На правах рукописи

ЛОГАЧЕВА Варвара Константиновна

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ АВТОМАТИЗАЦИИ ПРОЦЕССОВ ПРАКТИЧЕСКОЙ ТРАНСКРИПЦИИ ИМЕН СОБСТВЕННЫХ

05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Автореферат диссертации на соискание ученой степени кандидата физико-математических наук

Москва – 2013

Работа выполнена в Институте прикладной математики им. М.В. Келдыша РАН

Научный руководитель: кандидат технических наук, доцент,
Московский институт электроники и математики национального исследовательского университета «Высшая школа экономики», доцент кафедры информационных технологий и автоматизированных систем
Клышинский Эдуард Станиславович

Официальные оппоненты: доктор физико-математических наук, профессор,
ведущий научный сотрудник Института прикладной математики им.М.В.Келдыша Российской академии наук
Карташев Владимир Алексеевич;

кандидат физико-математических наук,
Московский государственный университет им.М.В.Ломоносова, старший преподаватель кафедры алгоритмических языков
Вылиток Алексей Александрович

Ведущая организация: Государственный научно-исследовательский институт авиационных систем

Защита состоится «16» апреля 2013 года в 11:00 час. на заседании диссертационного совета Д 002.024.01, созданного на базе Института прикладной математики им. М.В. Келдыша РАН по адресу: 125047, Москва, Миусская пл., д. 4.

С диссертацией можно ознакомиться в библиотеке Института прикладной математики им. М.В. Келдыша РАН

Автореферат разослан «____» _____ 2013 г.

Ученый секретарь диссертационного совета,
доктор физ.-мат. наук Т.А. ПОЛИЛОВА

Актуальность темы

В настоящее время в состав комплексных систем обработки и анализа данных всё чаще включаются подсистемы обработки текстовой информации. Если такие подсистемы предназначены для работы с данными на нескольких языках, перед ними может ставиться задача проведения автоматической транскрипции (передачи написания имени собственного с одного языка на другой с сохранением его звучания). В качестве примера можно привести преобразование больших списков имен, ручная обработка которых занимает много времени. Разработка средств автоматизации является полезной даже при небольших объемах данных, так как в этом случае исключается влияние человеческого фактора: устраняется возможность совершения ошибок, допущенных по невнимательности, отсутствуют расхождения в правилах транскрипции, используемых разными пользователями и так далее.

При построении систем автоматической транскрипции в первую очередь встает вопрос о принципах их работы. Первые системы машинной транскрипции использовали уже имевшиеся наработки в этой области – имеющиеся в литературе правила практической транскрипции. Таким образом, эти системы просто применяли правила транскрипции, написанные вручную. Такие системы широко используются и сейчас, так как справляются со стоящей перед ними задачей и отвечают поставленным перед ними требованиям: транскрибируют имена быстрее и аккуратнее, чем человек. Редактируемые правила транскрипции имеют то важное достоинство, что они позволяют исследователю свободно расширять их список при обнаружении новых правил или изменении предпочтений. Более того, каждое вновь введенное правило расширяет научное знание и может использоваться другими исследователями в дальнейшем.

Однако такая автоматизация процесса транскрипции часто недостаточна. Существующие на сегодняшний день руководства содержат правила транскрипции для сравнительно небольшого числа языков. Ручное составление правил, пригодных для машинной обработки – однократный процесс, но он довольно долог и трудоемок. В условиях постоянно расширяющихся международных контактов, требующих составления правил между всё новыми и новыми парами языков, требуется решение задачи автоматического создания правил транскрипции. При автоматическом создании правил, то есть обучении системы транскрипции, в качестве обучающих данных используется множество имен на языке оригинала и сопоставленных им переводов на целевой язык. В исследованиях по компьютерной лингвистике такие множества текстов на двух и более языках называются параллельными корпусами текстов. Иностранные исследователи также проводят обучение с использованием параллельных корпусов, однако в целях повышения качества обучения отказываются от правил, записанных в явном виде, в пользу статистических моделей транскрипции. Автор данной работы придерживается противоположного принципа: автоматически порожденные правила должны быть представлены в явном виде, чтобы

сделать возможным их ручное редактирование. При этом оба подхода едины в понимании того, что автоматическое извлечение правил транскрипции (в явном или в неявном виде) является обязательной частью системы машинной транскрипции. Таким образом, является весьма актуальной задача разработки комплексных систем автоматизированной транскрипции, позволяющих не только транскрибировать имена собственные по правилам транскрипции, но и проводить автоматическое извлечение таких правил при обучении.

Цель диссертации

Целью диссертационной работы является повышение эффективности обработки документов за счет автоматизации и ускорения процессов практической транскрипции (автоматизированной генерации правил и преобразования имен собственных по этим правилам).

Для достижения поставленной цели необходимо решить следующие задачи.

- Проанализировать существующие методы и программные системы машинного обучения практической транскрипции с целью выявления возможности их практического применения;
- Проанализировать существующие методы машинной транскрипции с целью выявления технологий, обеспечивающих наиболее быстрое преобразование строк;
- Разработать метод автоматической генерации правил транскрипции на основе параллельного обучающего корпуса;
- Разработать метод транскрипции имен собственных по набору правил транскрипции, отличающийся повышенной скоростью работы по сравнению с существующими аналогами;
- Разработать программную систему машинной транскрипции, использующую предложенные методы.

