Министерство образования и науки Российской Федерации
Южно-Уральский государственный университет
Кафедра «Общая лингвистика»

Ш.я7
Л84

А.В. Луканин

АВТОМАТИЧЕСКАЯ ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

Учебное пособие

Челябинск
Издательский центр ЮУрГУ
2011

ББК Ш.я7
Л84

Одобрено
учебно-методической комиссией
факультета лингвистики

Рецензенты:
П.И. Браславский, Б.Г. Фаткулин

Луканин, А.В.

Л84 Автоматическая обработка естественного языка: учебное пособие / А.В. Луканин. — Челябинск: Издательский центр ЮУрГУ, 2011. — 70 с.

В учебном пособии рассматриваются вопросы автоматизации анализа и синтеза текста и речи на естественном языке с использованием различных формальных моделей: от задач компьютерной морфологии, дешифрования, аннотирования, до распознавания и синтеза звучащей речи и автоматического перевода.

Пособие предназначено для студентов языковых вузов по специальностям 021800 «Теоретическая и прикладная лингвистика», 035800.62 «Фундаментальная и прикладная лингвистика».

ББК Ш.я7

© Издательский центр ЮУрГУ, 2011

Глава 1. Системы автоматической обработки текста и лингвистический автомат

Начало в истории автоматической обработки текста (АОТ)[1], относящееся к периоду 60–80-х гг. прошлого века, известно как эпоха романтических проектов по созданию искусственного разума и высококачественного машинного перевода (МП), который должен был соперничать по своей результативности с деятельностью высококвалифицированного переводчика. Вторая когнитивная революция второй половины 80-х гг., окончательно разъяснив несостоятельность исходных положений и причины провалов всех этих проектов, поставила точку в истории логистико-романтических подходов к решению проблемы АОТ.

В настоящее время реально работающие системы АОТ представляют собой достаточно грубые аналоги некоторых аспектов речемыслительной деятельности (РМД) человека, объединяемые концепцией так называемого лингвистического автомата (ЛА) (Пиотровский 2006; 2008).

1.1. Лингвистический автомат

ЛА рассматривается как сбалансированный комплекс аппаратных, программных, лингвистических, а иногда и лингводидактических средств, взаимодействующих с мощной базой лингвистических данных и знаний (БДЗ). В идеале ЛА должен обладать следующими свойствами:

- полифункциональностью, т.е. способностью осуществлять разные виды обработки больших потоков текстовой информации,
- минимизацией информационных потерь и ослаблением эффекта отторжения естественного языка (ЕЯ) языком ЛА,
- живучестью, под которой понимается способность ЛА сохранять свои наиболее существенные свойства в результате воздействия на ЛА таких катастрофических факторов, как выход из строя внешних устройств или участков оперативной памяти, искажение отдельных фрагментов текста и т.п.,
- способностью к дальнейшему развитию и усовершенствованию, которые диктуются необходимостью адаптировать ЛА не только к коммуникативно-информационной эволюции общества, но и к прагматике отдельных потребителей,
- возможностью подключения к различным каналам связи, в частности к сети Интернет.

В состав ЛА входят следующие крупные функциональные блоки:

- база данных и знаний;
- блок ввода, распознавания и первичной обработки печатного текста;
- блок распознавания устной речи, преобразующий аудио-сигнал в текст;

[1] Также распространён термин «автоматическая переработка текста» (АПТ). В частности, Р.Г. Пиотровский использует в своих работах именно его, т.к. предполагается, что текст в ходе переработки должен изменяться или дополняться служебной информацией.

- блок индексирования и аннотирования текста;
- блок машинного перевода;
- блок вывода печатного текста;
- блок, озвучивающий печатный текст;
- средства человеко-машинной синергетики и блок управления.

В самом общем виде обработка текста в ЛА выглядит следующим образом. Принятый модулем ввода и коррекции текст поступает на вход лексико-морфологического анализатора, который с помощью соответствующего подблока БДЗ создает подстрочник. Подстрочник поступает в подблоки синтаксического и семантического анализа. Там он разбивается на именные и глагольные группы, каждая из которых анализируется и обрабатывается с использованием графов переходов (конечных автоматов, см. главу 4).

Полученный результат передаётся в подблок порождения выходного текста (перевода, аннотации либо индекса текста), а затем выдаётся потребителю.

Многофункциональная управляющая оболочка системы, обеспечивающая ввод текста, его машинную обработку и выдачу пользователю, оформлена в виде автоматизированного рабочего места (АРМ) пользователя.

Следует отметить, что в ЛА формируется интегрированная среда, не выходя из которой пользователь может создать и полностью оформить качественную профессиональную обработку текста. Эта среда даёт возможность оперативно реализовать всю технологию обработки документа, начиная с его ввода в компьютер, пред-, интер-, постредактирования и заканчивая получением готового перевода, аннотации или индекса текста необходимого уровня на требуемом языке.

Новые технологии позволяют использовать в полной мере возможности, предоставляемые такими считывающими устройствами, как карманный и главный сканеры и программы проверки орфографии, а также адаптером для связи с компьютером. Предоставляя пользователю быстрый доступ к любому из разделов системы, эти средства дают возможность:

- получать быстрый пословный перевод (подстрочник);
- переводить фрагменты текста не только с использованием типовой БДЗ, но также путём обращения к собственным словарям и другим внешним источникам (например, расположенным в сети Интернет);
- ускорить рутинную работу с традиционными словарями, когда найденный в словаре перевод слова, словосочетания или целого фрагмента с помощью ручного сканера может быть «вклеен» на экране в нужное место текста и введён в переводческую память или автоматический словарь.

И что самое главное, эти технологии дают возможность лингвисту и программисту оперативно корректировать перевод, дополнять БДЗ и лексико-грамматические алгоритмы ЛА.

1.2. Человеко-машинная синергетика и самоорганизация в системах АОТ

Говорить о синергетичности современных систем автоматической обработки устной и письменной речи, предусматривающей их саморегуляцию, саморазви-

тие, устойчивость, живучесть, а также другие синергетические функции и свойства, пока ещё рано (Луканин 2008б). И всё же уже сейчас делаются попытки создать для ЛА и других систем АОТ независимые от человека синергетические модули. Среди них наибольший интерес представляют следующие модули, подблоки и архитектурные решения (Пиотровский 2008):

- уровневые построения систем АОТ;
- модуль распознания языка, на котором написан текст или его фрагменты;
- подблок опознания формата текста и его частей, а также определения их жанровой и тематической принадлежности;
- средства, обеспечивающие живучесть и устойчивость ЛА при обработке текста;
- фреймовые модули и подблоки (см. главу 2).

1.3. Уровневое построение систем АОТ и ЛА

Лингвистический автомат, как и другие системы автоматической обработки устной или письменной речи, строится как многоуровневая система, в которой поступающие на вход сообщения последовательно обрабатываются разными модулями и подблоками, образующими автономные уровни. В этот процесс последовательно вступают сначала уровни, осуществляющие анализ плана выражения для того, чтобы передать получаемые результаты на более высокие, также иерархически соотносящиеся уровни. К последним относятся лексико-морфологический (словарный), синтаксико-семантический и ситуативно-прагматический уровни. Эта иерархическая организация ЛА работает наиболее эффективно в среде подъязыков (см. главу 3), тексты которых имеют жёстко стандартное построение. К таким подъязыкам относятся тексты инструкций, патентов и т.п. Указанные подъязыки либо являются номенклатурами, обладающими ограниченным набором реплик, либо строятся как исчисления, с помощью которых можно автоматически распознавать каждый фрагмент текста. Для этого используются заранее заданные фильтрующие эталоны для разрешённых акустико-графических, лексико-грамматических и других ситуаций (Пиотровский 2006, с. 119–125).

1.4. Опознание языка документа и нормализация входного текста

Эксплуатация систем АОТ показывает, что обрабатываемые в масштабе реального времени большие текстовые массивы (например, документы ЕС или НАТО) могут включать тексты на разных европейских языках. Если же говорить о восточных текстах, то для них характерны вставки на других языках, чаще всего на английском. При этом переход с языка на язык ни в том, ни в другом случае никак формально не отмечается. Поэтому начальным модулем в рассматриваемом блоке должен быть подблок опознания языка.

Задача опознания может решаться либо детерминистским способом с помощью графических индикаторов, либо путём вероятностного отнесения текста к

одному из заранее сообщённых автомату языков. В качестве графических индикаторов может выступать сам алфавит, например, японская хирагана, либо отдельные специфические графемы (например, сербские буквы Ђ, Љ) или диакритические знаки типа французских «accents» (´, `, ^).

Если же представленные в наборе ЛА языки пользуются графикой, не содержащей букв, которые могли бы стать индикатором того или иного языка, или система получает текст, в котором снята диакритика, то в этом случае применяется вероятностное отнесение к определённому языку. В качестве индикаторов используются частотные словоформы, при условии, что каждая из них характеризует только один язык из имеющихся в наборе ЛА. Так, из первых пяти наиболее частых английских словоформ — *the, of, and, to* и *in*, только первые четыре могут выступать в качестве показателей принадлежности текста к английскому языку. Что касается словоформы *in*, то она таким индикатором быть не может, поскольку используется в немецком, итальянском и других языках (Пиотровский 2006, с. 116–117). Если же говорить о распознавании языка транслитерированного текста, то выбранные показатели могут привести к ошибкам системы. К примеру, в персидском языке, который в нормальном состоянии использует арабицу, есть частотные транслитерированные слова *to* и *in*. В последнем случае, для определения языка можно использовать вероятности следования букв друг за другом (биграммы, см. раздел 6.4).

Опознание языка текста или его сегментов даёт возможность лингвистическому автомату, реализуя обратную связь, вызывать для обработки опознанного фрагмента нужное лингвистическое обеспечение.

1.5. Подблок опознания формата текста и его частей, а также определение их жанровой и тематической принадлежности

Этот подблок справляется со своими задачами без участия человека только при обработке жёстко структурированных текстов. Так, общая жанровая принадлежность и формат спецификаций или ведомостей автоматически определяются путём выявления в тексте большого числа следующих друг за другом таблиц. Для патентов такими сигналами служит использование стандартных заголовочных шаблонов типа *Страна патентования, Номер патента, Заявитель* или *Наименование темы, Характер НИР, Организация, Исполнитель*.

Сложнее обстоит дело при опознании конкретной тематики не жёстко структурированного текста. Эта задача решается путём автоматического анализа вероятностного соотнесения ключевых слов и словосочетаний документа с заранее заданными словарными эталонами, привязанными к определённым стилям, предметным областям, подъязыкам или тематическим рубрикам. Автоматическое соотнесение анонимного текста с заранее описанными лексико-грамматическими особенностями и стилевыми манерами определённых авторов позволяет ЛА осуществлять достаточно надёжную атрибуцию таких текстов.

