{T/t\} \{G/g\}$.

2. $\lambda = \text{Unif}(r(H, t, g), R)$. Если $\lambda \neq \varepsilon$, то перейти к шагу 3, иначе перейти к шагу 4.

3. $\lambda = \{H/h\} \Rightarrow v = h(n) \Rightarrow \{V/v\}; f(v, n, g) = f(V, N, G) \{V/v\} \{N/n\} \{G/g\}$; закончить алгоритм с результатом – отношением $f(v, n, g)$.

4. Закончить алгоритм с результатом: основа $s(n, p, c, t)$ не имеет словоформ с грамматическим значением g .

Исходными данными для определения форм слов является словоформа $f(v, N, G)$. Необходимо получить список основ и словоформ $s(n, p, c, t)$ и $f(v, n, g)$.

Алгоритм определения форм слов с помощью унификации включает следующие шаги.

1. Для всех элементов $r(h_i, t_i, g_i)$ отношения R , где h_i, t_i, g_i – составляющие i -го элемента, $i = 1, 2, \dots, M_r$, выполнить шаги 2-5.

2. $r(h_i, t_i, g_i) \Rightarrow \{G/g_i\}, \{T/t_i\}; h_i \Rightarrow h'_i; n_i = h'_i(v) \Rightarrow \{N/n_i\}$.

3. $s(n_i, P, C, t_i) = s(N, P, C, T) \{N/n_i\} \{T/t_i\}$.

4. $\alpha = \text{Unif}(s(n_i, P, C, t_i), S)$.

5. Если $\alpha \neq \varepsilon$, то выполнить следующие действия:

а) $s(n_i, p, c, t_i) = s(n_i, P, C, t_i)\alpha$, где $\alpha = \{P/p; C/c\}$;

б) $f(v, n_i, g_i) = f(v, N, G) \{N/n_i\} \{G/g_i\}$ (получены на шаге 2).

в) занести элементы $s(n_i, p, c, t_i)$ и $f(v, n_i, g_i)$ в список результатов.

6. Если список результатов пуст, то закончить алгоритм с результатом – форма отсутствует в словаре, иначе закончить алгоритм с результатом – список элементов отношений $s(n, p, c, t)$ и $f(v, n, g)$.

В главе анализируется формобразование естественных языков различных семейств и групп: русского, английского, немецкого, испанского и финского и приводятся примеры цепочек преобразований для специфических видов формобразования.

Для доказательства универсальности предложенного метода генерации и определения обосновано следующее утверждение.

Утверждение 1 (о представлении формобразования в виде цепочек преобразований). Получение любой формы слова любого языка (даже не естественного) можно представить в виде цепочки преобразований.

Обоснование. Пусть в парадигме слова с основой S существует форма F с грамматическим значением G .

Получение формы F из основы S можно представить как замену подстроки S на подстроку F :

$$S \rightarrow F \quad (S \Rightarrow F)$$

или добавление формы F справа и удаления основы S слева:

$$+F; S- \quad (S \Rightarrow SF \Rightarrow F).$$

Добавление и замена подстрок является типами преобразований, а предложенные способы получения форм – цепочками. На основу S и форму F не накладывается никаких ограничений. Следовательно, получение любой формы F любого языка можно представить в виде цепочки преобразований. ■

Таким образом, теоретически доказана применимость метода генерации и определения к естественным языкам различных групп и семейств.

С одной стороны, согласно утверждению 1 цепочка преобразований описывает формобразование. С другой стороны, в работе показано, что цепочка преобразований является алгоритмом и соответствует его основным признакам, таким как элементарность шагов, детерминированность, результативность и массовость. Следовательно, формобразование – это алгоритм.

Чтобы решать задачи генерации и определения форм слов, необходимо записать получение формы F из основы S цепочкой преобразований R . Данная задача выполняется оператором-лингвистом, однако она может быть автоматизирована.

Предлагается классификация типов цепочек получения из основы S формы F в зависимости от их вида. Используются следующие обозначения: $S = \langle klmno \rangle$ – основа, $\langle ab \rangle$ – префикс, $\langle yz \rangle$ – постфикс.

Можно выделить четыре типа цепочек преобразований, некоторые из которых имеют подтипы.

1. Основа остается без изменений. Основа встречается в форме целиком, полностью. Например, «дум(ать)» → «подумал». Цепочка преобразований для получения формы $F = \langle abklmnoyz \rangle$: $R = (ab^+, +yz)$.

2. В форме присутствуют левая и правая части основы, но не вся основа целиком. Можно выделить три подтипа.

а. Между левой и правой частями подстрока в основе заменяется новой подстрокой. Например, форма глагола испанского языка «contar» – «считать, рассказывать»: «cont(ar)» → «cuento». Цепочка для получения формы $F = \langle abklqrnoyz \rangle$: $R = (m \rightarrow qr, ab^+, +yz)$.

б. Между левой и правой частями основы вставляется подстрока. Например, форма глагола испанского языка «jugar» – «играть, развлекаться»: «jug(ar)» → «juego». Цепочка для получения формы $F = \langle abklmqnoyz \rangle$: $R = (m \rightarrow mq, ab^+, +yz)$.

в. Из основы удаляется подстрока, деля ее на левую и правую части. Например, «собр(ать)» → «собрал». Цепочка для получения формы $F = \langle abklnoyz \rangle$: $R = (lmn \rightarrow ln, ab^+, +yz)$.

3. В форме присутствует только левая часть основы, а правая часть основы отсутствует. Например, форма глагола немецкого языка «bringen» – «приносить»: «bring(en)» → «brachte». Цепочка для получения формы $F = \langle abklmqryz \rangle$: $R = (-no, ab^+, +qryz)$.

4. Основа отсутствует в форме. Например, «бр(ать)» → «взял». Цепочка преобразования для получения формы $F = \langle abqryz \rangle$: $R = (klmno \rightarrow abqryz)$.

Данная классификация цепочек преобразований является полной, то есть любая цепочка может быть отнесена к одному из перечисленных типов, так как если цепочка не будет отнесена к типам 1-3, то она будет отнесена к типу 4 согласно утверждению 1.

На основе данной классификации предложен алгоритм автоматизированного построения цепочек преобразований по основе и словоформе (рисунок 4), который заключается в выделении и поиске в форме F (в том числе и аналитической) левой части L и правой части Q основы S , анализе их относительных позиций и проверки условий для определения типа цепочки.

Система автоматизированного построения цепочек преобразований (ХАРНИРИА) на основе данного алгоритма зарегистрирована в РОСПАТЕНТе.

Предложенный метод генерации и определения форм слов был использован для разработки системы генерации и определения форм слов (рисунок 5). Процедурную часть системы составляют процедуры обработки данных в словарях системы для решения задач генерации и определения и реализации типов преобразований: замены и добавления подстрок. Декларативная часть системы включает словари системы. Словарь основ слов содержит основы слов и соответствующий им тип основы, позволяющий определить часть речи. Словарь преобразований содержит преобразования и их параметры. Словарь правил об-

разования форм слов содержит цепочки преобразований для каждого типа основы.