Научная новизна результатов работы

На сегодняшний день наиболее распространенными подходами к решению задачи практической транскрипции являются реализация программной системы, применяющей статистическую модель транскрипции, созданную с помощью методов машинного обучения, или автоматическая транскрипция с помощью написанных вручную правил. Оба подхода имеют недостатки, ограничивающие качество транскрипции.

Для преодоления этих недостатков автором была разработана программная система, реализующая новую методику создания системы транскрипции с исходного языка по параллельному корпусу имен и включающая в себя:

- Подсистему автоматической генерации правил транскрипции на основе параллельного обучающего корпуса для произвольной пары

языков. Подсистема основана на новом предложенном автором методе, преимущество которого перед другими методами машинного обучения состоит в возможности обучения на сравнительно небольшом обучающем корпусе. В отличие от статистических моделей машинной транскрипции, правила генерируются в явном виде и в случае необходимости могут затем редактироваться вручную;

- Подсистему транскрипции строк с исходного языка на целевой с использованием системы правил. Эта подсистема также основана на новом методе, разработанном автором. Скорость работы метода, в противоположность существующим методам транскрипции с помощью правил, не зависит от объема системы правил и линейна относительно длины преобразуемой строки.

Практическая ценность результатов

Предложена и реализована новая методика, позволяющая в автоматическом режиме создавать правила транскрипции и эффективно применять их для преобразования строк. Программная реализация методики позволяет быстро получить модель транскрипции для пары произвольных языков, обеспечивающую высокое качество транскрипции, даже при небольшом количестве обучающих данных и применить ее для преобразования строк с высокой скоростью.

Апробация работы и публикации

Основные положения диссертационной работы докладывались и обсуждались на конференции по искусственному интеллекту КИИ-2010 и международной конференции по компьютерной лингвистике «Диалог-2011», а также на научно-практическом семинаре «Новые информационные технологии в автоматизированных системах» в 2010 г., на семинаре в ГосНИИАС в 2011 г. и на научном семинаре направления «Программирование» им. М.Р. Шура-Бура ИПМ им. М.В.Келдыша РАН.

По результатам работы имеется 6 публикаций, включая 2 статьи в рецензируемых научных журналах из списка ВАК [1, 2], 4 статьи в сборниках докладов на международных научных конференциях и семинарах [3-6].

Структура и объем диссертации

Работа состоит из введения, четырех глав и заключения. Основная часть работы изложена на 132 страницах машинописного текста, содержит 9 таблиц и 23 рисунка. Список литературы включает 129 наименования.

СОДЕРЖАНИЕ РАБОТЫ

Во введении дается обоснование актуальности темы диссертационной работы, определяется направление исследования, формулируются цели и задачи исследования, определяются научная новизна и практическая значимость результатов.

В первой главе рассмотрены различные подходы к решению задачи автоматизации практической транскрипции.

Сама проблема создания системы практической транскрипции содержит в себе две задачи: задачу создания модели транскрипции, то есть системы явных или неявных правил преобразования строк с исходного языка на целевой, и метода транскрипции, то есть способа применения этих правил.

Исходя из этой особенности, почти все системы транскрипции имеют стандартную двухчастную структуру: они состоят из модуля обучения, порождающего модель, и модуля транскрипции, получающего готовую модель и генерирующего транскрипцию для поступающих на вход строк на исходном языке. Модуль обучения, в свою очередь, состоит из двух процедур: выравнивания и собственно обучения. Выравнивание – важный этап обучения, состоящий в сопоставлении символам или подстрокам исходного языка символов и подстрок целевого языка. Большинство исследователей пользуется процедурами выравнивания, предоставляемыми статистическими машинными переводчиками (например, GIZA++). Однако некоторые системы транскрипции пользуются другими алгоритмами выравнивания, разработанными специально для этой задачи (Covington, Kang и Choi, Gao и др.).

В целом все известные решения можно охарактеризовать по следующим параметрам:

- Структура правил транскрипции. Преобразование строк может быть основано на сопоставлении фонем исходного языка фонемам целевого языка (Knight, Jeong, Jung, Oh), или на сопоставлении символов или подстрок исходного алфавита символам целевого алфавита (Kang, Sherif);
- Методы генерации правил. Самым распространенным подходом к обучению модели транскрипции является использование статистических методов машинного обучения: порождающих (Knight, Gao, Sproat) или, реже, дифференциальных (Zelenko). Применяются также алгоритмические методы (Karimi, Linden), и эмпирические методы – то есть генерация правил вручную (Бондаренко, Клышинский, Arbabi);
- Вид правил. Правила могут быть стохастическими, то есть описывать межъязыковые соответствия фонем/подстрок с некоторой вероятностью (Knight и почти все работы, основанные на статистических методах), или детерминированными – предполагающими однозначные соответствия (такой вид имеют обычно правила, составленные вручную);

- Тип обучающих данных. В некоторых исследованиях используются созданные вручную корпуса имен и их переводов, другие извлекают информацию о переводе имени из двуязычных словарей терминов (Knight, Huang, Haizhou, Li), из параллельных текстов (Sherif и Kondrak, Goldwasser и Roth). В качестве дополнительной информации привлекаются фонетические словари (CMU Pronunciation dictionary). Существуют методы обучения по одноязычному корпусу (Ravi).