Экспериментальная и промышленная эксплуатация систем индексации и аннотации текстов, построенных на описанных принципах, которая проводилась в 80–

90-х гг. показала, что основным синергетическим недостатком этих систем является их изначально заданная жёсткая структура. Она не позволяет реагировать на изменения в содержании и глубинной организации конкретного текста, не говоря уже о разнообразии семантики и форматов текстов, которые образуют информационные потоки, обрабатываемые системой в режиме реального времени. Жёсткость этих систем затрудняет также их адаптацию к терминологическому развитию предметных областей, т.е. подъязыков. Поиски преодоления этих недостатков ведутся по двум направлениям: во-первых, путём использования динамичных сценарных фреймов, с помощью которых АОТ последовательно анализирует все фрагменты документа, а во-вторых, — с помощью последовательной семантико-прагматической рубрикации этих фрагментов. К сожалению, создание массива сценарных фреймов, который был бы достаточен для эффективного анализа больших текстовых потоков, наталкивается на кризис размерности. Поэтому более реалистичным приёмом оказывается семантико-прагматическая рубрикация с помощью лексических сигнализаторов, в качестве которых используются словоформы и словосочетания, способные с достаточно высокой степенью вероятности указывать на появление в тексте (или в потоке текстов) новых ситуаций, которые соответствуют одной из заданных тематических рубрик. Выбор в качестве сигнализаторов ключевых словоформ и словосочетаний и их систематизация производится на основе предварительного лингво-статистического исследования обучающих тематических выборок. Кроме того используется получаемая от экспертов информация о референтных ситуациях. Более или менее надёжными сигналами обратной связи, которые указывают ЛА на появление в тексте новой ситуации, служат:

- изменение формата текста;
- статистически значимые отклонения в использовании ключевых лексических единиц от их употреблений в предыдущих фрагментах текста (эти отклонения сигнализируют обычно об изменениях в семантических и прагматических намерениях отправителя сообщения);
- «скучивание», т.е. резкий рост на определённом участке текста употребительности ключевых лексических единиц-нотонимов, которые традиционно обозначают известные в данной предметной области понятия;
- «скучивание» не зафиксированных в словаре терминов, выражающих новые научно-технические понятия, так называемые неонимы.

При всей своей простоте описанная технология используется в человеко-машинных системах текущего контроля над потоками документов, проходящих по разным каналам связи, в плане их общих смысловых и прагматических ориентаций и с точки зрения соответствия этих ориентаций привычной смысловой и прагматической структуре потока. Более того, на этой основе могут строиться системы АОТ, которые осуществляют оперативное слежение за семантикой, прагматикой и направлениями информационных потоков, отражающих динамику экономических или геополитических интересов конкурента или потенциального противника (Пиотровский 2006, с. 117–119).

1.6. Средства обеспечения живучести ЛА

Опыт эксплуатации коммерческих систем АОТ в режиме реального времени говорит о том, что все такие системы должны быть оснащены собственными, независимыми от пользователя, интер- или постредактора средствами поддержания их живучести. Это значит, что при возникновении сбойных ситуаций в ходе обработки больших потоков текстов ЛА не должен прекращать своей работы. Поэтому, если на $(n+1)$-м уровне автомата происходит отказ от предусмотренного этим уровнем формирования структуры выходного текста, то пользователю выдаются те результаты, которые получены системой на предыдущем n-м уровне. Иными словами, принятие решения опирается на самостоятельную «синергетическую» способность системы к декомпозиции или упрощающей модификации некоторой общей задачи P, предварительно поставленной перед ЛА. В общем случае эта задача представляется в виде множества частных задач

$$P=(P_1,P_2, ..., P_i, ..., P_k),$$

из которых ЛА автоматически выбирает для решения одну частную упрощённую задачу, с тем чтобы, не останавливаясь, продолжить обработку поступающих на вход документов. Позволяя выходить из тупиковых ситуаций, возникающих при отказе системы от заданной формы обработки текста, декомпозиция и упрощение P обеспечивает живучесть ЛА (Пиотровский 2006, с. 126).

Вопросы к семинару

1. Архитектура лингвистического автомата.
2. Принципы функционирования модуля машинного перевода.
3. Устройство обучающего лингвистического автомата.
4. Два подхода к описанию обучающего лингвистического автомата.
5. Лингвистический автомат и человеко-машинный обучающий диалог.

Литература к семинару

1. Пиотровская, К.Р. Обучающий лингвистический автомат: учебное пособие к спецкурсу «Основы квантитативной лингводидактики» / К.Р. Пиотровская — СПб.: Изд-во ИНТЕРЛАЙН, 2002. — С. 5–11.
2. Пиотровский, Р.Г. Лингвистический автомат (в исследовании и непрерывном обучении): 2-е изд., доп. и испр. / Р.Г. Пиотровский — СПб.: Изд-во РГПУ им. А. И. Герцена, 2008. — http://www.silod.org/txt/text2.zip — Главы 6–7.

Глава 2. Фреймы в системах автоматической обработки текстов

Одним из популярных приёмов автоматической обработки документа в ЛА и в других системах АОТ является использование заранее заготовленных шаблонов-ожиданий или фреймов. Этот приём был подсказан машинной метафорой человеческого интеллекта, предложенной когнитологией 70-х и начала 80-х гг. XX в. Согласно этой гипотезе мышление человека оперирует фреймовыми структурами знаний разной организации — планами, сценариями, схемами. Аналоги этих структур и следовало, по мнению теоретиков искусственного разума 50–80-х годов использовать в системах АОТ и искусственного интеллекта (ИИ). Вторая когнитивная революция, поставила под сомнение предположение о жёстко фреймовом характере человеческого мышления. Однако она никак не отменяет применение фреймовых шаблонов и матриц при построении систем АОТ (Пиотровский 2006, с. 127).

2.1. Основные определения

Термин фрейм был предложен Марвином Минским, одним из пионеров ИИ, в 70-е годы для обозначения структуры знаний для восприятия пространственных сцен.

«Фрейм любого вида — это та минимально необходимая структурированная информация, которая однозначно определяет данный класс объектов» (Минский 1979, с. 3).

«Фрейм является структурой данных для представления стереотипной ситуации» (Минский 1979, с. 7).

В случае возникновения конкретной ситуации, например, необходимости совершить роботом, управляемым ЭВМ, определённое действие, воспринять с помощью сенсоров, связанных с ЭВМ, какой-то зрительный образ и т.д., из памяти ЭВМ должен быть выбран фрейм, соответствующий данному классу ситуаций и согласован с рассматриваемой конкретной ситуацией из этого класса путём изменения подробностей, т.е. путём конкретизации данных из набора, которые могут удовлетворить выбранный фрейм (Минский 1979, с. 128).

Фрейм можно представить в виде сети, состоящей из узлов и связей между ними. Каждый узел представляет собой определённое понятие, которое — и в этом заключается основной смысл теории — может быть, а может и не быть задано в явном виде. «Верхние уровни» фрейма чётко определены, поскольку образованы такими понятиями, которые всегда справедливы по отношению к предполагаемой ситуации. На более низких уровнях имеется много особых вершин-терминалов или «ячеек», которые должны быть заполнены характерными примерами или данными, т.е. данные узлы не заданы явно.

Одни и те же терминалы могут входить в состав нескольких фреймов системы — это один из центральных моментов теории, позволяющий согласовывать информацию, поступающую из различных источников.

Таким образом, совокупность заданных в явном виде узлов — понятий — образует основу для «понимания» любой конкретной ситуации из определённого для данного фрейма класса ситуаций. «Понимание» происходит путём конкретизации терминалов и согласования возможных для каждого из них понятий с вполне определённой, существующей во внешнем мире обстановкой.

Системы фреймов связаны, в свою очередь, сетью поиска информации. Если предложенный фрейм нельзя приспособить к реальной ситуации, т. е. если не удаётся найти такие задания терминалов, которые удовлетворяют условиям соответствующих маркеров, сеть поиска информации позволяет выбрать более подходящий для данной ситуации фрейм.

В своей теории М. Минский не разграничивает теорию человеческого мышления и теорию искусственного интеллекта. Он полагает, что процессы человеческого мышления базируются на хранящихся в его памяти материализованных, многочисленных запомненных структурах данных — фреймах, с помощью которых человек осознает зрительные образы (фреймы визуальных образов), понимает слова (семантические фреймы), рассуждения, действия (фреймы-сценарии), повествования и т.д. Процесс понимания при этом сопровождается активизацией в памяти соответствующего фрейма и согласованием его терминальных вершин с текущей ситуацией. В случае неудачи из памяти с помощью сети поиска информации, объединяющей системы фреймов, «выбирается» другой фрейм, терминалы которого, возможно, окажутся между собой в более подходящих отношениях применительно к той же рассматриваемой ситуации (Минский 1979, с. 7–11).

2.2. Типы и структура фреймов

Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить всё многообразие знаний о мире через:

- фреймы-**структуры**, использующиеся для обозначения объектов и понятий;
- фреймы-**роли** (менеджер, кассир, клиент);
- фреймы-**сценарии** (банкротство, собрание акционеров, празднование именин);
- фреймы-**ситуации** (тревога, авария, рабочий режим устройства) и др.

Традиционно структура фрейма может быть представлена как список свойств:

(ИМЯ ФРЕЙМА:
(имя 1-го слота: значение 1-го слота),
(имя 2-го слота: значение 2-го слота),
…
(имя N-го слота: значение N-го слота)).

Ту же запись можно представить в виде таблицы (табл. 2.1), дополнив её двумя столбцами.

В таблице дополнительные столбцы предназначены для описания способа получения слотом его значения и возможного присоединения к тому или иному слоту специальных процедур, что допускается в теории фреймов. В качестве значения слота может выступать имя другого фрейма, так образуются сети фреймов.

Таблица 2.1

Имя фрейма: Ученик			
Имя слота	**Значение слота**	**Способ получения значения**	**Присоединённая процедура**
АКО	Ребёнок		
Возраст	7–17 лет	Присоединённая процедура	«Вычислить на основе класса школы»
Рост		АКО	
Класс		Спросить у пользователя	

Существует несколько способов получения слотом значений во фрейме-экземпляре:

- по умолчанию от фрейма-образца (Default-значение);
- через наследование свойств от фрейма, указанного в слоте АКО;
- по формуле, указанной в слоте;
- через присоединённую процедуру;
- явно из диалога с пользователем;
- из базы данных.

Важнейшим свойством теории фреймов является заимствование из теории семантических сетей — так называемое наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО-связям (A-Kind-Of = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся, значения аналогичных слотов.

Например, в сети фреймов на рис. 2.1 понятие «ученик» наследует свойства фреймов «ребёнок» и «человек», которые находятся на более высоком уровне иерархии. Так, на вопрос «Любят ли ученики сладкое?» следует ответ «да», так как этим свойством обладают все дети, что указано во фрейме «ребёнок». Наследование свойств может быть частичным, так возраст для учеников не наследуется из фрейма «ребёнок», поскольку указан явно в своём собственном фрейме (Гаврилова 2000, с. 24–25).

2.3. Применение фреймов в системах автоматической обработки текстов

В системах АОТ фреймы строятся по традиционной схеме, в которой заранее заполненная топиковая (тематическая) строка сопровождаются пустым полем —

«дырой»-слотом. В него ЛА должен вставить обнаруженные им в тексте рематические комментарии. Задача алгоритмизатора состоит в том, чтобы поместить в базу знаний ЛА индикаторы, выявляющие с достаточно большой вероятностью и отправляющие в слот тот рематический фрагмент текста, который комментируют соответствующую им топиковую строку. Алгоритм должен достаточно полно учитывать вероятности коммуникативно-семантических связей между заранее заданными во фрейме топиками и попадающими в слоты текстовыми фрагментами или их переводами. В этом случае он не только даёт возможность автомату организовывать содержание текста. Он обеспечивает также пользователю определённый психологический комфорт при восприятии машинной аннотации, перевода и т.п. Уязвимой стороной фреймовой методики является то, что выбранный системой рематический фрагмент может попасть в «чужой» слот.