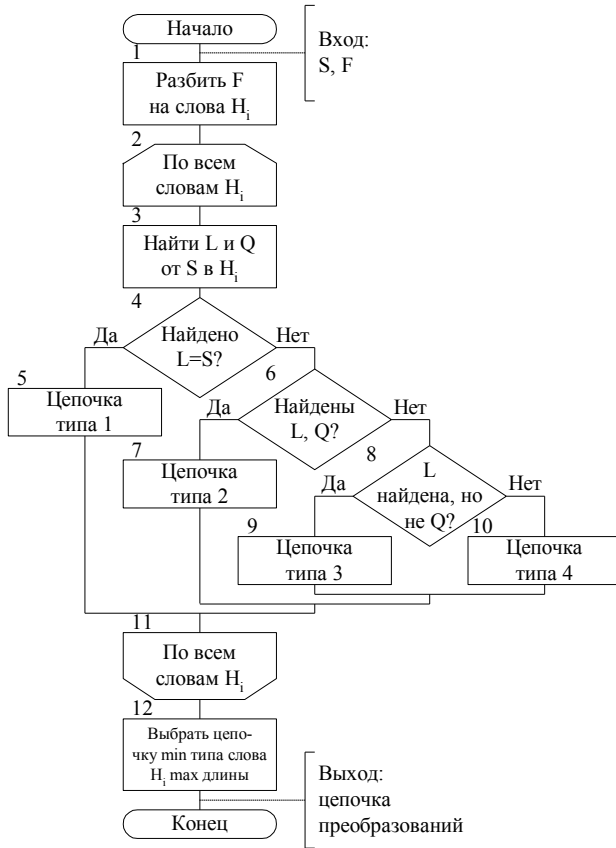


Рисунок 4. Алгоритм построения цепочек преобразований

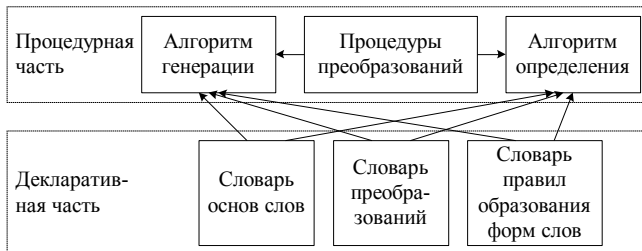


Рисунок 5. Структура системы генерации и определения форм слов

Количественные характеристики предложенной системы генерации и определения форм слов и коммерческих систем, полученные с веб-страниц разработчиков, приведены в таблице 1. Низкую скорость определения форм слов можно считать платой за универсальность предложенного метода. Система генерации и определения форм слов (ADREA) зарегистрирована в РОСПАТЕНТЕ.

Таблица 1. Количественные характеристики предлагаемой системы и аналогичных коммерческих систем

Характеристики	МЛ Морфология SDK (компания Медиалингва)	RCO Morphology Professional SDK (компания RCO)	Предлагаемая система генерации и определения форм слов
Тип и частота процессора	Intel Pentium; 0,2 ГГц	AMD Athlon; 1 ГГц	Intel Pentium IV m; 1,8 ГГц
Определение, тыс. слов/с	0,5	100	0,7
Генерация, тыс. слов/с	Нет данных	Нет данных	5

В главе 3 предложен метод обработки количественных числительных естественных языков различных семейств и групп.

Под количественными числительными будем понимать части речи, объединяющие слова, служащие для наименования чисел (в том числе и с дробной частью) или количества. Количественные числительные имеют две формы записи в тексте: 1) символьную (словесную), например, «*триста пятьдесят два*»; 2) цифровую (числа), например, «352». Количественные числительные в символьной форме далее будем называть числительными, а в цифровой форме – числами.

Как и над другими словами, над числительными производятся операции генерации, определения и их перевода на другой язык. Также существуют специальные операции над числительными. Синтез числительных заключается в преобразовании цифровой записи числительного в символьную запись. Задача анализа числительных состоит в преобразовании словесной формы записи числительных в цифровую форму записи.

Гари Хардегри из Массачусетского университета (США) предложил контекстно-свободную грамматику NG3, позволяющую описывать порядок и процедуры синтеза числительных английского языка. Можно выделить следующие направления развития грамматики Г. Хардегри для решения большего числа задач обработки числительных: анализ числительных; обработка числительных других естественных языков, а не только английского языка; описание правил образования дробной части числительного; учет склонения числительных.

Перечисленные направления развития грамматики Г. Хардегри реализуем в методе обработки числительных, основой которого является трехуровневая обобщенная модель числительного.

В модели числительного и связанных с ней алгоритмах обработки будем использовать следующие обозначения.

Обозначения в числе включают следующие классы: цифры: $Z = \{Z_0|0, Z_1|1, Z_2|2, \dots, Z_9|9\}$; знак минуса: S_- ; разделитель целой и дробной частей: J .

Обозначения в числительных разделены на следующие классы: названия цифр (простые непроединные числительные): $C = \{C_0, C_1, C_2, \dots, C_9\}$; названия десятков и сотен: $D = \{D_1, D_2\}$, где D_1 – обозначение десятков; D_2 – обозначение сотен; названия порядков триад: «тысячи», «миллионы», «миллиарды» и т. д.: $M = \{M_1, M_2, M_3, \dots\}$, где M_i – название $i + 1$ -го порядка триады числительного; название знака отрицательных чисел: P_- ; название разделителя целой и дробной частей: E ; название окончания дробной части: B .

Нормализованным числительным будем называть числительное, записанное по правилам естественного языка или модели обозначениями числительного.

Пример 4. В предложенных обозначениях число будет иметь вид

1	0	0	0	4	0	0	9	7	3
Z_1	Z_0	Z_0	Z_0	Z_4	Z_0	Z_0	Z_9	Z_7	Z_3

а числительное будет записано следующим образом

один	миллиард	четыреста	тысяч	девятьсот	семьдесят	три
C_1	M_3	C_4D_2	M_1	C_9D_2	C_7D_1	C_3

На основе проведенного анализа правил образования числительных естественных языков предлагается трехуровневая обобщенная модель числительного. Модель состоит из трех уровней.

Уровень 1. Знак числа, целая и дробная части. Разделителями частей являются слова-связки «целых», «запятая».

Уровень 2. Трехразрядные составляющие (триады). Каждая часть разделяется на триады, начиная от разделителя целой и дробной частей. Разделителями триад являются слова-связки «тысяч», «миллион» и т. д.

Уровень 3. Элементы триад. Разделителями являются слова-связки «десятки», «сотни».

Составим контекстно-свободную грамматику (КС-грамматику), описывающую порядок частей числительного, заложенный в предложенную модель числительного. Правила развиты на три уровня, соответствующих уровням модели числительного.