Первые системы машинной транскрипции использовали правила, созданные вручную (Arbabi). Они обладали высоким качеством транскрипции, но из-за сложности ручного составления правил такой подход используется только в тех случаях, когда требуется преобразование строк между парой языков, и расширение системы не планируется (например, Malik, система перевода между письменностями шахмукхи и гурмукхи пенджабского языка).

Наиболее популярными методами обучения транскрипции стали порождающие методы. Порождение модели транскрипции осуществляется с помощью различных алгоритмов обучения: алгоритма Витерби (Sherif), алгоритма Expectation Maximization (Knight), нейронных сетей (Jeong), скрытых цепей Маркова (Jung), алгоритма наименьших средних квадратов (Lin).

Дифференциальные методы, рассматривающие транскрипцию как задачу классификации, не получили широкого распространения. Многие из них обладают весомым достоинством – не требуют процедуры выравнивания, информация о правилах преобразования извлекается из пары строк без разделения на более мелкие единицы, но по качеству обучения они не могут соперничать с порождающими алгоритмами. Дифференциальные методы описаны в работах Bellare, Zelenko, Cherry.

Следует отметить существенный недостаток статистических методов – необходимость большого количества обучающих данных. Качество обучения зависит от качества и объема обучающего корпуса.

Еще один недостаток существующих решений – отсутствие универсальности, то есть возможность работать только с одной или несколькими фиксированными парами языков. Лишь некоторые методы предназначены для генерации правил более чем для одной пары языков. Многие исследователи утверждают, что их методы универсальны, однако большое количество уточнений для некоторого конкретного языка и отсутствие экспериментов с именами других языков заставляет усомниться в универсальности метода. Не отличается разнообразием и выбор языков: внимание исследователей привлекают прежде всего китайский, корейский, арабский и японский языки, в качестве целевого языка в подавляющем большинстве случаев выступает английский.

Однако самым главным недостатком статистических методов обучения является закрытость получаемой модели транскрипции. Ошибки могут быть исправлены только расширением обучающего корпуса, причем результат такого расширения непрогнозируем. Модели, основанные на правилах,

записанных в явном виде, с точки зрения редактирования гораздо удобнее вероятностных моделей, но они имеют другой недостаток – ручное составление правил трудоемко, а методов автоматического порождения таких правил пока не было предложено. Как следствие возникает необходимость в создании нового метода автоматического порождения правил транскрипции, записанных в явном виде.

Во второй главе описан метод генерации системы правил транскрипции имен собственных по двуязычному обучающему множеству имен.

Одной из задач исследования является создание метода обучения модели межъязыковой транскрипции. В качестве модели была выбрана система правил. Под обучением в данном случае понимается извлечение системы правил транскрипции из обучающего корпуса. Преобразование должно осуществляться напрямую из строк исходного языка в строки целевого языка, без промежуточного фонетического представления. Правила должны быть записаны в явном виде, чтобы обеспечить возможность дальнейшего ручного редактирования. При разработке формата правил за основу был взят формат, использовавшийся в системе «Трансскриба». Правило определяется как пара $\mathbf{r} = <\mathbf{p}, \beta>$, где:

p – левая часть правила – преобразовываемая цепочка символов входного алфавита, возможно, с контекстами;

β – правая часть правила – цепочка символов выходного алфавита (возможно, пустая), которая соответствует левой части правила в целевом языке.

Левая часть правила определяется следующим образом:

$\mathbf{p} = <\mathbf{p_l}, \alpha, \mathbf{p_r}>$, где $\mathbf{p_l}$ и $\mathbf{p_r}$ – левый и правый контексты соответственно, α – преобразовываемая строка, $\mathbf{p_l} = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, $\gamma_i \in \mathbf{V_I}^+$, $\mathbf{p_r} = \{\delta_1, \delta_2, \ldots, \delta_m\}$, $\delta_i \in \mathbf{V_I}^+$, где $\mathbf{V_I}$ **–** алфавит исходного языка.

Исходя из этого, правило применимо с текущей позиции, если на текущей позиции находится подстрока α, перед ней полностью представлена одна из подстрок из $\mathbf{p_l}$, после нее – подстрока из $\mathbf{p_r}$. Считается, что к входной строке последовательно ищутся применяемые правила. При нахождении такого правила текущая позиция сдвигается вправо на |α| символов, а на выход подается β (под обозначением |α| здесь и далее понимается длина подстроки α).

Очевидное решение задачи порождения подобных правил – перебор всех возможных соответствий символов и подстрок всех пар имен из обучающего множества. Недостаток такого решения – экспоненциальный рост сложности. В связи с этим становится ясно, что для порождения правил необходима некоторая априорная информация о соответствиях символов.