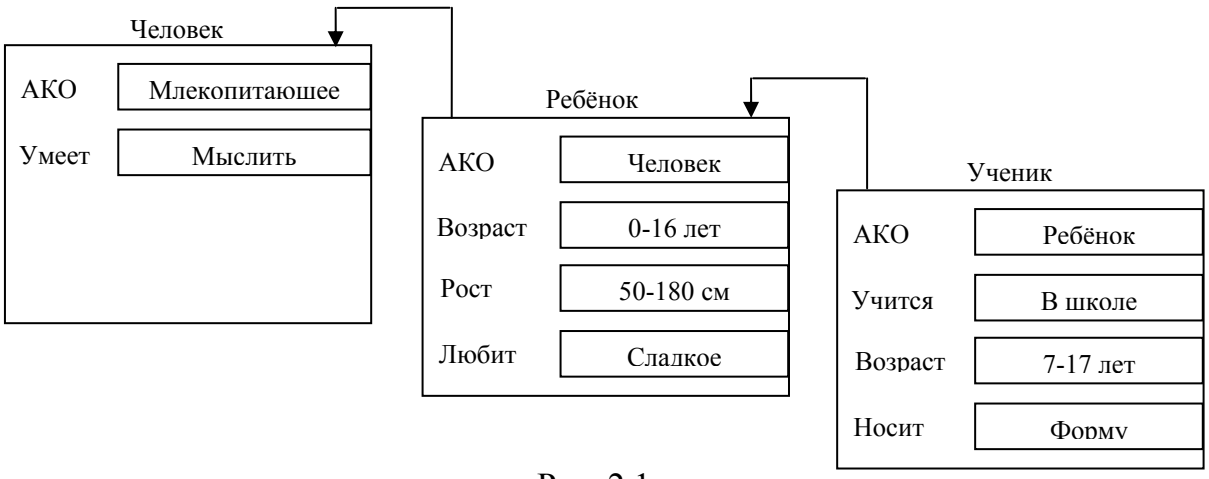

Рис. 2.1

К сожалению, уязвимой стороной «лобового» фреймового подхода является, с одной стороны, отсутствие модуля обратной связи, дающего возможность оценить правильность заполнения слотов теми или иными отрезками входного текста, а при фреймовом синтезе и их переводами (эту функцию выполняют интер- или постредактор, либо сам пользователь). С другой стороны, создание большого количества индивидуальных фреймов ставит разработчиков перед угрозой кризиса размерности. Поэтому целесообразно сочетать фреймовую методику с другими приемами анализа и синтеза текста (Пиотровский 2006).

Фреймы могут быть использованы для решения различных задач АОТ, к примеру, для сбора информации из массивов текстов с жёстко заданной структурой. Так множество сайтов выводят однотипную информацию (словарные статьи, новости и т.д.) из базы данных, оформляя её единообразно. Задачей аналитика в данном случае является определение маркеров — текста, тегов HTML, обрамляющих изменяемую информацию. Фреймы словарной статьи, новости и др., таким образом, задаются списком этих маркеров — заданных слотов, слоты-терминалы же извлекают нужную информацию (Луканин 2008а).

Для извлечения однотипных данных из слабо-структурированных текстов можно использовать фреймы-мишени. В тексте ищутся слова, цифры, которые должны заполнить все или большинство терминальных слотов фрейма, описывающего ситуацию, например, землетрясения. Такой фрейм может активироваться, если в тексте присутствуют слова или выражения *a powerful earthquake*, *the quake* и т.д. Этот фрейм содержит слоты даты, количества жертв, раненых, интенсивности землетрясения и его эпицентра (Saggion 2008).

Генерация текста также может быть основана на фреймовом подходе. Например, в системе генерации словоформ персидских глаголов Persian Verb Conjugator каждой форме глагола соответствует строго определённый фрейм, слоты которого либо остаются пустыми, либо заполняются псевдо-аффиксами, вычисляемыми для каждого глагола на основе фонологических правил. Псевдо-аффикс может отсутствовать, а значит и слот, соответствующий ему, будет пустым. К примеру, слот COMPOUND заполняется именем только тогда, когда введённый глагол является составным, образованным с помощью имени. Аналогично, слот VyV, соответствующий межвокальной йотированной согласной (персидскому йе), будет пустым для глагола *bar gozidan* (**بر گزیدن**) и заполнен для глагола *afzudan* (**افزودن**) в настоящем определённом времени в 1 лице ед. числа, имеющем форму *dāram miafzāyam* (**دارم می‌افزایم**) (табл. 2.2) (Луканин 2010).

Таблица 2.2

Фрейм настоящего определённого времени 1 л. ед.ч. с 2 примерами: *bar gozidan* (**بر گزیدن**) и *afzudan* (**افزودن**)

Слот 1	**Слот 2**	**Слот 3**	**Слот 4**	**Слот 5**	**Слот 6**	**Слот 7**	**Слот 8**
DAAR	**SG1**	**SPACE**	**COMPOUND**	**PREF_MI**	**PresStem**	**VyV**	**SG1**
dār	*am*	*SPACE*	*bar+SPACE*	*mi*	*gozin*		*am*
دار	م	*SPACE*	بر+*SPACE*	می+*ZWNJ*	گزین		م
dār	*am*	*SPACE*		*mi*	*afzā*	*y*	*am*
دار	م	*SPACE*		می+*ZWNJ*	افزا	ی	م

Вопросы к семинару

1. Фреймы в системах машинного перевода.
2. Фреймы и семантика понимания.
3. Концептуальный подход к обработке языка.
4. Теория концептуальной зависимости.
5. Элементы теория концептуальной зависимости: роли, правила, действия, состояния.

Литература к семинару

1. Пиотровский, Р.Г. Новые горизонты машинного перевода / Р.Г. Пиотровский // НТИ/2, 2002, №1.
2. Филлмор, Ч. Фреймы и семантика понимания / Ч. Филлмор // Новое в зарубежной лингвистике. Вып. XXIII. Когнитивные аспекты языка: Пер. с англ. — М.: Прогресс, 1988. — С. 52–93.
3. Шенк, Р. Обработка концептуальной информации / Р. Шенк. Пер. с англ. — М.: Энергия, 1980. — С. 7–68.

Глава 3. Подъязыки в системах машинного перевода

Эффективность работы современной системы МП в решающей степени зависит от её удачной настройки на конкретный подъязык (или микроподъязык) естественного языка, на определённую лексику и ограниченный набор грамматических средств, характерных для текстов данной предметной области, а также на определённые типы документов.

Учение о подъязыках с точки зрения машинного перевода было впервые сформулировано Н.Д.Андреевым (Ленинградский университет) в 1967, хотя представления о языковых регистрах, стилях, жанрах письменного текста и т.п. были хорошо известны и в традиционной лингвистике.

Подъязык, с точки зрения МП, определяется в первую очередь некоторым исходным набором текстов, в рамках которого определяется входной и выходной словари, степень распространения и характер лексической неоднозначности лексем, характер и распространённость синтаксических конструкций, способы их перевода в данной языковой паре и пр. Большую роль играют параллельные тексты и словари-конкордансы, с помощью которых можно достаточно эффективно изучить и использовать в составлении алгоритмов лексическую сочетаемость и дистрибуцию (распределение) языковых элементов в речи. Статистические характеристики подъязыков помогают упорядочить структуру соответствующих алгоритмов анализа и синтеза. Выходной словарь, ориентированный на потребности синтеза и передачи основных видов соответствий в конкретной языковой паре, обеспечивает приемлемый выходной текст.

В любом из современных видов машинного перевода необходимо участие человека-редактора, удобство работы которого обеспечивается качеством и надёжностью соответствующего программного обеспечения (Беляева 1996).

3.1. Подъязыки технической документации

Подъязыки (Controlled languages, CL) используются в системах технической документации в крупных корпорациях. Они улучшают удобочитаемость документов, т.к. требуют ясности и однозначности написанного, и улучшают переводимость текста, приводя его в соответствие с требованиями машинного перевода.

Результатом использования подъязыков является последовательность в стиле документов, возможность повторного использования текстов и соответствующее сокращение затрат на авторскую разработку (процесс авторского создания документов в системе управления документами) и перевод, не говоря уже о более высоком качестве документации и переводов.

Размер словаря подъязыка и количество значений слов ограничены, чтобы избежать лексической многозначности. В грамматике подъязыка (controlled grammar) указываются обычно следующие правила: эллиптические конструкции, согласованные предложения, синтаксическая многозначность сводятся к минимуму. Обычно в среду написания документации встраиваются программы проверки, редактор, переводческая память, машинный перевод, системы управления терминологией (terminology management systems) и средства автоматизированного перевода (computer-aided translation, CAT).

Одним из первых разработанных подъязыков МП был AECMA Simplified English, созданный в авиационной промышленности, чтобы справиться с постоянно растущим объёмом технической документации по самолетам. Его расширенная версия, Boeing Technical English (BTE), используется в компании Boeing для улучшения удобочитаемости и последовательности технической документации. Специальная программа контроля используется при написании текстов на BTE для поддержания единого стиля. Французская ассоциация аэрокосмической промышленности (GIFAS) разработала французский подъязык, который используется для перевода в подъязык Simplified English.

Компания Caterpillar Inc. (тяжелое машиностроение) использует подъязык Caterpillar Technical English (CTE) для всей технической документации на английском языке. CTE состоит из ограниченного терминологического словаря (controlled terminology inventory, более 70 000 терминов) и ограниченной грамматики.

В 1993 году General Motors начала проект CASL (Controlled Automotive Service Language) для технической документации. CASL — это подъязык английского языка, который содержит 62 грамматических правила и ограниченную терминологию (controlled terminology). Свои подъязыки используются также в шведской автомобильной компании Scannia, в немецкой компании BMW и др. (Torrejón 2002).

3.2. Предредактирование

Переводчикам необходимо адаптировать технические тексты к определённому подъязыку, т.е. выполнять их предредактирование. Для каждого подъязыка технической документации есть руководство по предредактированию, где описываются лексические и грамматические ограничения. К примеру, в подъязыке английского языка KANT описаны следующие лексические ограничения (кроме ограниченного словаря, который сокращает лексическую многозначность): поощряется использование функциональных слов, таких как артикли, в то время как не приветствуется использование местоимений и союзов, т.к. они повышают синтаксическую многозначность. Также необходимо сократить использование причастных форм (на -ing и -ed) после союзов (**While driving the vehicle*...) или в сокращённых определи-

тельных придаточных предложениях (**Directional stability caused by wheel lock-up…*).

Оба предложения должны быть переписаны следующим образом: *While you are driving the vehicle…* и *The directional stability that is caused by the wheel lock-up.*

Что касается грамматических ограничений, то их желательно использовать, даже если текст не предназначен для последующего перевода, т.к. они улучшают удобочитаемость и уменьшают многозначность исходных текстов. Различаются ограничения на уровне фраз и на уровне предложений. К первой категории относится замена фразовых глаголов однословными глаголами *(turn on* следует заменить на *start*) и повторяющиеся предлоги в соединительных конструкциях для избегания двойственности их прочтения.

К примеру, фраза *recorded memory of radio and each control unit* имеет 2 значения:

- {*recorded memory of radio*} *and* {*each control unit*},
- *recorded memory of {radio and each control unit}*.

Она должна быть записана следующим образом: *recorded memory of the radio and of each control unit*.

К уровню предложений относятся сочинение (2 части сложносочинённого предложения должны быть одного вида), определительные придаточные предложения (они всегда должны вводиться относительным местоимением) и эллиптические конструкции, которых нужно полностью избегать (Torrejón 2002).

Так при использовании программы ПроМТ рекомендуется избегать пропуска служебных слов. Пример:

Английское предложение «*Your e-mail address is the address other people use to send e-mail messages to you*» переведём на русский язык и получим не совсем понятный текст: «*Ваш адрес электронной почты — адрес другое использование людей, чтобы послать почтовые сообщения Вам.*» Теперь восстановим одно-единственное пропущенное слово — союз **that**: «*Your e-mail address is the address* ***that*** *other people use to send e-mail messages to you*» — и получим вполне корректный вариант: «*Ваш адрес электронной почты — адрес, который другие люди используют, чтобы послать почтовые сообщения Вам*» (Рекомендации по подготовке исходного текста).