Уровень 1 $K = \{\emptyset|P_- \} + N_1 + \{\emptyset|(E + N_2 + \{\emptyset|B\})\}$

Уровень 2 $N_1 = C_0|N_{10}|N_{11}|N_{12}|\dots|N_{1i}$

$N_2 = (\{T_1|C_0\} + N_2)|T_1|C_0$

$N_{10} = T$

$N_{11} = T + M_1 + \{\emptyset|N_{10}\}$

$N_{12} = T + M_2 + \{\emptyset|N_{10}|N_{11}\}$

...

$N_{1i} = T + M_i + \{\emptyset|N_{11}|N_{12}|\dots|N_{1(i-1)}\}$

Уровень 3

$T = T_1|T_2|T_3$

$T_1 = C_1|C_2|C_3|C_4|C_5|C_6|C_7|C_8|C_9$

$T_2 = T_1 + D_1 + \{\emptyset|T_1\}$

$T_3 = T_1 + D_2 + \{\emptyset|T_1|T_2\}$

Первый уровень модели числительного описывает общий порядок построения числительного K : знак числа P_+ , целую N_1 и дробную N_2 части.

Второй уровень модели числительного включает описание целой и дробной частей числительного. Целая часть числительного N_1 состоит из числа ноль C_0 или последовательности $i + 1$ триад N_{1i} . В числительном обязательно присутствует первая слева триада, остальные триады могут отсутствовать. Триады T разделены названиями порядка триад M . Дробная часть числительного N_2 состоит из последовательности названий цифр.

Третий уровень модели числительного описывает порядок сотен, десятков и единиц в триаде. Существуют три вида триад T : триады, в которых обязательно присутствуют единицы T_1 , десятки T_2 , сотни T_3 .

Пример 5. Запишем с помощью грамматики предложенной модели числительное, соответствующее числу 34 567,89 (рисунок 6). Обойдя конечные вершины (выделены на рисунке) слева направо, получим нормализованное числительное $C_3D_1C_4M_1C_5D_2C_6D_1C_7EC_8C_9$. □

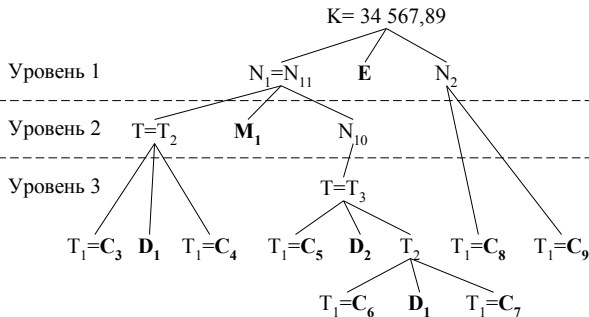


Рисунок 6. Пример записи числа 34 567,89 с помощью трехуровневой модели числительного

Предложенная модель числительного является промежуточным этапом в следующих операциях над числительными (рисунок 7): синтезе и анализе числительных; переводе числительных с одного языка на другие языки с заданным грамматическим значением; генерации и определении числительных с заданным грамматическим значением; выявлении ошибок в числительных.

В работе приводится порядок действий для выполнения данных операций. В настоящее время реализовано 30 направлений преобразований числительных (5 языков: русский, английский, немецкий, испанский, финский и число). Число языков может быть увеличено, так как данный подход является универсальным.

В работе показано, что перевод числительных через модель имеет меньшую трудоемкость, чем через число в качестве промежуточного этапа. Это вызвано различием структуры числительного и числа, тогда как структуры числительного и числительного модели одинаковы.

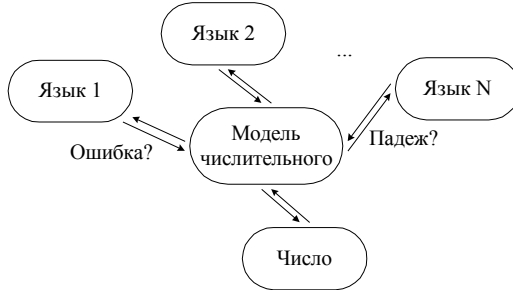


Рисунок 7. Взаимодействие модели числительного, языков и чисел

Стрелки на рисунке 7 соответствуют алгоритмам обработки числительных, реализованным нормальными алгоритмами Маркова. Например, алгоритм анализа целой части числительных имеет вид:

- 1) $M_j \gamma_i^L \rightarrow \gamma_1^j [Z_0]_{(4-i)+3 \times (j-L-1)}$; $i = 1, 2, 3; L = 0, 1, 2, \dots$;
; $j = L, L+1, L+2, \dots$
- 2) $C_k \gamma_1^j \rightarrow \gamma_2^j Z_k$; $j = 0, 1, 2, \dots; k = 0, 1, \dots, 9$
- 3) $C_k D_1 \gamma_i^j \rightarrow \gamma_3^j Z_k [Z_0]_{2-i}$; $i = 1, 2; j = 0, 1, 2, \dots; k = 1, 2, \dots, 9$
- 4) $C_k D_2 \gamma_i^j \rightarrow \gamma_1^{j+1} Z_k [Z_0]_{3-i}$; $i = 1, 2, 3; j = 0, 1, 2, \dots; k = 1, 2, \dots, 9$
- 5) $C_k \gamma_i^j \rightarrow C_k \gamma_i^j \cdot$; $i = 2, 3; j = 0, 1, 2, \dots; k = 0, 1, \dots, 9$
- 6) $C_k D_1 \gamma_3^j \rightarrow C_k D_1 \gamma_3^j \cdot$; $j = 0, 1, 2, \dots; k = 1, 2, \dots, 9$
- 7) $P_- \gamma_i^j \rightarrow S_- \cdot$; $i = 1, 2, 3; j = 0, 1, 2, \dots$
- 8) $\gamma_i^j \rightarrow \emptyset \cdot$; $i = 1, 2, 3; j = 0, 1, 2, \dots$
- 9) $E \rightarrow \gamma_1^0 J$
- 10) $\rightarrow + \gamma_1^0$

Здесь операция $\rightarrow +\gamma$ означает добавление символа γ к строке справа.

В главе 4 предложена модификация нормальных алгоритмов Маркова, позволяющая уменьшить трудоемкость решения задач и количество подстановок в схеме алгоритма.

В теории нормальных алгоритмов Маркова существуют две классические задачи. Задача обращения заключается в записи символов, составляющих строку, в обратном порядке (рисунок 8). Задача удвоения состоит в копировании символов строки слева или справа (рисунок 9).

Несмотря на линейность решения данных задач, нормальные алгоритмы, предложенные А.А. Марковым и другими исследователями, имеют квадратичную трудоемкость. Кроме того, Г.С. Цейтин доказал, что число шагов нормального алгоритма обращения строки из не ме-

нее чем двухбуквенного алфавита должно быть не меньше, чем cm^2 , где c – константа.