Такую информацию было решено извлекать из слов, в которых символы находятся во взаимнооднозначном соответствии – предположительно, слова с одинаковым количеством символов в оригинале и переводе. Однако такое предположение не вполне оправдалось – из-за устойчивых сочетаний символов, обозначающих один звук, порождалось большое количество некорректных соответствий символов. Например, в паре «Гжель – Gzhel»

оригинал и перевод одинаковой длины, но из-за отсутствия в английском языке символа для записи звука [ʒ] и противопоставления согласных по наличию или отсутствию палатализации на основе этой пары порождаются соответствия «h – е», «е – л», «l – ь».

В качестве еще одного варианта получения начального набора правил для одиночных символов было рассмотрено разделение пары слов на более мелкие единицы – слоги, при использовании которых из-за меньшей длины снижается вероятность генерации ошибочных соответствий. Здесь были использованы только слоги, которые и в оригинале, и в переводе содержат две буквы: гласную и согласную. Из каждого такого слога извлекалось два правила, на основе которых затем составлялись более сложные правила.

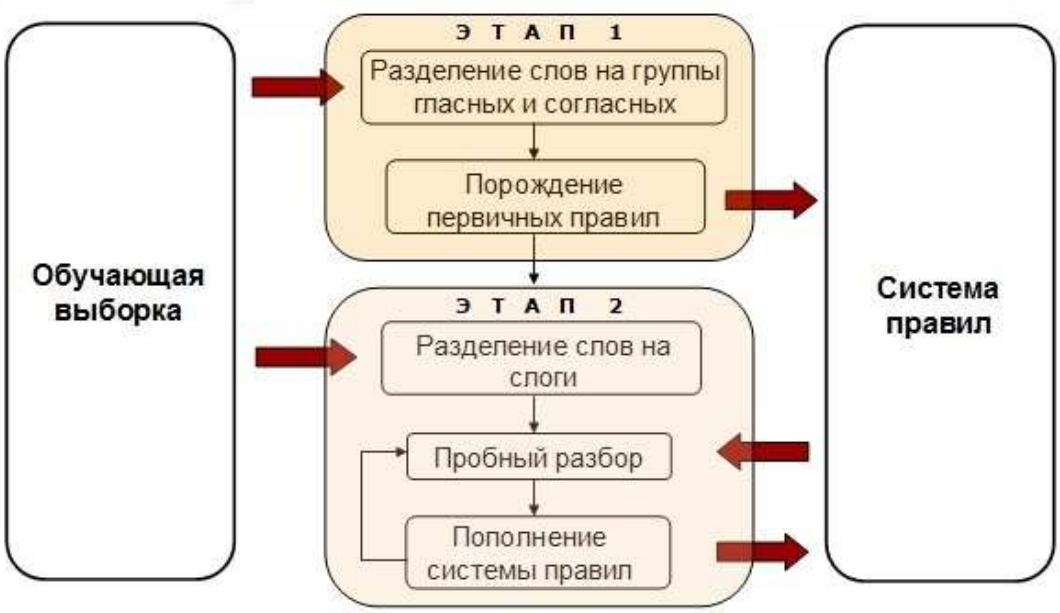

Рис.1. Схема работы метода порождения правил.

Такой алгоритм обеспечивал качество, сопоставимое с другими системами машинной транскрипции, но было предложено его усовершенствование, увеличившее полноту извлекаемых правил: вместо слогов было предложено использование групп гласных и согласных, которые с большей вероятностью находятся во взаимнооднозначном соответствии в оригинале и переводе.

Перейдем к изложению предлагаемого в данной работе метода. Общая схема его работы представлена на рисунке 1.

Метод состоит из двух этапов. Рассмотрим их более подробно.

Этап 1. Порождение первичных правил

Этап состоит из двух подэтапов, изображенных на рис.1: разделение слов на группы гласных и согласных и собственно порождение правил.

Определим предикат isVowel(), который принимает на вход символ входного или выходного алфавита и возвращает **true**, если символ является гласной буквой, и **false** в противном случае. Для разделения имени на группы вызовем предикат для каждого символа оригинала имени и перевода имени.

Границы групп будут располагаться между символами l_i и l_{i+1} такими, что $\text{isVowel}(l_i) \neq \text{isVowel}(l_{i+1})$. Таким образом, в каждой группе окажутся буквы одного типа: гласные или согласные. И имя, и его перевод теперь могут быть представлены как кортежи групп. Для получения системы первичных правил каждой группе из оригинала имени ставится в соответствие группа с тем же номером из перевода имени, в том случае, если тип символов в этих группах совпадает. Множество таких соответствий и образует систему первичных правил. Из множества пар удаляются некоторые пары: слишком длинные, слишком редкие или те, которые могут быть объяснены другими правилами.

В такой системе будет присутствовать неоднозначность: многие строки над входным алфавитом могут быть преобразованы с ее помощью несколькими способами, потому что в системе почти неизбежно найдется хотя бы одна пара правил $\mathbf{r_1}$ и $\mathbf{r_2}$ таких, что $\mathbf{r_1}(\alpha) = \mathbf{r_2}(\alpha)$ и $\mathbf{r_1}(\beta) \neq \mathbf{r_2}(\beta)$. Неоднозначность во многих случаях можно разрешить включением в правила контекстов. К каждому неоднозначному правилу добавляются символы, которые встретились перед (левый контекст) и после (правый контекст) строки α этого правила в именах обучающей выборки.