В подъязыке Boeing Technical English (BTE) используются следующие правила написания:

а) определяющие слова (*the a, an, this, these и т.д.)* должны использоваться, когда это необходимо;
б) пассивные предложения должны избегаться в описаниях;
в) длина предложений должна быть не более 25 слов;
г) именные группы с более чем 3 словами должны избегаться;
д) не должно быть больше 2 прилагательных, определяющих существительное;
е) необходимо избегать форм на -ing;
ж) слово *that* нельзя опускать после глаголов;
з) относительные местоимения должны вводить определительные придаточные предложения;

и) приветствуется использование параллельных конструкций в сочинительных конструкциях.

Также существуют руководства, предоставляемые разработчиками систем МП для написания текстов, предназначенных для перевода. Некоторые из этих руководств имеют те же правила, что и руководства для подъязыков KANT Controlled English и BTE. Например, IBM рекомендует использовать короткие предложения (не более чем из 20 слов), избегать идиоматичных и жаргонных выражений, избегать многозначных слов, повторять существительные и именные группы вместо использования местоимений там, где это возможно, правильно расставлять знаки препинания, в списках использовать только полные предложения и т.д.

Некоторые правила для программы ПроМТ (Рекомендации по подготовке исходного текста):

1. Старайтесь использовать простые синтаксические конструкции с прямым порядком слов. Например, на первом месте должно идти подлежащее или его группа (*я, ты, он, мой кот, мой начальник, сын моей подруги*). На втором месте сказуемое, выраженное глаголом (*хочу, знаю, люблю*). Дальше идут обстоятельства, выраженные разными частями речи.

2. Используйте только общепринятые сокращения! Неправильный перевод сокращения — это только половина проблемы. Дело в том, что даже одно непереведённое слово может помешать программе перевода правильно проанализировать синтаксическую структуру предложения (а аббревиатуры участвуют в синтаксических связях наряду с обычными словами). Неприятные последствия может иметь тот факт, что некоторые аббревиатуры совпадают по написанию с часто используемыми словами. Например, *ПО* (программное обеспечение) пишется так же, как русский предлог *по* (регистр букв в данном случае не играет роли, так как ничто не запрещает нам написать предлог *по* большими буквами, например, в составе заголовка). Поэтому, увы, вполне закономерно выглядит перевод фразы «*Я часто использую это ПО*» как «*I frequently use it ON*». С другой стороны, если написать «*Я часто использую это программное обеспечение*» мы получим перевод «*I frequently use this software*».

3. Если перед русским существительным можно по смыслу поставить притяжательное местоимение (свой, мой, твой, его), то это нужно обязательно сделать. Особенно это касается одежды, частей речи, названий родственников. Не пишите *брат, платье, начальник*, пишите: *мой брат, её платье, наш начальник*.

3.3. Постредактирование

Т.к. адекватный МП производится на 100% только при соблюдении определённых требований (использовании ограниченного словаря и грамматики), любой другой МП требует постредактирования. В зависимости от целей постредактирование может быть как полным (когда перевод направлен на массового потребителя), так и частичным (перевод для понимания содержания). В зависимости от этих двух факторов различают несколько степеней постредактирования:

- МП без постредактирования для ознакомления с содержанием;

- быстрое постредактирование для быстроменяющейся информации и срочных текстов, допускающее исправление только самых грубых ошибок;
- частичное постредактирование, при котором постредактор решает, какое количество исправлений делать в зависимости от аудитории клиентов/читателей;
- полное постредактирование, когда МП используется в сочетании с подъязыком и выполняется в специализированных промышленных проектах быстрее ручного перевода.

Вопросы к семинару

1. Принципы работы памяти переводов.
2. Популярные программные системы памяти переводов.
3. Стандарты и форматы памяти переводов.

Глава 4. Теория конечных автоматов

В 1950-х гг. теория конечных автоматов (КА) стала продолжением модели алгоритмического вычисления Тьюринга. Машина Тьюринга (the Turing machine) — это гипотетический вычислитель, предложенный английским математиком Аланом Тьюрингом в 1936 г. (Turing 1937) как инструмент для изучения сложности алгоритмов. Целью Тьюринга было описание самой простой из возможных вычислительной машины в абстрактных математических терминах с тем, чтобы увидеть границу между тем, что вычислительная машина может делать, и тем, что ей не под силу. Одно из упрощений, которое он ввёл, было использование только двух символов, 0 и 1.

Он предположил, что абстрактная машина имеет неограниченную память, представленную в виде бесконечной ленты с ячейками, с которой могут быть считаны или на которую могут быть записаны символы 0 или 1. Считывание и запись производится на одной и той же ленте, но легче себе представить 3 отдельные ленты: одна — для считывания, вторая — для записи, а третья — для хранения (лента памяти).

Вычисления состоят из последовательности шагов, задаваемых программой блоку управления. Ячейка, находящаяся под считывающей головкой называется текущей. Каждый шаг может включать в себя считывание символа в текущей ячейке, запись в неё символа, возможное перемещение головки в соседнюю ячейку слева или справа, смену состояния и остановку (машина Тьюринга отличается от конечного автомата в основном возможностью записи символов на ленте). Ещё одним необходимым условием является наличие условной операции, зависимой от считанного символа. Вычисления начинаются в специальном состоянии, называемом стартовым, и заканчиваются в состоянии, называемом терминальным. Ис-

пользуя такую простую модель, можно выполнять самые различные вычисления. Кроме наличия бесконечной памяти, современные процессоры очень похожи на машину Тьюринга.

На базе работы Тьюринга Маккалох и Питтс (McCulloch 1943) создали автоматоподобную модель нейрона. Данная модель представляет собой бинарное устройство, которое в тот или иной момент времени может быть активным или нет. Оно принимало возбуждающие или подавляющие импульсы от других нейронов и срабатывало, если импульс превышал какой-то определённый порог. Основываясь на модели нейрона Маккалоха-Питтса (McCulloch-Pitts neuron) Клини (Kleene 1951; 1956) дал определение конечного автомата (finite automaton) и регулярных выражений (regular expressions), и доказал их эквивалентность. М.О. Рэбин и Д. Скотт (Rabin 1959) дали определение недетерминированных автоматов и доказали также их эквивалентность детерминированным автоматам.

4.1. Моделирование овечьего языка с помощью конечного автомата

Предположим, что овечий язык состоит из следующего (бесконечного) множества строк:

baa!
baaa!
baaaa!
baaaaa!
baaaaaa!
…

Регулярное выражение для этого языка на языке программирования Perl будет выглядеть следующим образом: baa+!, а конечный автомат так, как показано на рис. 4.1.

Конечный автомат (finite automaton, finite-state automaton или FSA) представляет собой направленный граф с конечным набором вершин (узлов) и набора дуг.

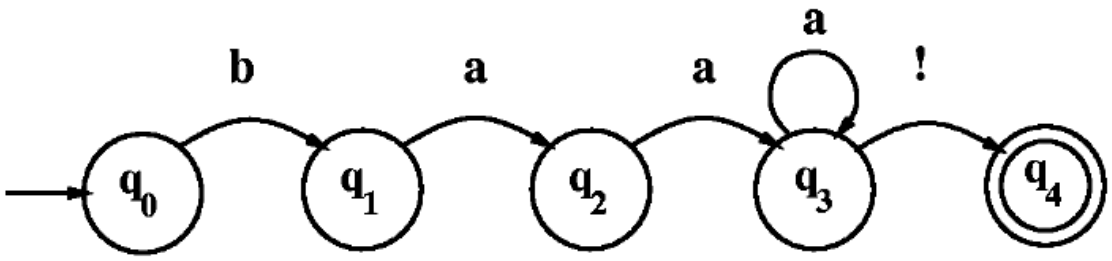

Рис. 4.1

Данный конечный автомат имеет 5 состояний q_0, q_1, ... q_4 и 5 переходов. Состояние 0 — начальное состояние, 4 — конечное, поглощающее состояние конечного автомата.

Конечный автомат может быть использован для распознавания строк символов следующим образом. Предположим, что на вход автомата поступает длинная лента, разбитая на ячейки с записанными в них символами (рис. 4.2).

Алгоритм конечного автомата:

- Работа автомата начинается в начальном состоянии (q_0).
- Проверяется следующий символ на входе. Если он совпадает с символом дуги, перейти к следующему состоянию и продвинуть ленту на 1 ячейку вперёд.
- Если автомат находится в конечном состоянии (q_4), когда больше нет символов на ленте, машина успешно распознаёт «слово» из овечьего языка.
- Если машина никогда не достигает конечного состояния или если на ленте заканчиваются символы или если появляется символ, не совпадающий с символом дуги (как в примере), или если машина остаётся в неконечном состоянии, это означает, что автомату не удаётся или он отказывается принять входной сигнал.

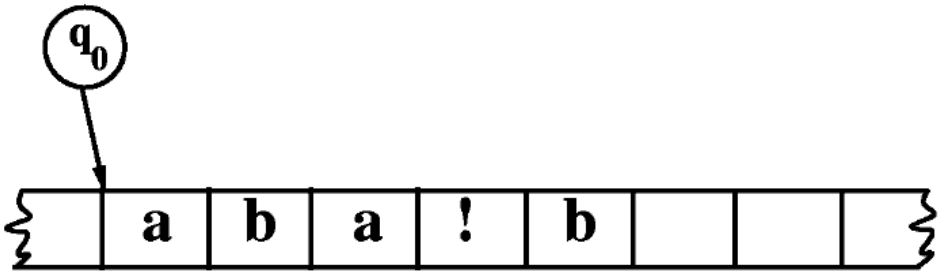

Рис. 4.2

Конечный автомат можно также представить в виде таблицы переходов состояний (state-transition table, табл. 4.1).

Таблица 4.1

	Вход		
Состояние	b	a	!
0	1	0	0
1	0	2	0
2	0	3	0
3	0	3	4
4:	0	0	0

Конечное состояние отмечается двоеточием (можно задать любое количество конечных состояний). `0` обозначает недопустимый или отсутствующий переход.

1-я строка таблицы читается следующим образом: «если мы находимся в состоянии 0 и видим на входе *b*, мы должны перейти в состояние 1. Если мы находимся в состоянии 0 и видим на входе *a* или *!*, распознавание завершается неудачно».

4.2. Формальные языки

Мы можем использовать конечный автомат и для генерации. В этом случае автомат, начиная с состояния q_0, переходит по дугам к новым состояниям, печатая символы этих дуг. Автомат прекращает вывод символов по достижении конечного состояния. Обратите внимание, что в состоянии 3 автомату предстоит выбор: напечатать *!* и перейти в состояние 4, или напечатать *a* и вернуться в состояние 3. Пока будем считать, что этот выбор производится случайным образом.

Формальный язык — это множество строк, состоящих из символов конечного множества символов, называемого алфавитом. Алфавит овечьего языка — это множество **Z = {a, b, !}**. Используя модель **m** (конкретный конечный автомат), мы можем определить формальный язык **L(m)**. Тогда формальным языком, определённый нашим конечным автоматом овечьего языка, будет следующее бесконечное множество:

L(m) = { baa! , baaa! , baaaa! , baaaaa! , baaaaaa! , … }

Полезность автомата для определения языка состоит в том, что бесконечное множество может быть закодировано в конечной форме. С помощью формальных языков можно моделировать естественный язык, например его фонологию, морфологию или синтаксис.

Для обозначения грамматики формального языка иногда используется термин *генеративная грамматика*. Происхождение этого термина связано с использованием автомата для определения языка путём генерации всех возможных строк данного языка.

Алфавит формального языка может состоять не только из букв. Алфавит более высокого уровня может состоять из слов. В этом случае можно создать конечные автоматы для моделирования сочетаемости слов.

К примеру, можно построить конечный автомат, который бы моделировал часть выражений английского языка, обозначающих количество денег: *ten cents*, *three dollars*, *one dollar thirty-five cents* и т.д.

Эту задачу можно разбить на две подзадачи. Сначала построим автомат для чисел от 1 до 99, который будет генерировать количество центов (рис. 4.3).

Затем можно добавить слова *cents* и *dollars* (рис. 4.4).