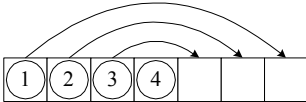


Рисунок 8. Задача обращения

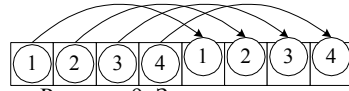


Рисунок 9. Задача удвоения

В алгоритмах существуют три типа вычислительных процессов: 1) линейный; 2) разветвляющийся; 3) циклический. Но в нормальных алгоритмах Маркова возможна реализация только разветвляющегося и циклического вычислительных процессов. Предлагается метод устранения ограничения нормальных алгоритмов Маркова, который заключается в модификации этих алгоритмов таким образом, чтобы они могли реализовывать линейный вычислительный процесс.

Данный метод реализуем в предлагаемой модификации нормальных алгоритмов Маркова. Модификация позволяет реализовать линейный вычислительный процесс и, таким образом, сократить количество шагов выполнения алгоритма и подстановок схемы. Данную модификацию назовем линейными нормальными алгоритмами, общая схема которых имеет вид:

$$P_1: P_1^{(1)} \rightarrow Q_1^{(1)}; P_2^{(1)} \rightarrow Q_2^{(1)}; \dots; P_{k_1}^{(1)} \rightarrow Q_{k_1}^{(1)} [\bullet]$$

$$P_2: P_1^{(2)} \rightarrow Q_1^{(2)}; P_2^{(2)} \rightarrow Q_2^{(2)}; \dots; P_{k_2}^{(2)} \rightarrow Q_{k_2}^{(2)} [\bullet]$$

...

$$P_d: P_1^{(d)} \rightarrow Q_1^{(d)}; P_2^{(d)} \rightarrow Q_2^{(d)}; \dots; P_{k_d}^{(d)} \rightarrow Q_{k_d}^{(d)} [\bullet]$$

В схеме обозначения P только с нижним индексом и P и Q с верхним и нижним индексами являются подстроками; $k_1, k_2, \dots, k_d = 1, 2, \dots$.

Каждая строка схемы содержит метку P с нижним индексом и соответствующие данной метке одну или несколько подстановок. Метка строки может не совпадать с левой частью первой подстановки.

Процесс применения алгоритма к строке S заключается в следующем. Схема просматривается сверху вниз, начиная с первой строки. Если метка является подстрокой строки S , то происходит применение подстановок, соответствующих данной метке. Подстановки в одной строке схемы выполняются последовательно, однократно, одна за другой. После этого схема просматривается вновь, начиная с первой строки. Как и в нормальных алгоритмах Маркова, подстановка заключается в замене в строке S первого вхождения слева подстроки из левой части подстановки на подстроку из правой части подстановки.

Алгоритм заканчивает выполнение в трех случаях: 1) ни одна метка P не встречается в строке S ; 2) левая часть подстановки L , которая соответствует метке P , не встречается в строке S ; 3) выполнена заключительная подстановка, то есть подстановка, заканчивающаяся символом «•».

Линейный нормальный алгоритм обращения строки символов алфавита $Z = \{Z_1, Z_2, \dots, Z_n\}$ включает следующие подстановки.

- 1) $\lambda Z_i: \lambda Z_i \rightarrow \lambda; \rightarrow Z_i+$; $i = 1, 2, \dots, n$
- 2) $\lambda: \lambda \rightarrow \emptyset \bullet$
- 3) $\emptyset: \rightarrow \lambda+$

Здесь операция $\rightarrow \lambda+$ означает добавление символа γ к строке слева, а операция $\lambda \rightarrow \emptyset$ – удаление символа λ из строки.

Принцип работы данного алгоритма следующий. Указатель λ не входит в алфавит Z , поэтому первой будет выполнена подстановка 3, так как в подстановках 1-2 указатель λ содержится в метках. Подстановка 3 добавляет указатель λ к строке слева. Обращение строки заключается в удалении символов строки перед указателем λ и добавлении их к строке слева (подстановка 1). Когда обработаны все символы строки (не осталось символов перед указателем λ), указатель λ удаляется из строки и алгоритм заканчивает выполнение (подстановка 2).

Линейный нормальный алгоритм удвоения строки символов алфавита $Z = \{a, b\}$ имеет следующую схему:

- 1) $\alpha a: \alpha a \rightarrow \alpha \alpha; \beta \rightarrow \alpha \beta$
- 2) $\alpha b: \alpha b \rightarrow \beta \alpha; \beta \rightarrow \beta \beta$
- 3) $\alpha: \alpha \rightarrow \emptyset; \beta \rightarrow \emptyset \bullet$
- 4) $\emptyset: \rightarrow \beta \alpha+$

Поясним работу алгоритма. Строка S не содержит вспомогательных символов α и β , поэтому первой выполняется подстановка 4. В результате строка имеет вид $\beta \alpha S$. Далее выполняются подстановки 1 или 2 в зависимости от символа, который находится справа от указателя α . Работу этих подстановок можно записать так: $S' \beta S' \alpha \chi \gamma S'' \Rightarrow S' \chi \beta S' \chi \alpha \gamma S''$. Здесь S' и S'' – подстроки строки S , символы χ и γ – символы a или b . До и после выполнения подстановки слева от указателей α и β находятся одинаковые подстроки, что доказывает удвоение символов. Когда справа от указателя α не остается символов, то складывается ситуация $S \beta S \alpha$. В этом случае выполняется подстановка 3, удаляющая указатели α и β . Результатом работы алгоритма является строка SS .

Предложенные алгоритмы обращения и удвоения имеют линейную трудоемкость решения данных задач и позволяют сократить количество подстановок в схеме (таблица 2), что подтверждает правильность предложенного метода устранения ограничения.

Количество подстановок алгоритмов удвоения приведено для алфавита $Z = \{a, b\}$.

В работе показано, что схему любого нормального алгоритма Маркова можно привести к схеме линейного нормального алгоритма, а также предложен линейный нормальный алгоритм реализации такого сведения. Это влечет два следствия: 1) любую задачу, которую можно решить с помощью нормального алгоритма Маркова, можно решить и с помощью линейного нормального алгоритма; 2) трудоемкость любо-

го линейного нормального алгоритма будет не хуже трудоемкости нормального алгоритма Маркова, но не наоборот.

Таблица 2. Трудоемкости алгоритмов решения задач обращения и удвоения

	Трудоемкость (m – длина строки)	Количество подстановок (n – размер алфавита Z)	Тип алго- ритма
Алгоритм обращения А.А. Маркова и Н.М. Нагорного	$m^2/2 + 5m/2 + 3$	$n^2 + n + 4$	Квадратичный
Обращающий самопополняемый слева алгоритм И.А. Цветкова	$m^2/2 + 3m/2 + 2$		Квадратичный
Предлагаемый алгоритм обращения	$2m + 2$	$2n + 2$	Линейный
Алгоритм удвоения А.А. Маркова и Н.М. Нагорного	$m^2/2 + 5m/2 + 2$	10 (при $n=2$)	Квадратичный
Алгоритм удвоения В.А. Мошенского	$m^2/2 + 3m/2 + 2$	9 (при $n=2$)	Квадратичный
Предлагаемый алгоритм удвоения	$2m + 3$	7 (при $n=2$)	Линейный

В главе 5 предложены информационные системы на основе разработанных методов обработки форм слов и числительных: система проверки знаний (СПЗ) морфологии естественных языков (пример формулировки задания: «Записать форму слова «*rahayksikkö*» с заданным грамматическим значением»); СПЗ правил образования количественных числительных («Перевести числительное «*zweihundertachtundsechzig*» на испанский язык»); доступна по адресу <http://prutzkow.com/numbers/test.htm>); СПЗ функционирования алгоритмических моделей («Записать результат работы линейного нормального алгоритма с начальным словом «*aab*»); система сравнения статей коллективных договоров.