Добавление контекстов не снимает неоднозначность полностью, так как некоторые символы и сочетания букв могут читаться по-разному даже в одинаковых контекстах. Но эти случаи уже, скорее всего, являются неоднозначностями правил чтения самого языка и могут быть разрешены человеком только путем явного перечисления имен.

Этап 2. Порождение сложных правил

Система, порожденная на предыдущем этапе, для большинства языков не является полной, то есть с ее помощью невозможно получить правильную транскрипцию для имен из обучающей выборки. Требуется порождение дополнительных правил.

Этап порождения сложных правил также состоит из двух подэтапов: разделение имен обучающей выборки на слоги и пробный разбор слогов с помощью уже имеющейся системы правил. Подэтапы могут повторяться неограниченное количество раз, в зависимости от входных данных.

В данном исследовании под термином «слог» понимается группа согласных и следующая за ней группа гласных, то есть граница слога располагается между символами l_i и l_{i+1} такими, что $\text{isVowel}(l_i) = \textbf{true}$ и $\text{isVowel}(l_{i+1}) = \textbf{false}$ с одним исключением: группа согласных, располагающаяся в конце слова, не выделяется в отдельный слог, а присоединяется к предыдущему слогу. Таким образом, аналогично с этапом разделения имени на группы, каждое имя и его перевод могут быть представлены как кортежи слогов. Каждому слогу из оригинала имени ставится в соответствие его перевод – слог с тем же номером из перевода имени. В дальнейшем основной единицей, с которой оперирует алгоритм, будет пара «слог – перевод». В случае, если в оригинале и переводе количество слогов не совпадает, имя сопоставляется переводу целиком.

Каждый слог разбирается с помощью системы правил. Это значит, что среди правил системы ищется правило, чья левая часть совпадает с начальной подстрокой разбираемого слога, а правая часть – с начальной подстрокой перевода разбираемого слога. Если у правила, удовлетворяющего этим условиям, есть контексты, к множеству контекстов добавляется контекст из разбираемого слога. При обнаружении подходящего правила начинается поиск правил для разбора оставшейся части слога. Если ни одного правила не найдено, предпринимается попытка разобрать слог справа налево: происходит поиск правила, чья левая часть совпадает с конечной подстрокой слога, а правая часть – с конечной подстрокой перевода слога.

В результате этого разбора слог либо будет объяснен полностью с помощью существующих правил (в этом случае он больше не участвует в алгоритме), либо может быть представлен в виде пары $< u_1,\ldots,u_i, \boldsymbol{\lambda_x}, u_{i+1},\ldots, u_n > \rightarrow < v_1,\ldots,v_j, \boldsymbol{\mu_x}, v_{j+1},\ldots, v_m >$, где $\boldsymbol{\lambda_x}$ и $\boldsymbol{\mu_x}$ – подстроки оригинала и перевода слога соответственно, которые не могут быть объяснены существующими правилами. На основе частично неразобранного слога строится новое правило, такое, что $\boldsymbol{\lambda_x}$ – левая часть правила $\boldsymbol{\mu_x}$ – правая часть правила.

Система правил, полученная на данном этапе, будет полной относительно обучающего корпуса, то есть все имена обучающего корпуса будут правильно транслитерированы с использованием этой системы правил. Это не означает выделения всех правил преобразования с исходного языка на целевой. Однако проведенные эксперименты позволяют утверждать, что для выделения подавляющего большинства таких правил достаточно небольшого обучающего корпуса (1-2 тыс. имен). В случае же отсутствия применимого в данной ситуации правила используется наиболее частотное правило для текущего символа. Такой подход позволяет преобразовать любую поступившую на вход строку.

Разработанный алгоритм подходит для работы с любой парой языков, использующих алфавитное письмо. Он не требует лингвистической информации, за исключением списка гласных букв входного и выходного алфавитов.

В третьей главе дается описание предлагаемого автором метода транскрипции имен собственных с помощью конечного автомата.

После порождения модели транскрипции – системы правил преобразования – встает вопрос о способе их применения для передачи имен с исходного языка на целевой. Как было показано в обзоре, все существующие методы транскрипции, преобразующие строки с помощью записанных в явном виде правил, обладают не очень высокой (обычно квадратичной) скоростью.

Конечный автомат был выбран в качестве метода проведения транскрипции из-за того, что он обеспечивает линейную скорость разбора строк, причем скорость не зависит от объема системы правил, что особенно важно при работе с языками, характеризующимися сложными зависимостями фонетики и графики (например, английский, французский и пр.).

Автомат, используемый в работе, определяется как кортеж $\mathbf{g} = <\mathbf{V_I}, \mathbf{V_O}, \mathbf{Q}, q_0, \mathbf{F}, \theta>$, где:

$\mathbf{V_I}$ – входной алфавит (алфавит исходного языка);

$\mathbf{V_O}$ – выходной алфавит (алфавит целевого языка);

$\mathbf{Q}$ – множество состояний автомата;

$q_0 \in \mathbf{Q}$ – начальное состояние автомата;

$\mathbf{F} \subset \mathbf{Q}$ – множество конечных состояний;

θ – функция переходов $\theta(q_i, u) = <q_j, n, \mathbf{A}>$, то есть функция θ переводит область определения $\mathbf{Q}\times\mathbf{V_I}$ в область значений $\mathbf{Q}\times\mathbb{Z}\times\mathbf{V_O}^*$. Здесь $q_i, q_j \in \mathbf{Q}$ – это текущее состояние и состояние, в которое должен быть произведен переход, соответственно, $n \in \mathbb{Z}$, где $\mathbb{Z}$ – множество целых чисел – количество символов, на которое должен быть произведен сдвиг по разбираемой строке при переходе, $\mathbf{A} \subset \{\mathbf{V_O}^*\}$ – множество выходных строк, приписанных переходу.