Далее можно расширить этот автомат за счёт добавления грамматики для различного количества долларов, включая слова *hundred* и *thousand*. Также необходимо, чтобы учитывалось число (единственное: *one cent, one dollar*, и множественное: *ten cents, two dollars*).

4.3. Недетерминированные конечные автоматы

Представьте теперь, что конечный автомат овечьего языка будет выглядеть так, как показано на рис. 4.5.

Единственное отличие заключается в том, что теперь петля расположена не в 3-м, а во 2-м состоянии. Если данный автомат будет использоваться для распознавания строк овечьего языка, то при переходе в состояние 2 у нас появляется выбор: либо перейти в состояние 3, либо остаться в состоянии 2. Автоматы с точками принятия решений называются недетерминированными конечными автоматами (НКА).

Существует ещё один общий тип НКА, в котором присутствуют дуги без символьных меток, называемые ε-переходами. Следующий автомат повторяет предыдущий, но использует ε-переход (рис. 4.6).

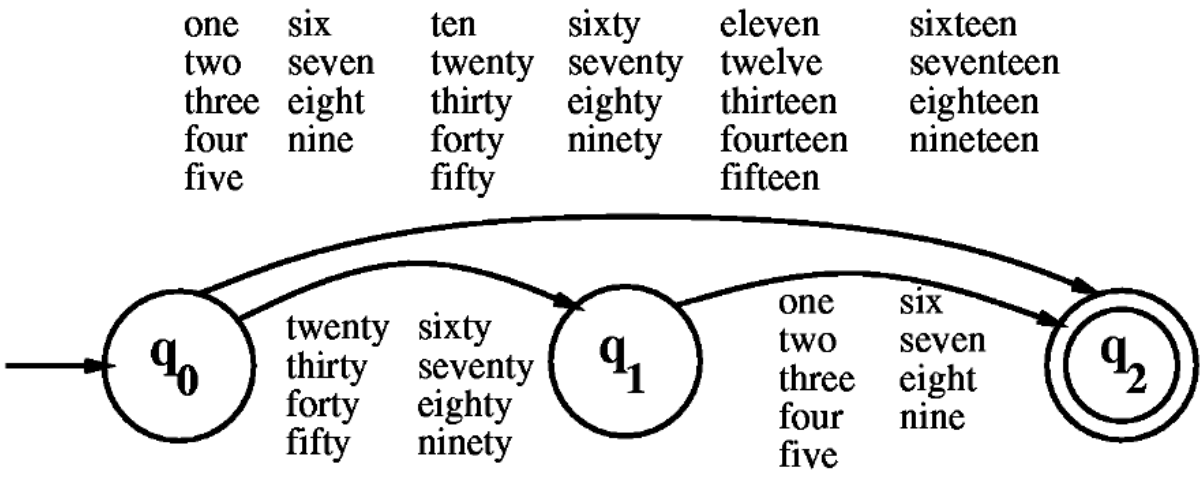

Рис. 4.3

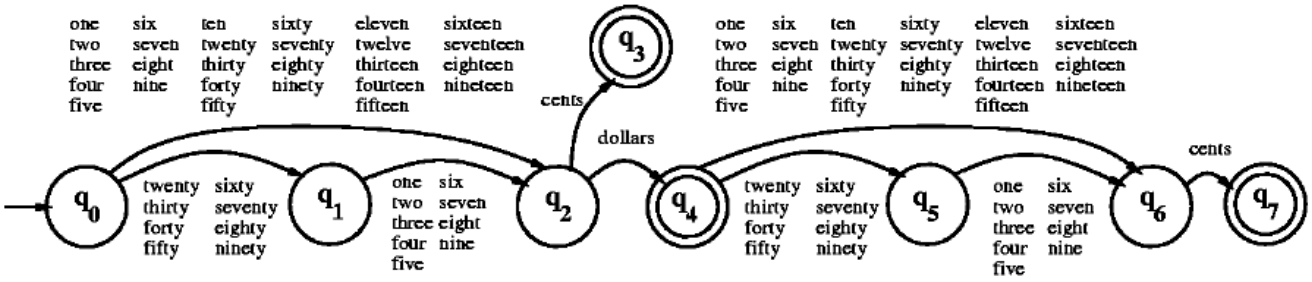

Рис. 4.4

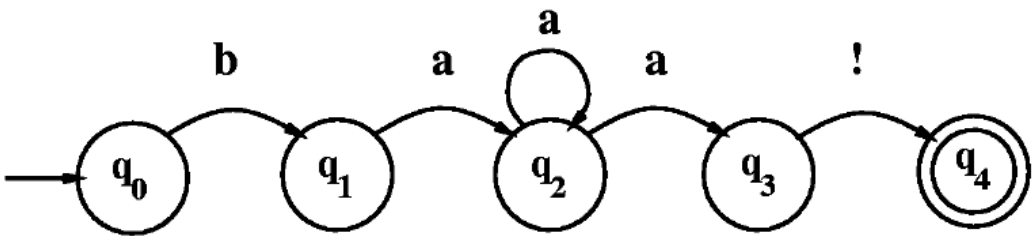

Рис. 4.5

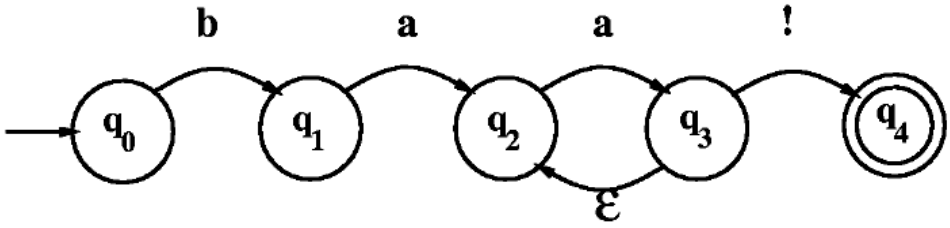

Рис. 4.6

Значение этой дуги следующее: если мы находимся в состоянии 3, то можем перейти к состоянию 2, не глядя на входной символ, либо перейти к следующему входному символу. Таким образом, мы можем не знать, какой переход выбрать: ε-переход или дугу *!*.

При использовании НКА для обработки строк возникает проблема выбора между несколькими альтернативами (дугами) — мы можем получить неправильный результат, если выберем неверную дугу. Эта проблема очень часто возникает при построении вычислительных моделей, в частности для парсинга (морфологического, синтаксического и др. разборов).

Существует 3 решения этой проблемы:

- Возврат: когда мы достигаем точки принятия решения, необходимо запомнить, в каком состоянии мы были, и какая часть входной информации была в это время разобрана. Если была выбрана неверная дуга, можно будет возвратиться к точке принятия решения и попробовать другой путь.
- Просмотр вперёд: мы можем взглянуть на следующую часть входной информации, чтобы выбрать, какой путь надо выбрать.
- Параллелизм: при достижении точки принятия решения можно проверять каждую альтернативу параллельно.

При 1-м подходе мы последовательно перебираем все пути, пока автомат не примет строку. Если нельзя сделать следующий переход (символ дуги отличается от входного символа) или мы достигаем конечного состояния, когда строка ещё не полностью обработана, мы возвращаемся к запомненному состоянию и пробуем другой путь.

Необходимо внести изменения в таблицу переходов состояний для недетерминированного конечного автомата (табл. 4.2). Для автоматов, имеющих ε-переходы, необходимо добавить соответствующую колонку ε. Также для нескольких переходов из одного узла в таблице можно записывать список узлов, в которые можно перейти из этого узла.

Таблица 4.2

	Вход			
Состояние	b	a	!	ε
0	1	0	0	0
1	0	2	0	0
2	0	2,3	0	0
3	0	0	4	0
4:	0	0	0	0

4.4. Распознавание как поиск

Алгоритмы систематичного поиска решений (последовательного перебора альтернатив) называются алгоритмами **поиска в пространстве состояний**. В таких алгоритмах создаётся пространство возможных решений. Целью такого алгоритма является исследовать это пространство, чтобы вернуть ответ, если решение найдено, или отвергнуть входную строку, если решение не найдено после просмотра всего пространства.

Эффективность таких программ часто зависит от порядка, в котором просматриваются состояния в пространстве. Неправильный выбор порядка перебора состояний может привести к слишком долгому процессу поиска. К сожалению, практически невозможно сказать, какой порядок будет более эффективным и часто лучшее, что можно сделать — это проверить все возможные варианты.

Список состояний, к которым необходимо вернуться, реализуется в виде стека. Такой подход обычно называют **поиском в глубину** или стратегией **LIFO** (Last In First Out, последним пришёл — первым обслужен).

У стратегии поиска в глубину есть один недостаток: при некоторых условиях перебор может войти в бесконечный цикл. Это может произойти, если пространство поиска организовано так, что в некоторое предыдущее состояние можно вернуться вновь, либо если существует бесконечное число состояний.

Вторым решением является организация состояний в пространстве поиска таким образом, чтобы они проверялись в порядке их появления, т.е. при встрече новой альтернативы, записывать её в конец плана обработки состояний. Такой поиск организуется в виде очереди и называется **поиском в ширину** или стратегией **FIFO** (First In First Out, первым пришёл — первым обслужен).

Как и при поиске в глубину, поиск в ширину имеет свои недостатки. Поиск может никогда не закончиться, если пространство состояний бесконечно. Более важно то, что даже при сравнительно большом пространстве поиска, размер стека может сильно увеличиться, что соответственно потребует большого количества памяти.

При одинаковой эффективности обоих подходов обычно выбирают поиск в глубину, т.к. он более эффективен с точки зрения использования памяти.

4.5. Построение лексикона на основе конечного автомата

Существует множество способов моделирования морфотактики, наиболее распространённым является использование конечного автомата. Самая простая модель конечного автомата для английского словоизменения существительных может выглядеть следующим образом (рис. 4.7):

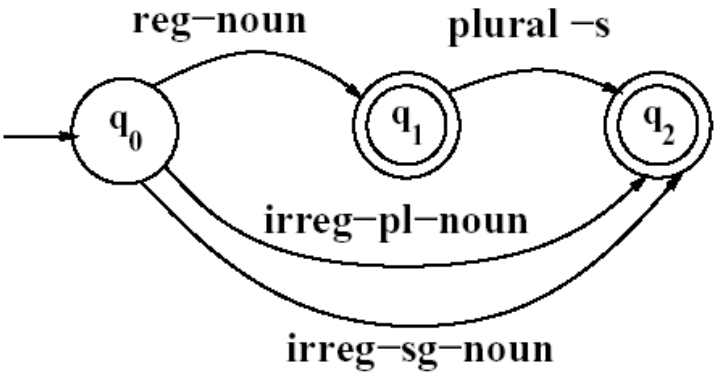

Рис. 4.7

В данном автомате предполагается, что лексикон включает правильные существительные (**reg-noun**), которые образуют мн. ч. при помощи окончания *-s* (например, *cat*, *dog*, *fox*, *aardvark*). Таких большинство (пока мы не берём во внимание случаи фонетическое изменения окончания в таких словах как *fox* : *foxes*. Лексикон также включает неправильные формы существительных, которые образуют формы единственного (**irreg-sg-noun:** *goose, mouse*) и множественного числа (**irreg-pl-noun:** *geese, mice*) по-другому.

Модель словоизменения английских глаголов может выглядеть следующим образом (рис. 4.8).