Рассмотрим принцип работы СПЗ на примере СПЗ по морфологии (рисунок 10). СПЗ отправляет запрос (1) системе генерации и определения форм слов. В ответ на запрос СПЗ получает задание обучаемому и ответ к нему (2). СПЗ выдает полученное задание обучаемому (3), получает ответ обучаемого (4) и сравнивает его с правильным ответом. Если ответ правильный, то СПЗ сообщает об этом обучаемому (7). Если ответ неверный (например, если обучаемый неправильно указал искомую словоформу (см. формулировку задания)), то система передает ответ системе генерации и определения форм слов (5), которая анализирует ответ и выдает его грамматическое значение (6). СПЗ выдает результат анализа с пояснениями обучаемому (7).

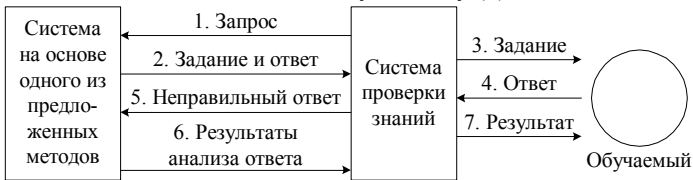


Рисунок 10. Схема проверки знаний

Другие СПЗ работают по этому же принципу. Шаги 5 и 6 не реализованы в СПЗ функционирования алгоритмических моделей.

Особенностью данных СПЗ является динамическая генерация заданий, что позволяет получить большое число отличных друг от друга вариантов, и проверка неправильных ответов обучаемого, что позволяет пояснять обучаемым сделанные ими ошибки.

Предложенные СПЗ зарегистрированы в РОСПАТЕНТе.

СПЗ правил образования количественных числительных является частью Интернет-приложения для обработки количественных числительных, доступного по адресу <http://prutzkow.com/numbers/> в сети Интернет. Ежемесячно ресурс посещают более 1 000 человек из более, чем 80 стран (рисунок 11). Интернет-приложение используется в более 200 университетах США, России, Канады и других стран.



Рисунок 11. География посетителей Интернет-приложения

Наибольшее число пользователей Интернет-приложения проживают в США, России, Европейском Союзе (ЕС) и на Украине (таблица 3).

Таблица 3. Распределение долей стран, в которых проживают пользователи Интернет-приложения

Страны	Доли от общего числа пользователей (в III квартале 2014 г.)
Россия	27,8%
Украина	4,9%
ЕС	4,8%
США	44,7%
Остальные 89 стран	17,8%

Направления перевода количественных числительных и их доли представлены в таблице 4.

Разработанные методы используются в информационной системе анализа коллективных договоров (КД). При принятии нового КД, включающего 100-150 статей, необходимо провести его экспертизу.

Для этого эксперт должен выставить оценки каждой статье и на их основе определить суммарную оценку КД. Экспертизу можно упростить, если найти аналоги статей из предыдущей редакции КД и их оценки. Для решения данной задачи предлагается система сравнения статей КД.

Статья нового КД может быть сформирована следующими способами: 1) статья перенесена из предыдущей редакции КД без изменений; 2) статья получена из аналога путем удаления, замены или добавления одного или нескольких слов; 3) статья добавлена в КД заново или ее формулировка полностью переработана.

Таблица 4. Направления перевода количественных числительных

Направления перевода	Доли
Число – испанский язык	42,1%
Число – русский язык	23,1%
Число – немецкий язык	22,2%
Число – английский язык	8,6%
Остальные направления	4,0%

Поиск аналога заключается в нахождении статьи предыдущей редакции с максимальным значением коэффициента сходства

$$K = c_1 \cdot F/N + c_2 \cdot S/N + c_3 \cdot 1/P \cdot W/N,$$

где c_1 , c_2 , c_3 – весовые коэффициенты; N – количество слов в анализируемой статье; W – количество слов в последовательностях; F – количество одинаковых словоформ в сравниваемых статьях; S – количество одинаковых слов (лексем) в сравниваемых статьях («уволенный» – «увольняемый»); также составлен словарь синонимов («вуз» – «университет»); P – количество общих последовательностей словоформ; например, в первой статье «*(расходование финансовых средств подразделениями) осуществляется (с учетом мнения профсоюзного комитета)*» и во второй статье «*выделение и (расходование финансовых средств подразделениями) происходит (с учетом мнения профсоюзного) органа подразделения*» присутствуют две общие последовательности, выделенные скобками.

В зависимости от значения коэффициента оценки переносятся на новую статью автоматически или статья выдается для оценки эксперту.

Для определения количества одинаковых слов (лексем) L используется предложенный метод генерации и определения форм слов.

В текстах статей могут встречаться как числительные, так и числа («*продолжительностью двадцать дней*» или «*продолжительностью 20 дней*»). Поэтому текст статьи нормализуется, и все числительные преобразуются в числа. Для нормализации текста используется метод обработки числительных.

Автоматизация сравнения статей позволила упростить работу экспертов по анализу статей КД, а, значит, сократить время и повысить качество их работы.

Также в работе приведены описание и структура перспективных информационных систем, раскрывающих весь потенциал предложенных методов обработки форм слов и числительных.

В **заключении** сформулированы основные научные результаты, полученные в рамках решения поставленной крупной научно-технической проблемы.

В **приложениях** приводятся свидетельства о государственной регистрации программ для ЭВМ, акты внедрения результатов диссертационной работы и экранные формы информационных систем, описанных в главе 5.

ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ РАБОТЫ

В диссертационной работе решена крупная научно-техническая проблема разработки универсальных методов и программных средств обработки форм слов и количественных числительных естественных языков различных групп и семейств, имеющая важное теоретическое и прикладное значение. Данная обработка необходима для построения естественно-языковых интерфейсов, а также для решения большинства задач АОТ, таких как машинный перевод, выявление знаний в тексте, поиск текста по запросу в информационных сетях. В работе получены следующие результаты.

1. Проведен анализ задач и систем АОТ. Сделан вывод, что морфологический анализ и синтез и обработка количественных числительных являются важными этапами решения задач и частями данных систем.

2. Проанализированы достоинства и недостатки различных подходов и методов морфологического анализа и синтеза. Сформулированы требования к разрабатываемому методу генерации и определения форм слов.