Конечный автомат строится на основе системы правил транскрипции, описанных во второй главе. Каждое правило транскрипции преобразуется отдельно. Преобразование осуществляется по следующему алгоритму:

- Преобразование левого контекста правила:
  - Определяется значение функции $\theta(q_0, l_0) = <q_1, {}^-|\gamma|, \varnothing>$, где l_0 – первый символ α, $\gamma \in \mathbf{p_l}$, q_1 – новое состояние.
  - Для каждой строки $\gamma = <t_1 \ldots t_k>$ левого контекста создаются пути такие, что $\forall\gamma \in \mathbf{p_l} \rightarrow \theta(q_i, t_i) = <q_{i+1}, 1, \varnothing>$, $1 \le i \le k$, причем все пути должны начинаться в состоянии q_1 и заканчиваться в состоянии q_k, но их промежуточные состояния q_j, $1<j<k$, должны быть различными.
  - При отсутствии левого контекста $q_k = q_0$.
- Преобразование строки правила:
  - $\forall l_i \in \alpha \rightarrow \theta(q_{k+i}, l_i) = <q_{k+i+1}, 1, \varnothing>$, $0 \le i < n$,
  - При наличии правого контекста: $\theta(q_{k+n}, l_n) = <q_{k+n+1}, 1, \varnothing>$,
  - При отсутствии правого контекста: $\theta(q_{k+n}, l_n) = <q_f, 1, \beta>$, $q_f \in F$.
- Преобразование правого контекста правила:
  - Создаются пути для каждого правого контекста: $\forall\delta \in \mathbf{p_r}$, $\delta = <s_1 \ldots s_m> \rightarrow \theta(q_{k+n+i+1}, s_i) = <q_{k+n+i+2}, 1, \varnothing>$, $1 \le i < m$, $\theta(q_{k+n+m}, s_m) = <q_f, {}^-|\delta|+1, \beta>$, $q_f \in F$. Аналогично с процедурой создания путей для левых контекстов промежуточные состояния путей для разных δ должны быть различны.
  - $\forall f \in F, \theta(f, \varepsilon) = <q_0, 0, \varnothing>$.

Иными словами, при отсутствии контекстов преобразование правила к конечному автомату состоит в создании пути из начального состояния в конечное по всем символам строки α. При наличии левого контекста создается переход из начального состояния по первому символу из строки α со сдвигом назад по разбираемой строке на длину контекста, затем –

построение путей для всех левых контекстов, и только после этого создается путь по всем символам строки α (включая первый). При наличии правого контекста за путем, построенным по строке α, следуют пути для строк правого контекста, причем при переходе по последнему символу каждой из строк происходит переход назад по строке.

Однако автомат, построенный по такому алгоритму, с большой вероятностью будет недетерминированным (НКА). Для этого достаточно существования в системе пары правил, у которых совпадает первый символ строки α. Разбор может быть проведен и с помощью недетерминированного автомата, но в этом случае будет потеряно преимущество линейной скорости разбора. По этой причине после построения конечного автомата осуществляется его преобразование к детерминированному конечному автомату (ДКА).

Для преобразования расширенного НКА в ДКА может быть использован следующий алгоритм. Для каждой вершины A, из которой исходит n дуг в вершины $B_1,\ldots,B_n$, помеченных одним символом:

- Создается новая вершина A', в которую входит дуга из вершины A, помеченная тем же символом;
- Из вершины A' проводятся все дуги, которые выходили из вершин $B_1,\ldots,B_n$;
- Вершины $B_1,\ldots,B_n$ удаляются.

Этот алгоритм совпадает со стандартной процедурой преобразования НКА в ДКА, за исключением того, что здесь не производится удаления e-дуг (переходов по пустому символу), так как e-дуги необходимы для корректного проведения транскрипции. Не производится и объединения начальных состояний, поскольку начальное состояние данного автомата единственно по построению.

Однако преобразование к ДКА не всегда может быть осуществлено из-за некоторых особенностей конечного автомата:

- Разное значение сдвига двух объединяемых дуг. В этом случае:
    - Необходимо будет продолжать разбор сразу по двум ветвям (что по сложности аналогично НКА);
    - Невозможно определить, какие из дуг нового состояния к какой ветви относятся;
- При объединении конечного состояния с неконечным ветвь, следующая за неконечным состоянием, никогда не будет пройдена, так как из конечного состояния существует переход в начальное по пустому символу.

Для преодоления этих проблем требуется предварительная подготовка системы правил, названная унификацией.