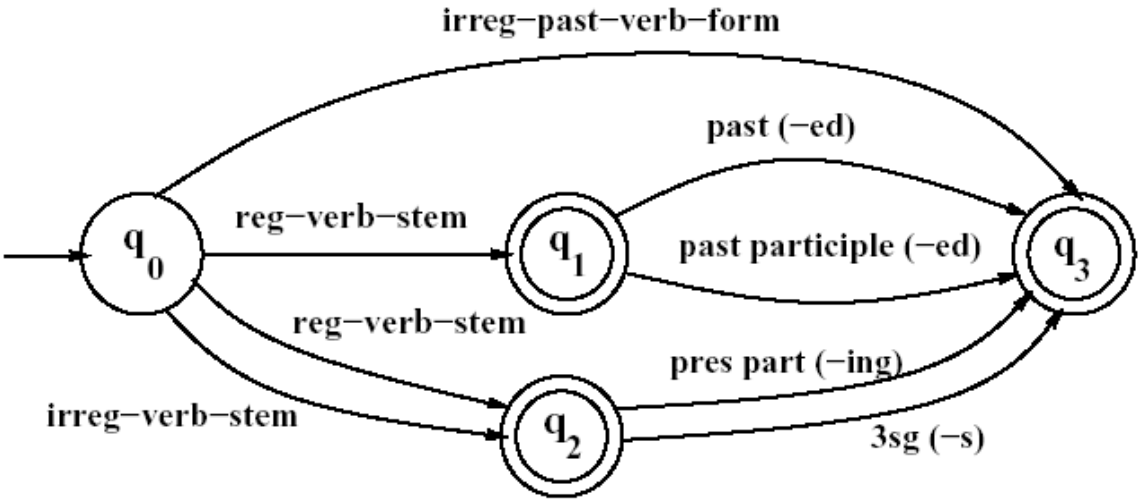

Рис. 4.8

Этот лексикон имеет 3 класса основ (**reg-verb-stem**, **irreg-verb-stem** и **irreg-past-verb-form**) и 4 класса аффиксов (*-ed* прошедшего времени, *-ed* причастия прошедшего времени, *-ing* причастия настоящего времени и *-s* 3-го лица ед. ч.) (табл. 4.3).

Таблица 4.3

reg-verb-stem	**irreg-verb-stem**	**irreg-past-verb**	**past**	**past-part**	**pres-part**	**3sg**
work fry call print	write fly buy		-ed	-ed	-ing	-s

Английская **деривационная морфология** намного сложнее словоизменительной, и автомат для её моделирования должен быть соответственно довольно сложным. В качестве примера приведём часть морфотактики английских прилагательных. Возьмём следующие формы прилагательных:

big, bigger, biggest
cool, cooler, coolest, coolly
happy, happier, happiest, happily
unhappy, unhappier, unhappiest, unhappily
clear, clearer, clearest, clearly, unclear, unclearly
red, redder, reddest
real, unreal, really

Первой гипотезой может быть то, что прилагательные могут иметь необязательную приставку (*un-*), обязательный корень (*big, cool* и т.д.) и необязательный суффикс (*-er, -est* или *-ly*). Таким образом, можно предложить следующий конечный автомат (рис. 4.9).

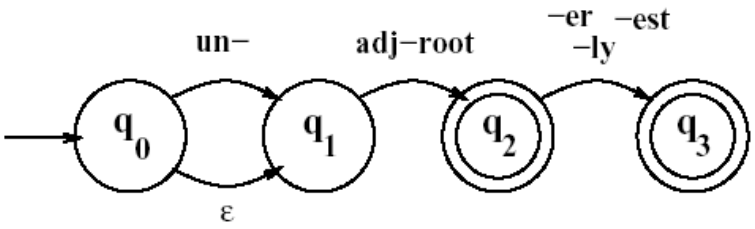

Рис. 4.9

Увы, в то время как данный конечный автомат распознаёт все прилагательные из приведённого списка, он также распознаёт такие неграмматичные формы как *unbig*, *redly* и *realest*.

Необходимо создать классы корней и указать, какие суффиксы они могут присоединять. Пусть класс **adj-root$_1$** включает прилагательные, которые могут присоединять аффиксы *un-* и *-ly* (*clear*, *happy* и *real*), в то время как класс **adj-root$_2$** включает остальные (*big*, *cool* и *red*).

Е.Л. Антуорт (Antworth 1990) предлагает другую схему для решения этой проблемы (рис. 4.10).

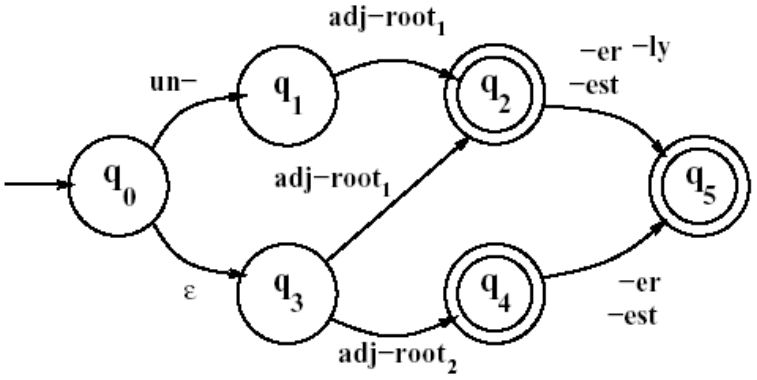

Рис. 4.10

В качестве другого примера, показывающего сложность английского словообразования, приведём фрагмент конечного автомата для английской деривационной морфологии имён и глаголов, основанной на работах Р. Спроута (Sproat 1993), Л. Бауэра (Bauer 1983) и М.Ф. Портера (Porter 1980) (рис. 4.11).

Данный конечный автомат моделирует некоторые деривационные факты, такие как **обобщение**, когда после любого глагола, заканчивающегося на *-ize* может следовать суффикс существительного *-ation* (Bauer 1983; Sproat 1993). Так, если есть слово *organize*, мы можем предсказать слово *organization,* проходя состояния q_0, q_1 и q_2.

Аналогично, прилагательные на *-al* или *-able* в состоянии q_5 (*equal*, *formal*, *realizable*) могут присоединять суффикс *-ity*, или иногда суффикс *-ness* при переходе в состояние q_6 (*naturalness*, *casualness*).

Теперь можно использовать эти конечные автоматы для решения проблемы **морфологического распознавания**, т.е. определения, является ли входная строка

правильным английским словом или нет. Это делается путём подключения к автомату морфотактики каждого подлексикона, т.е. расширения каждой дуги (например, дуги **reg-noun-stem**) за счёт всех морфем, которые сочетаются с данным классом. Полученный автомат затем может быть представлен на уровне букв.

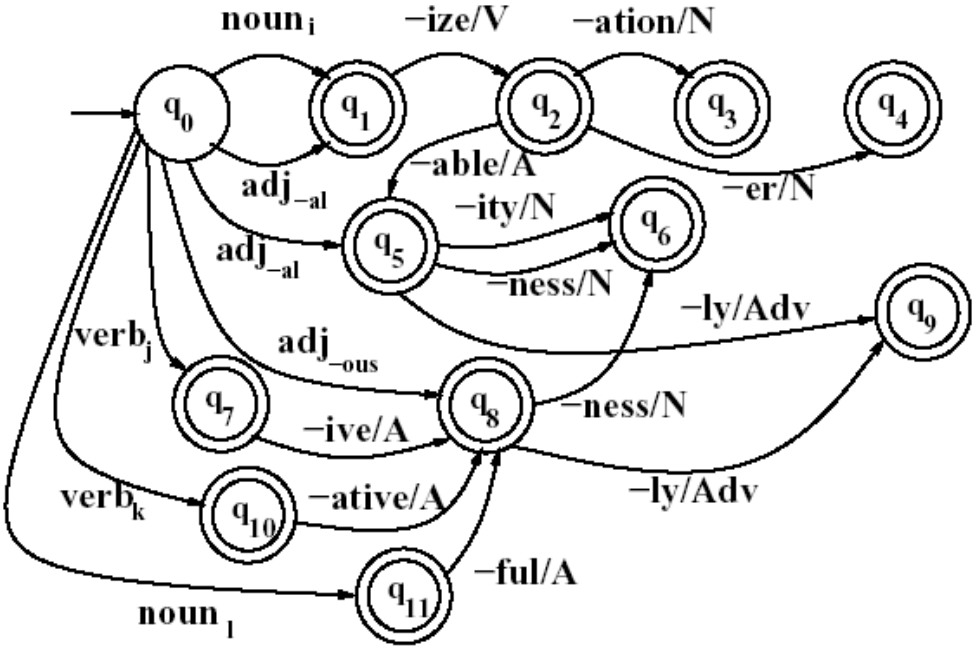

Рис. 4.11

Упражнения:

1. Дополните конечный автомат для английских денежных выражений (рис. 4.12) так, чтобы он работал с выражениями до $100,000, и чтобы слова *cent* и *dollar* стояли в нужном числе.

2. Разработайте конечный автомат, который бы распознавал английские даты, такие как *March 15*, *the 22nd of November*, *Christmas*. Постарайтесь включить все абсолютные даты (т.е. не дейктические, относящиеся к текущему дню, например, *the day before yesterday*). Над каждой дугой должно быть записано слово или набор слов. Используйте сокращения для классов слов, чтобы избежать слишком большого количества дуг (например, *Furniture → desk, chair, table*).

3. Расширьте последний автомат для обработки дейктических выражений таких как *yesterday*, *tomorrow*, *a week from tomorrow*, *the day before yesterday*, *Sunday*, *next Monday*, *three weeks from Saturday*.

4. Разработайте автомат для обработки русских выражений времени таких как *одиннадцать часов, пятнадцать двадцать, полночь, четверть первого, без пяти девять* и т.д.

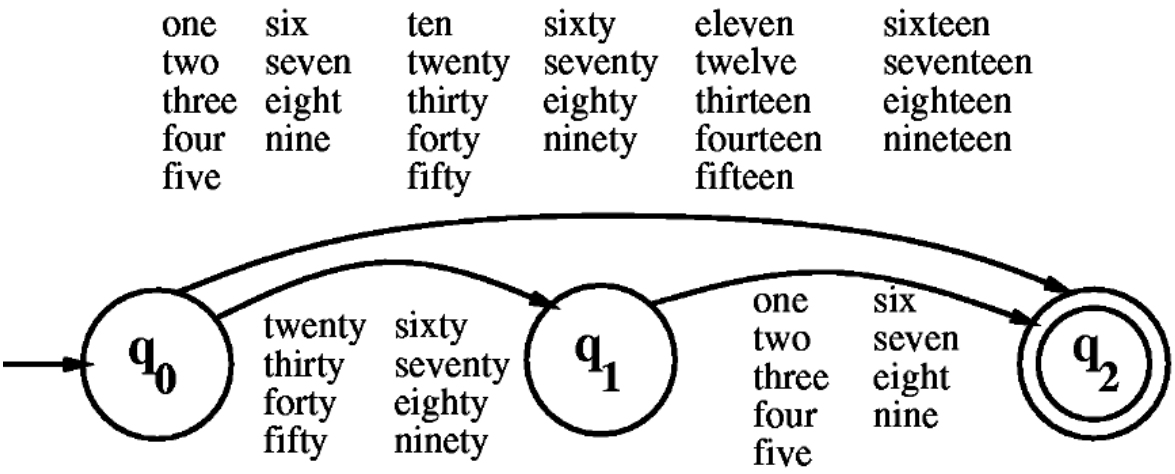

Рис. 4.12

Вопросы к семинару

1. Устройство конечного преобразователя (КП).
2. КП для морфологической обработки.
3. Орфографические правила в КП.
4. Лексикон и правила в КП, стеммер Портера.
5. Выделение слов и предложений (токенизация).
6. Коррекция орфографии на примере алгоритма минимального расстояния.

Литература к семинару

1. Jurafsky, D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition // D. Jurafsky, J.H. Martin. — New Jersey: Prentice Hall, 2000. — Ch. 3.4–3.12.