3. Разработана модель формообразования, позволяющая обрабатывать полную парадигму слова и являющаяся универсальной для различных естественных языков. На основе модели разработаны алгоритмы генерации и определения, составляющие вместе с моделью метод генерации и определения форм слов естественных языков. Показана применимость метода к русскому, английскому, немецкому, испанскому и финскому языкам, относящимся к различным группам и семействам. Для автоматизированного построения цепочек преобразований разработан алгоритм на основе классификации образования форм слов.

4. Разработана трехуровневая обобщенная модель числительного как развитие грамматики Г. Хардегри, которая позволяет уменьшить трудоемкость решения этих задач. Предложен метод обработки количественных числительных естественных языков, использующий модель числительного в качестве промежуточного этапа и позволяющий переводить числительные различных языков в любом направлении. Используемая структура взаимодействия модели и числительных языков позволяет легко увеличивать количество поддерживаемых языков.

Алгоритмы обработки числительных записаны с помощью нормальных алгоритмов Маркова.

5. Предложен метод устранения ограничения нормальных алгоритмов Маркова, который заключается в модификации их таким образом, чтобы они могли реализовывать линейный вычислительный процесс. Метод реализован в предложенной модификации нормальных алгоритмов Маркова, названной линейными нормальными алгоритмами. Модификация позволяет уменьшить трудоемкость решения задач по сравнению с ними. Разработаны линейные нормальные алгоритмы решения задач обращения и удвоения, имеющие линейную трудоемкость, тогда как нормальные алгоритмы Маркова решают эти задачи с квадратичной трудоемкостью.

6. Разработаны системы проверки знаний морфологии, правил образования количественных числительных и функционирования алгоритмических моделей, в основе которых лежат предложенные методы и модификация. Данные системы позволяют динамически генерировать задания и анализировать неправильные ответы обучаемых. Разработано Интернет-приложение на основе метода обработки количественных числительных, позволяющее переводить числительные русского, английского, немецкого, испанского и финского языков, ежемесячное число пользователей из различных стран мира которого превышает 1 000, в том числе из ведущих университетов США, России, Канады и других стран. Разработана система сравнения статей коллективных договоров, в которых используются предложенные методы, позволившая сократить время и повысить качество анализа коллективных договоров. Система разработана в рамках нескольких НИР и внедрена в профсоюзных организациях. Разработанные системы зарегистрированы в РОСПАТЕНТе.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в изданиях из перечня ВАК

1. Пруцков, А.В. Информационно-справочный ресурс по словообразованию естественных языков [Текст] / А.В. Пруцков // Информационные ресурсы России. – 2004. – № 6. – С. 22-24.
2. Пруцков, А.В. Морфологический анализ и синтез текстов посредством преобразований форм слов [Текст] / А.В. Пруцков // Вестник Рязанской государственной радиотехнической академии. – 2004. – № 15. – С. 70-75.
3. Пруцков, А.В. Применение информационных ресурсов для автоматизации обучения и проверки знаний [Текст] / А.В. Пруцков // Информационные ресурсы России. – 2005. – № 1. – С. 18-20.
4. Пруцков, А.В. Автоматизация обучения словообразованию иностранных языков [Текст] / А.В. Пруцков // Информатика и образование. – 2005. – № 5. – С. 117-119.
5. Пруцков, А.В. Построение систем дистанционного обучения на основе информационных ресурсов [Текст] / С.В. Алимпиева, А.В. Пруцков // Информационные ресурсы России. – 2005. – № 4. – С. 16-18.
6. Пруцков, А.В. Статический и динамический подходы к проектированию подсистем проверки знаний автоматизированных обучающих систем [Текст] / А.В. Пруцков // Информационные ресурсы России. – 2006. – № 1. – С. 27-29.

7. Пруцков, А.В. Методы поиска решений в лингвистических автоматизированных обучающих системах [Текст] / А.В. Пруцков // Научно-техническая информация. Сер. 2. Информационные процессы и системы. – 2006. – № 4. – С. 15-18.
8. Пруцков, А.В. Определение и генерация сложных форм слов естественных языков при морфологическом анализе и синтезе [Текст] / А.В. Пруцков // Известия Таганрогского государственного радиотехнического университета. – 2006. – № 15 (70). – С. 10-14.
9. Пруцков, А.В. Автоматизация обучения морфологии естественных языков с помощью электронных информационных ресурсов [Текст] / А.В. Пруцков // Научно-техническая информация. Сер. 2. Информационные процессы и системы. – 2007. – № 3. – С. 41-44.
10. Пруцков, А.В. Генерация и определения форм слов естественных языков на основе их последовательных преобразований [Текст] / А.В. Пруцков // Вестник Рязанского государственного радиотехнического университета. – 2009. – № 27. – С. 51-58.
11. Пруцков, А.В. Обработка числительных естественных языков с помощью формальных грамматик и нормальных алгоритмов Маркова [Текст] / А.В. Пруцков // Вестник Рязанского государственного радиотехнического университета. – 2009. – № 28. – С. 49-55.
12. Пруцков, А.В. Линейные нормальные алгоритмы [Текст] / А.В. Пруцков // Вестник Рязанского государственного радиотехнического университета. – 2010. – № 33. – С. 39-45.
13. Пруцков, А.В. Информационная система с использованием поиска решений задач генерации и определения в пространстве словоформ [Текст] / А.В. Пруцков, А.Н. Пылькин // Вестник Рязанского государственного радиотехнического университета. – 2011. – № 36. – С. 39-43.
14. Пруцков, А.В. Алгоритмическое обеспечение универсального метода генерации и определения форм слов [Текст] / А.В. Пруцков // Научно-техническая информация. Сер. 2. Информационные процессы и системы. – 2011. – № 9. – С. 19-25.
Перевод статьи на английский язык: Prutskov A.V. Algorithmic Provision of a Universal Method for Word-Form Generation and Recognition // Automatic Documentation and Mathematical Linguistics, 2011, Vol. 45, No. 5, pp. 232-238.
15. Пруцков А.В. Программное обеспечение методов обработки форм слов и числительных [Текст] / А.В. Пруцков, А.К. Розанов // Вестник Рязанского государственного радиотехнического университета. – 2011. – № 38. – С. 78-82.
16. Пруцков А.В. Решение задач обращения и удвоения с помощью линейных нормальных алгоритмов [Текст] / А.В. Пруцков // Известия Южного Федерального университета. Технические науки. – 2012. – № 1 (126). – С. 139-147.
17. Пруцков А.В. Интернет-приложение метода обработки количественных числительных естественных языков [Текст] / А.В. Пруцков, Д.М. Цыбулько // Вестник Рязанского государственного радиотехнического университета. – 2012. – № 41. – С. 70-74.
18. Пруцков А.В. Проблемно-ориентированное объектное программирование [Текст] / А.В. Пруцков, Д.М. Цыбулько // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 45. – С. 57-62.
19. Пруцков А.В. Анализ статистики использования Интернет-приложения обработки количественных числительных естественных языков [Текст] / А.В. Пруцков, Д.М. Цыбулько // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 46. – С. 130-134.
20. Пруцков А.В. Применение проблемно-ориентированного объектного программирования для описания порядка работы интеллектуальных и информационных систем [Текст] / А.В. Пруцков, Д.М. Цыбулько // Вестник Рязанского государственного радиотехнического университета. – 2014. – № 1 (47). – С. 92-96.
21. Пруцков А.В. Применение теории унификации в морфологическом анализе и синтезе форм слов естественных языков [Текст] / И.Ю. Каширин, А.В. Пруцков // Информатизация образования и науки. – 2014. – № 4 (24). – С. 85-91.
22. Пруцков А.В. Методы морфологической обработки текстов [Текст] / А.В. Пруцков, А.К. Розанов // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 3 (27). – С. 119-133.