Унификация правил состоит из четырех процедур:

- добавление контекстов по умолчанию. При наличии в системе правил **R** подсистемы $\mathbf{R_i}$ такой, что все левые части правил из $\mathbf{R_i}$ имеют совпадающую начальную подстроку, и при этом длины левых

контекстов у правил из $\mathbf{R_i}$ не совпадают (к этому случаю относится и отсутствие левых контекстов), к контекстам недостаточной длины добавляется символ «*», обозначающий любой символ входного алфавита.

- выравнивание контекстов представляет собой дополнение контекстов (как левых, так и правых) некоторого правила символом «*» в случае, если в множестве контекстов этого правила содержатся строки разной длины.
- удаление неоднозначных контекстов. Неоднозначным считается правый контекст γ правила $\mathbf{r_1}$, такой, что $\mathbf{r_1}(\alpha) + \gamma = \mathbf{r_2}(\alpha)$, где $\mathbf{r_1}, \mathbf{r_2} \in \mathbf{R}$.
- добавление правил по умолчанию. К системе правил добавляются правила с контекстами по умолчанию («*»), где α – каждый символ входного алфавита, а β – самый распространенный перевод этого символа на целевой язык (то есть это правая строка правила с соответствующим α, имеющего наибольшее число употреблений в обучающем множестве).

Первые три пункта направлены на то, чтобы сделать возможным преобразование автомата из НКА в ДКА, четвертый пункт делает систему правил универсальной – то есть позволяет вернуть выходную строку для любой строки над входным алфавитом.

Для всей совокупности процедур в тексте диссертационной работы доказано утверждение о том, что они переводят систему правил в равносильную, то есть возвращающую тот же результат при преобразовании любой строки над алфавитом $\mathbf{V_I}$.

Если система правил была предварительно унифицирована, не обязательно строить по ней сначала недетерминированный конечный автомат, а потом преобразовывать его в детерминированный. Автором была разработана процедура построения детерминированного конечного автомата из системы правил. Она в целом повторяет описанную выше процедуру построения НКА, с той разницей, что новый переход из текущего состояния строится, только если по текущему символу еще нет перехода из этого состояния. Если же переход уже существует, он просто включается в путь для рассматриваемой подстроки, текущим символом становится следующий символ, текущим состоянием – состояние, в которое построен переход.

В работе приведено доказательство эквивалентности процедур преобразования строк с помощью системы правил и с помощью конечного автомата, а также доказательство эквивалентности расширенного ДКА расширенному НКА. Эти утверждения иллюстрируют адекватность конечного автомата как средства применения правил транскрипции, то есть полного соответствия результатов процедуры преобразования строки с помощью конечного автомата ожидаемым результатам.

Кроме того, доказана линейная скорость разбора строк с помощью расширенного автомата, которая может вызывать сомнения из-за

возможности автомата осуществлять сдвиг по разбираемой строке в любом направлении на произвольное количество символов.

В четвертой главе содержится описание программной системы, реализующей разработанные автором методы.

Практическим результатом работы стало создание системы машинной транскрипции. Система предназначена для решения всего комплекса задач, связанных с автоматической транскрипцией имен собственных: она осуществляет обучение модели транскрипции на основе параллельного корпуса имен, предоставленных пользователем, выгружает правила, созданные в результате обучения, в текстовый формат, отличающийся простотой и подходящий для ручного редактирования.

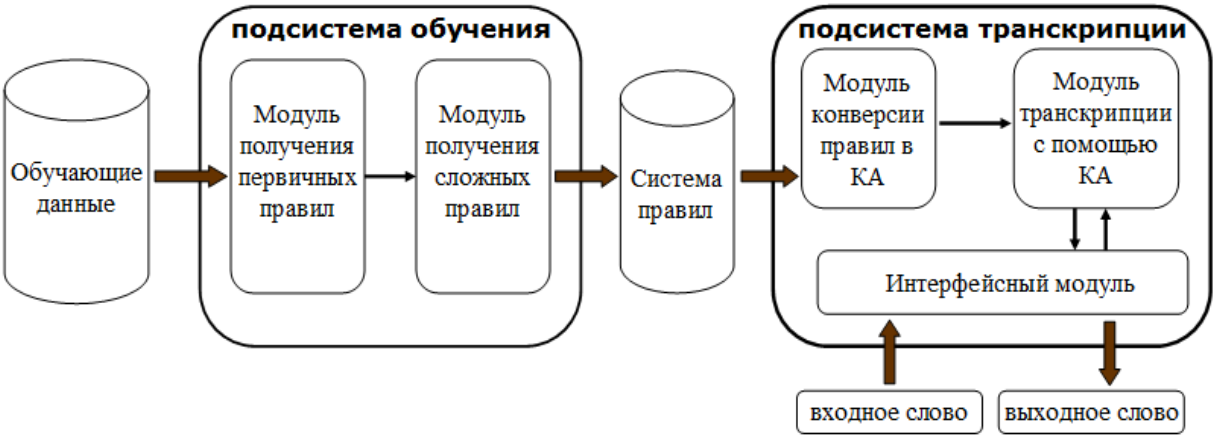

Рис. 2. Архитектура системы транскрипции

Архитектура системы имеет двухчастную структуру. Система состоит из двух подсистем: подсистемы обучения, с помощью которой может быть порождено множество правил транскрипции на основе обучающего множества, и подсистемы транскрипции, служащей для проведения транскрипции имен собственных по некоторому набору правил.