Глава 5. Системы аннотирования и реферирования

Одной из задач АОТ является автоматизация процессов аналитико-синтетической обработки информации (компьютерное свёртывание / развертывание), к числу которых относятся индексирование, аннотирование, реферирование, конспектирование, фрагментирование, перевод и другие формы информационного анализа и синтеза. Под свёртыванием принято понимать сжатие, или компрессию, текста первичного документа при его переработке в текст вторичного документа (Богданов 1996, с. 389).

«Аннотацией называется связный текст, который кратко выражает центральную тему или предмет какого-то документа (объём не превышает 150–200 знаков)». «Рефератом называется связный текст, который кратко выражает не только центральную тему или предмет какого-то документа, но и цель, применяемые ме-

тоды и основные результаты описанного исследования или разработки (средний объём в пределах 1000–1500 знаков)» (Михайлов 1970, с. 436).

Различию между аннотацией и рефератом соответствует в англоязычных странах различие между указательным рефератом (indicative abstract) и информативным рефератом (informative abstract). Аннотация и реферат различаются также объёмом, который напрямую зависит от выполняемых ими функций. Объём аннотации или реферата должен составлять от 5 до 30% исходного текста. Подготовка аннотаций нескольких источников информации или формирование сводок для карманных устройств предполагает ещё большую степень сжатия.

5.1. Подходы и методы автоматического реферирования

История применения вычислительной техники для реферирования насчитывает уже более сорока лет и связана с именами таких исследователей, как Г.П. Лун, Г. Эдмундсон, В.Е. Берзон, И.П. Севбо, Э.Ф. Скороходько, Д.Г. Лахути, В.П. Леонов, Р.Г. Пиотровский и многие другие. За эти годы были выработаны многочисленные подходы к решению данной проблемы, которые достаточно чётко подразделяются на два направления: квазиреферирование и собственно автоматическое реферирование. Квазиреферирование основано на экстракции из первичных документов с помощью определённых формальных признаков «наиболее информативных» фраз (фрагментов), совокупность которых образует некоторый экстракт (квазиреферат). Собственно автоматическое реферирование же основано на выделении из текстов с помощью специальных информационных языков наиболее существенной информации и порождении новых текстов (рефератов), в большей или меньшей степени изоморфных первичным документам (или их частям).

Квазиреферирование обладает той особенностью по сравнению с собственно реферированием, что основывается на анализе поверхностно-синтаксических отношений в тексте, которые выражены в нём и не требуют обращения к глубинно-семантическим процессам, изученность которых ещё явно недостаточна для описания свойств любого текста. Второе направление в настоящее время представлено экспериментальными исследованиями и до широкой реализации ещё не дошло.

5.2. Экстрактивные методы

Экстрактивный метод предполагает акцент на выделение характерных фрагментов (как правило, предложений). Для этого методом сопоставления фразовых шаблонов, выделяются блоки наибольшей лексической и статистической релевантности. Создание итогового документа в данном случае — это соединение выбранных фрагментов.

В большинстве методов применяется модель линейных весовых коэффициентов. Основу аналитического этапа в этой модели составляет процедура назначения весовых коэффициентов для каждого блока текста в соответствии с такими характеристиками, как расположение этого блока в оригинале, частота появления в тексте, частота использования в ключевых предложениях, а также показатели статистической значимости. Сумма индивидуальных весов, как правило, опреде-

лённая после дополнительной модификации в соответствии со специальными параметрами настройки, связанными с каждым весом, даёт общий вес всего блока текста U:

$$\text{Weight}(U) = \text{Location}(U) + \text{CuePhrase}(U) + \text{StatTerm}(U) + \text{AddTerm}(U)$$

Весовой коэффициент расположения (Location) в данной модели зависит от того, где во всём тексте или в отдельно взятом параграфе появляется данный фрагмент — в начале, в середине или в конце, а также используется ли он в ключевых разделах, например, вводной части или в заключении.

Ключевые фразы представляют собой лексические или фразовые резюмирующие конструкции, такие как «в заключение», «в данной статье», «согласно результатам анализа» и так далее.

Кроме того, при назначении весовых коэффициентов в этой модели учитывается показатель статистической важности (StatTerm). Статистическая важность вычисляется на основании данных, полученных в результате анализа автоматической индексации, при котором исследователи выявляют и оценивают целый ряд метрик, определяющих весовые коэффициенты термина. Эти метрики позволяют выделить документ из числа других в определённом наборе документов.

Одна группа метрик, например, метрика tf.idf, характеризует баланс между частотой появления термина в документе и частотой его появления в наборе документов (как правило, используется с другими метриками частоты и средствами нормализации длины) (Jurafsky 2000, p. 653).

И, наконец, эта модель предполагает просмотр терминов в блоке текста и определение его весового коэффициента в соответствии с дополнительным наличием терминов (AddTerm) — появляются ли они также в заголовке, в колонтитуле, первом параграфе и в тексте пользовательского запроса. Выделение приоритетных терминов, наиболее точно отражающих интересы пользователя, — это один из путей настроить реферат или аннотацию на конкретного человека или группу.

На аналитическом этапе применяется модель линейных весовых коэффициентов, предполагающая выполнение последовательности вычислений частоты и операций сопоставления строк или шаблонов, которые для каждого блока исходного текста выдают весовые коэффициенты четырех типов (Location, CuePhrase, StatTerm, AddTerm). Затем эти коэффициенты суммируются для каждого блока, после чего выбираются *n* блоков, обладающих наивысшей суммой коэффициентов (значение *n* может быть определено на основании степени сжатия) для включения в реферат.

Этот метод был создан ещё в 60–70-х годах, но большинство систем, подготавливающих такого рода конспект на основе выдержек, до сих пор используют подход, проиллюстрированный на рис. 5.1. Анализ сравнительных характеристик различных моделей, произведённый с целью определить производительность каждой, показал, что локализацию блоков текста можно считать одной из самых полезных функций, особенно в сочетании с функцией выявления ключевых фраз.

Конечно, главное преимущество линейной модели заключается в простоте её реализации. Однако выделение предложений (или параграфов), не учитывающее

взаимоотношений между ними, приводит к формированию бессвязных рефератов. Некоторые предложения могут оказаться пропущены, либо в них могут встречаться «висящие» слова или словосочетания (слово или фраза, которые невозможно понять без другого слова или фразы).

«Билл Диксон поступил на работу в Procter & Gamble в 1994 году. В 1996 году он стал её вице-президентом».

Если в реферате первая фраза будет потеряна, текст потеряет свою информативность.

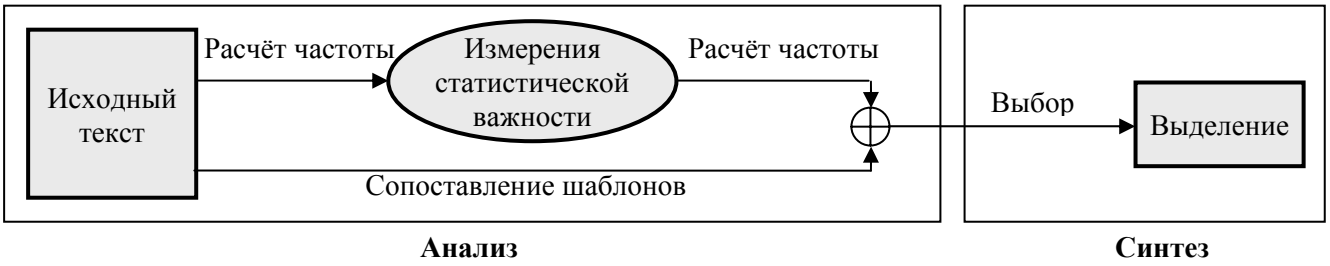

Рис. 5.1. Обобщённая архитектура реферирования без опоры на знания

Есть множество работ, в которых делаются попытки решить эту проблему, в основном за счёт разного рода «заплаток». В ряде подходов создаётся специальное окно для предыдущего предложения реферата, с помощью которого можно определить наличие смыслового разрыва или «висящего» слова. В других случаях предложения, содержащие «висящие» слова, исключаются из реферата. При таком подходе степень сжатия уменьшается, т.к. в реферат привносится посторонняя информация. Кроме того, когда основной реферат уже сформирован, трудно восстановить исходный процент сжатия.

5.2.1. Статистические методы

Метод Луна (Luhn). Метод, предложенный в 1958 г. Г.П. Луном, считается классическим. Он предположил, что (1) часто встречающиеся в тексте слова являются наиболее значимыми и (2) чем больше часто встречающихся слов оказывается рядом, тем более существенную информацию содержит предложение.

Главный недостаток такого подхода: при использовании только частотного критерия полностью игнорируются смысловые связи между словами. При таком методе реферирования объём и качество рефератов полностью зависят от статистики текста, а предложения, содержащие наиболее важную информацию, могут вообще быть не выделены. Однако эти недостатки в известной степени компенсируются простотой анализа и однородностью получаемых рефератов.

Метод ACSI-Matic (Assistant Chief of Staff for Intelligence, Automatic). В исследовании ACSI-Matic, статистический метод Луна был незначительно модифицирован. Метод ACSI-Matic можно проиллюстрировать следующим примером. Пусть некоторое предложение имеет следующий вид:

N R N N R N N N R N N,

где *N* — несущественные слова (*N*-слова), *R* — слова, подлежащие включению в словарь *(R* — representative). *R*-слова оцениваются в 1 балл, а группы *N*-слов, расположенные между ними, — в 1/2 балла; *n* — количество *N*-слов между двумя *R*-словами.

Общий вес такого предложения составляет: 1 + 1/4 + 1 + 1/8 + 1= 3 + 3/8

Данный подход основывался на допущении, что представительными являются слова, частота встречаемости которых превосходит среднюю частоту слов в документе.

Затем для определения объёма реферата общее количество отобранных предложений (если их не более 200) делилось на 10, и объём реферата составлял 10% от общего объёма документа. Предложения с большими весами подлежали включению в реферат, а со средним — помечались как «резервные». Нижний предел веса устанавливался экспериментальным путём.

При изучении избыточности информации в предложениях вводился следующий критерий; если число встретившихся в двух предложениях синонимов и близких по значению слов превышало 25% от общего количества слов в предложении, то такие предложения считались избыточными и вычеркивались. В этом случае для реферата выбирались предложения из резерва. Этот процесс длился до тех пор, пока не устранялись избыточные или не заканчивались «резервные» предложения. Окончательный набор и представлял собой реферат.

Однако метод ACSI-Matic, несмотря на усовершенствованную процедуру отбора предложений, игнорировал смысловые связи между словами.

Метод Освальда. Отличие этого метода проявлялось в том, что В. Освальд и его сотрудники применили к отбору предложений для включения в реферат инструмент, применяемый при индексировании, а именно: предложения анализируемого текста сопоставлялись с заданными перечнями слов (или с группами слов, т.е. со словами, объединёнными по смыслу). Для включения в реферат отбирались предложения с наибольшим количеством совпавших с перечнем слов, а отобранные предложения далее обрабатывались по методу Г. Луна. Метод В. Освальда предусматривал следующие процедуры обработки документов:

- выявление слов, отражающих содержание документа;
- выявление «мультитермов», т.е. устойчивых словосочетаний, при условии, что они встречаются в тексте чаще одного раза;
- выявление предложений с двумя и более мультитермами;
- расположение мультитермов в порядке частоты встречаемости;
- отбор предложений с наибольшим количеством основных слов и мультитермов.

Методы статистических ассоциаций. Идея В. Освальда о необходимости включать «мультитермы» была развита Л. Дойлом и М. Квиллианом, которые применили для отражения содержания документов ассоциативные методы создания семантически связанных групп (пучков) терминов (word clusters).

В качестве основания для ассоциирования терминов было предложено использовать их совместную встречаемость в предложениях. На основании совместной встречаемости слов рассчитывался коэффициент подобия. Такой метод позволил использовать для отражения содержания не только основные термины, но и другие элементы текста.