23. Пруцков А.В. Теоретико-множественное представление метода обработки количественных числительных естественных языков и особенности их перевода в различных странах [Текст] / А.В. Пруцков, Д.М. Цыбулько // Вестник Рязанского государственного радиотехнического университета. – 2014. – № 4 (50). – Вып. 2. – С. 69-75.

Свидетельства о государственной регистрации программ для ЭВМ

24. Свидетельство о государственной регистрации программы для ЭВМ № 2010615326, Россия. Система генерации и определения форм слов естественных языков (ADREA) [Текст] / А.В. Пруцков, Г.В. Овечкин. Зарегистрировано в РОСПАТЕНТ 18.08.2010, заявка № 2010613607.

25. Свидетельство о государственной регистрации программы для ЭВМ № 2011611621, Российская Федерация. Информационная система проверки знаний по формообразованию естественных языков (SALVINIA) [Текст] / А.В. Пруцков, А.К. Розанов. Зарегистрировано в РОСПАТЕНТ 17.02.2011, заявка № 2010618136.

26. Свидетельство о государственной регистрации программы для ЭВМ № 2011612830, Российская Федерация. Система автоматизированного построения цепочек преобразований (XAPHIRIA) [Текст] / А.В. Пруцков. Зарегистрировано в РОСПАТЕНТ 08.04.2011, заявка № 2011610974.

27. Свидетельство о государственной регистрации программы для ЭВМ № 2011615475, Российская Федерация. Информационная система проверки знаний по правилам образования количественных числительных (BRETTA) [Текст] / А.В. Пруцков, А.А. Суворов. Зарегистрировано в РОСПАТЕНТ 13.07.2011, заявка № 2011613616.

28. Свидетельство о государственной регистрации программы для ЭВМ № 2011615476, Российская Федерация. Информационная система проверки знаний по функционированию алгоритмических моделей (KAGURI) [Текст] / А.В. Пруцков, А.О. Никифоров. Зарегистрировано в РОСПАТЕНТ 13.07.2011, заявка № 2011613617.

29. Свидетельство о государственной регистрации программы для ЭВМ № 2012661379, Российская Федерация. Программное обеспечение Интернет-ресурса обработки количественных числительных естественных языков (CLEDONIA) [Текст] / А.В. Пруцков, Д.М. Цыбулько. Зарегистрировано в РОСПАТЕНТ 13.12.2012, заявка № 2012618855.

30. Свидетельство о государственной регистрации программы для ЭВМ № 2013615630, Российская Федерация. Программная система анализа и оценки статей коллективных договоров (ASTRELIA) [Текст] / А.В. Пруцков, А.С. Шустов. Зарегистрировано в РОСПАТЕНТ 17.06.2013, заявка № 2013613273.

Статьи в сборниках научных трудов

31. Пруцков, А.В. Способ устранения недостатков существующих методов генерации и определения форм слов [Текст] / А.В. Пруцков // Математическое и программное обеспечение вычислительных систем: межвуз. сб. науч. тр. / под ред. Л.П. Коричнева. – М.: Минобрнауки России; Рязань: Рязан. гос. радиотехн. акад., 2004. – С. 107-110.

32. Пруцков, А.В. Модульный подход к построению дистанционных систем проверки знаний [Текст] / А.В. Пруцков // Информационные технологии в процессе подготовки современного специалиста: межвуз. сб. – Липецк: ГОУ ВПО «Липецкий гос. пед. ун-т», 2005. – С. 131-136.

33. Пруцков, А.В. Оптимизация времени преподавания учебного курса [Текст] / К.И. Поляков, А.В. Пруцков // Системы и методы обработки и анализа информации: сб. науч. статей / под ред. С.С. Садыкова, Д.Е. Андрианова – М.: Горячая линия-Телеком, 2005. – С. 138-149.

34. Пруцков, А.В. Грамматика для формального представления количественных числительных естественных языков [Текст] / А.В. Пруцков // Математическое и программное обеспечение информационных систем: межвуз. сб. науч. тр. / под ред. А.Н. Пылькина. – М.: Горячая линия-Телеком, 2007 – С. 26-29.

35. Пруцков, А.В. Методы представления и хранения морфологических данных [Текст] / А.В. Пруцков // Информационные технологии в процессе подготовки современного специалиста: межвуз. сб. статей. – Липецк: ГОУ ВПО «Липецкий гос. пед. ун-т», 2007. – С. 205-212.

36. Пруцков, А.В. Принципы морфологической обработки текста [Текст] / А.В. Пруцков // Математическое и программное обеспечение информационных систем: межвуз. сб. науч. тр. / под ред. А.Н. Пылькина. – М.: Горячая линия-Телеком, 2008. – С. 33-36.

37. Пруцков, А.В. Генерация и определение форм слов через представление формобразования как последовательности преобразований [Текст] / А.В. Пруцков // Информационные технологии в процессе подготовки современного специалиста: межвуз. сб. статей. Вып. 11. – Липецк: ГОУ ВПО «Липецкий гос. пед. ун-т», 2008. – С. 191-200.

38. Пруцков, А.В. Автоматизированное построение цепочек преобразований по основе и результирующей словоформе [Текст] / А.В. Пруцков // Системы и методы обработки и анализа данных: сб. статей молодых исследователей; Вып. 1 / под ред. С.С. Садыкова, Д.Е. Андрианова; Владим. гос. ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2009. – С. 46-52.

39. Пруцков, А.В. Алгоритмическое обеспечение метода генерации и определения форм слов [Текст] / А.В. Пруцков // Задачи системного анализа, управления и обработки информации: межвуз. сб. науч. тр. Вып. 3 / под общ. ред. Е.В. Никульчева. – М.: Моск. гос. ун-т печати, 2010. – С. 130-137.

40. Пруцков, А.В. Линейная модификация нормальных алгоритмов Маркова [Текст] / А.В. Пруцков // Информационные технологии в процессе подготовки современного специалиста: межвуз. сб. статей. Вып. 13. – Липецк: ГОУ ВПО «Липецкий гос. пед. ун-т», 2010. – С. 166-174.

41. Пруцков, А.В. Информационные технологии в автоматизированном обучении морфологии естественных языков [Текст] / А.В. Пруцков // Традиции и инновации в лингвистике и лингвообразовании: сб. статей / отв. ред. К.А. Власова; АГПИ им. А.П. Гайдара. – Арзамас: АГПИ, 2011. – С. 121-124.