Первая подсистема принимает на вход множество имен и их переводов в текстовом формате, результатом ее работы является текстовый файл с множеством правил. Такой файл (как сгенерированный первой подсистемой, так и созданный или модифицированный пользователем) подается на вход второй подсистеме, которая преобразует правила в конечный автомат, осуществляющий преобразование строк с линейной относительно длины строки скоростью. После построения конечного автомата подсистема транскрипции может осуществлять преобразование как единичных имен, вводимых пользователем с помощью графического интерфейса, так и множеств имен, которые могут быть поданы на вход системе в виде текстового файла.

Подсистемы могут быть использованы как в комплексе, так и независимо друг от друга. Архитектура системы транскрипции представлена на рисунке 2.

Кроме практического применения в программных комплексах обработки информации, создание системы позволило подтвердить экспериментальным путем эффективность предложенных автором методов обучения системы транскрипции и преобразования строк.

Эксперименты проводились на двуязычных базах имен различного объема (от 160 до 7000 имен). Качество обучения метода порождения правил (количество корректно переданных имен обучающей выборки) составило от 95% до 100% на разных базах, качество транскрипции с помощью порожденных правил (количество корректно переданных имен тестовой выборки) – от 85% до 95%. Были проведены эксперименты с однобуквенными и двухбуквенными контекстами. В результате экспериментов было выяснено, что увеличение длины контекстов правил уменьшает неоднозначность системы, но общее качество транскрипции падает. Таким образом, сложно однозначно определить, какая длина контекста обеспечивает лучшее качество, предпочтение того или иного варианта, по мнению автора, зависит от задач, стоящих перед конкретной системой.

Были проведены также эксперименты, подтверждающие ускорение обработки строк с помощью конечного автомата по сравнению с существующими методами. Скорость транскрипции с помощью правил, произведенной с использованием конечного автомата, от 3 до 90 раз больше скорости транскрипции с помощью системы «Трансскриба».

В заключении перечислены основные научные результаты, полученные при решении поставленных задач, а также возможные направления дальнейших исследований: расширение метода генерации правил для возможности работы с языками, использующими консонантное письмо (арабский, иврит), слоговое письмо (японский), идеографическое письмо (китайский).

Основные результаты работы:

- Разработана новая методика транскрипции, позволяющая существенно повысить эффективность обработки документов, содержащих иностранные имена собственные. В состав методики входят:
    - новый метод генерации правил транскрипции по обучающей выборке, позволяющий автоматически генерировать правила транскрипции, записанные в явном виде и доступные для последующей модификации. Метод сокращает время подготовки системы правил специалистами.
    - новый метод транскрипции имен собственных, основанный, в отличие от существующих методов, на преобразовании системы правил к конечному автомату, за счет чего повышается скорость его работы.

- На основе предложенных методов и алгоритмов реализована программная система, автоматизирующая труд разработчиков и пользователей систем машинной транскрипции в области оформления документов, содержащих иностранные имена собственные.

Публикации по теме диссертации:

1. **Логачева В.К., Клышинский Э.С. Метод генерации конечного автомата для задач машинной транскрипции // Научно-техническая информация, серия 2, №1. М.: 2012. – С. 22-29.**
2. **Логачева В.К. Метод порождения правил межъязыковой транскрипции // Научно-техническая информация, серия 2, № 9. М.: 2011. – С. 26-33.**
3. Логачева В.К., Клышинский Э.С., Галактионов В.А. Современные методы практической транскрипции // Препринты ИПМ им. М.В.Келдыша. 2012. № 13. 18 с. URL: http://library.keldysh.ru/preprint.asp?id=2012-13
4. Логачева В.К., Клышинский Э.С., Галактионов В.А. Автоматическая генерация правил транскрипции и машинная транскрипция имен собственных с использованием конечного автомата // Препринты ИПМ им. М.В.Келдыша 2012. № 14. 24 с. URL: http://library.keldysh.ru/preprint.asp?id=2012-14
5. Логачева В.К., Клышинский Э.С. Non-stochastic learning of cross-language transliteration rules from a small dataset // Компьютерная лингвистика и интеллектуальные технологии: По материалам 17-й ежегодной Международной конференции по компьютерной лингвистике «Диалог» (Бекасово, 25-29 мая 2011 г.). Вып. 10 (17).- М.: Изд-во РГГУ, 2011. С. 448-457.
6. Клышинский Э.С., Логачева В.К. Автоматическое порождение правил транскрипции фамильно именных групп // Сб. трудов 12-й Национальной конференции по искусственному интеллекту с международным участием КИИ-2010 (Тверь, 20-24 сентября 2010 г), том 1, М.: Физматлит, 2010. С. 274-282.
7. Логачева В.К., Клышинский Э.С. Самообучающаяся система машинной транскрипции с использованием нестохастического конечного автомата // Сб. трудов XII Всероссийской научной конференции RCDL’2010 (Казань 13-17 октября 2010). Казань: Казанский ун-т 2010 С. 310-314.
8. Логачева В.К. Автоматическое порождение правил транскрипции фамильно-именных групп // Сб. трудов 13 научно-практического семинара «Новые информационные технологии», М., 2010, С. 117-121.