42. Пруцков А.В. Использование трехуровневой обобщенной модели числительного для машинного перевода количественных числительных [Текст] / А.В. Пруцков // Информационные технологии в процессе подготовки современного специалиста: межвуз. сб. статей. Вып. 14. – Липецк: ФГБОУ ВПО «Липецкий гос. пед. ун-т», 2011. – С. 167-174.

43. Пруцков А.В. Алгебраическое представление модели формобразования естественных языков [Текст] / А.В. Пруцков // Cloud Of Science. – 2014. – Т. 1. – № 1. – С. 88-97.

44. Пруцков А.В. Трехуровневая обобщенная модель числительного и ее прикладное приложение [Текст] / А.В. Пруцков // Задачи системного анализа, управления и обработки информации: межвуз. сб. науч. тр. / под общ. ред. Е.В. Никульчева. – М.: МТИ, 2015. – С. 121-128.

Тезисы докладов конференций

45. Пруцков, А.В. Принципы построения системы генерации и распознавания форм глаголов испанского языка [Текст] / А.В. Пруцков // Проблемы передачи и обработки информации в сетях и системах телекоммуникаций: материалы 10-й Междунар. науч.-техн. конф. / Рязан. гос. радиотехн. акад. – Рязань, 2001. – С. 204-205.

46. Пруцков, А.В. Система обработки числительных в автоматизированной обучающей системе иностранным языкам [Текст] / А.В. Пруцков, П.В. Овечкин // Электронные средства и системы управления: материалы Междунар. науч.-практ. конф. / Томский гос. ун-т систем управления и радиоэлектроники. – Томск: Изд-во Института оптики атмосферы СО РАН, 2004. – Ч. 2. – С. 198-199.

47. Пруцков, А.В. Применение эффективного метода генерации и определения форм слов в АОС иностранным языкам [Текст] / А.В. Пруцков // Техническая кибернетика, радиоэлектроника и системы управления: тез. докл. 7-й Всерос. науч. конф. студентов и аспирантов / Таганрогский гос. радиотехн. ун-т. – Таганрог, 2004. – С. 249.

48. Пруцков А.В. Система преобразования числительных и чисел в системах дистанционного обучения [Текст] / А.В. Пруцков // Сети и системы связи: материалы Всероссийского науч.-практ. семинара / Рязан. высш. военное командное училище связи им. Маршала Советского Союза М.В. Захарова. – Рязань, 2005. – С. 322-323.

49. Пруцков, А.В. Применение систем генерации и определения форм слов и количественных числительных в процессе понимания и синтеза речи [Текст] /

А.В. Пруцков // Новые информационные технологии в научных исследованиях и в образовании: материалы 12-й Всерос. науч.-техн. конф. студентов, молодых ученых и специалистов / Рязан. гос. радиотехн. ун-т. – Рязань, 2007. – С. 39-40.

50. Пруцков, А.В. Представление числительных с помощью категориальной грамматики [Текст] / А.В. Пруцков // Сети, системы связи и телекоммуникации. Деятельность ВУЗа при переходе на Федеральный государственный образовательный стандарт 3-го поколения: материалы 33-й Всерос. науч.-техн. конф. В 2 ч. / Рязан. высш. военное командное училище связи. – Рязань, 2008. – Ч. 1. – С. 38-39.

51. Пруцков, А.В. Расширение возможностей системы генерации и определения форм слов [Текст] / А.В. Пруцков // Новые информационные технологии в научных исследованиях и в образовании: материалы 13-й Всерос. науч.-техн. конф. студентов, молодых ученых и специалистов / Рязан. гос. радиотехн. ун-т. – Рязань, 2008. – Ч. 1. – С. 7-8.

52. Пруцков, А.В. Морфологический анализ и синтез в телекоммуникационных технологиях и автоматизированном обучении [Текст] / А.В. Пруцков // Информационные и телекоммуникационные технологии. Подготовка специалистов для инфокоммуникационной среды: материалы 34-й Всерос. науч.-техн. конф. В 2 ч. / Рязан. высш. военное командное училище связи. – Рязань, 2009. – Ч. 1. – С. 200-202.

53. Пруцков, А.В. Методы обработки словоформ и числительных в автоматической обработке текстов [Текст] / А.В. Пруцков // Проблемы передачи и обработки информации в сетях и системах телекоммуникаций: материалы 16-й Междунар. науч.-техн. конф. / Рязан. гос. радиотехн. ун-т. – Рязань, 2010. – С. 149-151.

54. Пруцков, А.В. Применение разработанного метода морфологического анализа и синтеза в системах искусственного интеллекта [Текст] / А.В. Пруцков // Интеллект и наука: тр. 9-й Междунар. науч.-практ. конф. – Красноярск: Центр информации, 2011. – С. 66-67.

55. Пруцков, А.В. Метод преобразования и перевода количественных числительных естественных языков [Текст] / А.В. Пруцков // Математические методы и информационные технологии в экономике, социологии и образовании: сб. статей 27-й Междунар. науч.-техн. конф. – Пенза: Приволжский Дом знаний, 2011. – С. 69-70.

56. Пруцков, А.В. Информационные системы в автоматизации обучения и проверки знаний морфологии естественных языков [Текст] / А.В. Пруцков // Инновации и традиции науки и образования: 2-я Всерос. науч.-метод. конф. – Сыктывкар: Сыктывкар. гос. ун-т, 2011. – Ч. 2. – С. 261-263.

57. Пруцков, А.В. Применение информационных систем анализа нормативных актов и договоров в обучении юридическим дисциплинам [Текст] / В.В. Александров, А.В. Пруцков, А.С. Шустов // Материалы 3-й Всероссийской научно-методической конференции «Методы обучения и организация учебного процесса в вузе». – Рязань, 2013. – С. 148-150.

58. Пруцков А.В. Автоматизированный анализ и оценка коллективных договоров [Текст] / А.В. Пруцков, А.С. Шустов // Сборник материалов XII Всероссийской научно-технической конференции студентов, магистрантов, аспирантов и молодых ученых «Техника XXI века глазами молодых ученых и специалистов». – Тула: Изд-во ТулГУ, 2013. – С. 63-64.

59. Prutzkow A.V. The Algorithms of Wordform Generation and Recognition. International Conference on Computer Technologies in Physical and Engineering Applications 2014 (ICCTPEA-2014), p. 151.

ПРУЦКОВ Александр Викторович

МОДЕЛИ, МЕТОДЫ И ПРОГРАММЫ
АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ФОРМ СЛОВ
В ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ИНТЕРФЕЙСАХ

Автореферат

диссертации на соискание учёной степени
доктора технических наук

Подписано в печать 20.09.2015. Формат бумаги 60×84 1/16.

Бумага офсетная. Печать трафаретная. Усл. печ. л. 2,0.

Уч.-изд. л. 2,0. Тираж 100 экз. Заказ 3201

Рязанский государственный радиотехнический университет.

390005, г. Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