Подход Л. Дойла базировался на статистических критериях построения ассоциативных карт с помощью ЭВМ. Созданию карты предшествовало создание исходной матрицы (ключевые слова текста) и матрицы взаимосвязанных элементов. Построив такую матрицу и используя коэффициент корреляции Пирсона, Л. Дойл вычислил коэффициенты подобия между терминами отдельных документов. В результате была получена ассоциативная карта. Составленный таким образом реферат представлял собой реферат телеграфного стиля — реферат, содержание которого представлено в виде коротких назывных предложений или набора ключевых слов (дескрипторов). Такие рефераты используются для констатации наличия определённых сведений в анализируемом документе.

Метод другого исследователя, М. Квиллиана, несколько отличается от метода Л. Дойла. В противоположность статистическому методу, М. Квиллиан предложил использовать не ассоциативные карты, составленные на основе статистических показателей, а семантические карты. При этом учитывались взаимосвязи не между отдельными словами, а между информативными концептами.

5.2.2. Логико-математические методы

Методы Эдмундсона-Виллиса (Edmundson-Wyllys). В 1961–1963 гг. Г. Эдмундсон и Р. Виллис критически рассмотрели существующие способы оценки качества рефератов: интуитивный метод; метод построения модели «идеального» реферата; «анкетный» метод; оценку поисковых возможностей реферата; метод статистической корреляции, применяемый при автоматической обработке текста.

Дальнейшее развитие этих методов заключалось в обосновании логико-математической концепции отбора предложений. В соответствии с данной концепцией, Г. Эдмундсон и Р. Виллис предложили четыре взаимодополняющих метода, которые они назвали «Методом намёка» (Cue method), «Методом ключевых слов» (Key method), «Методом заглавия» (Title method) и «Методом определения места» (Location method).

«Метод намёка» основан на использовании при отборе предложений списка слов, в котором выделены так называемые «премиальные» слова (bonus words), имеющие положительный вес, «клеймёные» слова (stigma words) с отрицательным весом и «нулевые» слова (null words), которые не учитываются при отборе предложений.

При «Методе ключевых слов» учитывалась частота встречаемости слов в тексте аналогично методу Г. Луна.

В «Методе заглавия» основную роль играет словарь терминов, встречающихся в заглавиях и подзаголовках (за исключением нулевых слов). Слова, содержащиеся в предложениях, в которых имеются термины, одинаковые с заглавием, полу-

чают более высокую оценку, чем те же слова из подзаголовка или предложения без таких терминов.

«Метод определения места» (местонахождения) основан на гипотезе, что наиболее существенная информация в документах концентрируется в самом начале или в конце определённого отрывка или параграфа текста.

Способы отбора предложений оценивались по формуле:

$$a_1C + a_2K + a_3T + a_4L,$$

где $a_1,..., a_4$ — положительные веса терминов, отобранных каждым методом. Полученные результаты, свидетельствуют о том, что используя метод заглавия, можно отразить содержание первоисточника на 30–40%; используя метод ключевых слов — на 15–40%. Наиболее эффективным оказалось совместное использование методов намёка, заглавия и местонахождения (30–60%). Исходя из этих результатов, Г. Эдмундсон и Р. Виллис обосновали отказ от применения метода ключевых слов для отбора предложений в процессе реферирования. Результаты описанных исследований представляют интерес по ряду причин. В частности, стало очевидно, что за успехами предыдущих экспериментов не последовали ожидаемые практические результаты, поскольку эти эксперименты проводились на ограниченных массивах специально подобранных текстов. Программы, рассчитанные на подобные тексты, заранее гарантировали успех. Авторы особо обратили внимание на необходимость дальнейших исследований по синтаксическому и семантическому анализу текстов.

5.2.3. Дистрибутивный метод

Метод Эрла (Earl). Эксперименты в 1965–1971 гг. под руководством Л. Эрла. В ходе этих исследований особое внимание обращалось на лингвистические проблемы обработки документальной информации, в частности, на роль морфологической, фонетической и синтаксической структуры английского языка. Кроме Л. Эрла, в состав группы входили ученые-лингвисты: Г. Резников (Resnikoff), Дж. Долби (Dolby), Г. Робинсон (Robinson), Б. Рудин (Rudin), О. Фиршейн (Firshein) и М. Фишлер (Fischler).

Если в ходе эксперимента группа предложений, отобранных для анализа, признавалась потенциальным источником индексируемых терминов (indexable), то синтаксические структуры этих предложений тщательно изучались. Исследование должно было подтвердить или опровергнуть сходство полученных синтаксических структур с потенциально пригодными для индексирования синтаксическими структурами предложений из заранее составленного списка. Полученные структуры, помеченные index или nonindex, входили в словарь частей речи. Однако первые эксперименты показали, что, основываясь только на анализе синтаксических структур, нельзя выделить предложения для реферата и индексирования.

5.2.4. Метод содержательных аспектов

Метод Раша. Метод автоматического реферирования документов (Automatic Document Abstracting Method — ADAM), разработанный группой Дж. Раша и реализованный на ЭВМ IBM-370, предусматривал выполнение следующих условий:

- объём машинного реферата не должен превышать 10% объёма первичного документа;
- в реферате используется та же лексика, что и в первичном документе;
- количественные показатели опускаются (кроме тех, которые являются результатом исследования);
- предварительные и негативные замечания, уравнения, сокращения, объяснения, цитаты, таблицы, чертежи, ссылки и т.п. не подлежат включению в реферат.

Включению в реферат подлежали предложения, содержащие информацию о цели и результатах исследования и о выводах автора. Следовательно, основная задача заключалась в том, чтобы выявить такие предложения.

Проанализировав существующие методики выбора предложений, Дж. Раш, Р. Сальвадор и А. Замора предложили собственный метод, названный ими «Методом отказа от предложений» (method for rejecting sentences).

В известном смысле метод Раша можно считать развитием «метода намёка» Эдмундсона-Виллиса: так же задаётся список лексических единиц, чьё присутствие в предложении свидетельствует о необходимости включить это предложение в реферат (или наоборот). Каждому такому слову, в зависимости от его расположения в тексте документа, присваивался определённый «семантический вес». Например, предполагалось что предложения, начинавшиеся словами *Our work, This paper, Present research* сообщали о цели исследований; предложения, начинавшиеся словом *some* и артиклем *a,* — подробности описания. Исходя из этого, данным лексическим единицам присваивается положительный или отрицательный семантический вес.

Кроме семантического веса лексических единиц, устанавливалась также синтаксическая ценность отдельных элементов предложения. Индикатором при выделении таких элементов служили знаки препинания внутри предложений (запятые).

Вход в словарь — Word Control List — представлял собой алфавитный перечень слов и фраз, снабжённых ассоциативным кодом. Ассоциативные коды отражали семантический вес и синтаксическую ценность предложения, подлежащего отбору или вычёркиванию.

Такое построение входа позволяло вводить документы различной тематики и получать рефераты, отражающие различные аспекты содержания. Длина выполненных таким образом рефератов не превышала 35% длины оригинала.

5.2.5. Метод текстовых связей

Метод Берзона. Под влиянием когнитивной психологии постулировалось, что информационная значимость предложения в тексте может определяться не только статистическими свойствами его компонентов, но и характером и числом его связей с другими предложениями. На этом принципе построены модели автоматического реферирования, разработанные В.Е. Берзоном, а также С.М. Приходько и Э.Ф. Скороходько, И.П. Севбо и др.

В соответствии с одним из методов для отбора информационно наиболее значимых предложений задаётся некоторая пороговая величина, выражающая число связей. Регулируя эту величину, можно менять информационную насыщенность предложений и, следовательно, объём реферата. При этом характер самих предложений может не приниматься во внимание.

Другой, более сильный вариант этого метода заключается в том, что в тексте разграничиваются эксплицитные и имплицитные сверхфразовые единицы. При этом оказывается, что основное содержание несёт обычно первое предложение эксплицитной сверхфразовой единицы. Это предложение, как правило, является автосемантичным. Информативность первого предложения эксплицитной сверхфразовой единицы возрастает при увеличении длины этой единицы. К числу информативных также относятся крайние предложения с многократными покрытиями, под которыми понимаются перекрывающие межфразовые связи (Богданов 1996).

Учёт параметра межфразовых связей делает результирующий реферат более связным и целостным.

5.3. Методы с опорой на знания

В отличие от линейной модели в методах подбора выдержек, для подготовки краткого изложения информации, требуются мощные вычислительные ресурсы для систем обработки естественных языков, в том числе грамматики и словари для синтаксического разбора и генерации естественно-языковых конструкций. Кроме того, для реализации этого метода нужны некие онтологические справочники, отражающие соображения здравого смысла и понятия, ориентированные на предметную область, для принятия решений во время анализа и определения наиболее важной информации.

Как показано на рис. 5.2, метод формирования краткого изложения предполагает два основных подхода.

Первый (вверху) опирается на традиционный лингвистический метод синтаксического разбора предложений.

В этом методе применяется также семантическая информация для аннотирования деревьев разбора. Процедуры сравнения манипулируют непосредственно деревьями с целью удаления и перегруппировки частей, например, путём сокращения ветвей на основании некоторых структурных критериев, таких как скобки или встроенные условные или подчинённые предложения. После такой процеду-

ры дерево разбора существенно упрощается, становясь, по существу, структурной «выжимкой» исходного текста.

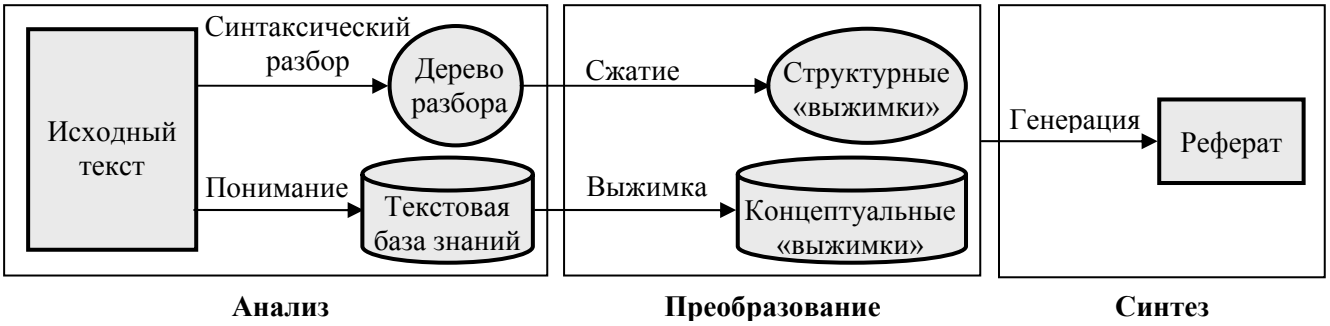

Рис. 5.2. Два основных подхода метода формирования краткого изложения

Второй подход к составлению краткого изложения уходит корнями в системы искусственного интеллекта и опирается на понимание естественного языка. Синтаксический разбор также входит составной частью в такой метод анализа, но деревья разбора в этом случае не порождаются. Напротив, формируются концептуальные репрезентативные структуры всей исходной информации, которые аккумулируются в текстовой базе знаний. В качестве структур могут быть использованы формулы логики предикатов или такие представления, как семантическая сеть или набор фреймов.

В процессе преобразования концептуальное представление претерпевает несколько изменений. Избыточная и не имеющая прямого отношения к тексту информация устраняется путём удаления поверхностных суждений или отсечения концептуальных подграфов. Затем информация подвергается дальнейшему агрегированию путём слияния графов (или шаблонов) или обобщения информации, например, при помощи таксономических иерархий отношений подклассов. В результате преобразования формируется концептуальная репрезентативная структура реферата, по существу, концептуальные «выжимки» из текста.

Наличие этих формальных репрезентативных слоев (структурные и концептуальные «выжимки») отличает подход, основанный на базе знаний, от подхода, не предполагающего опору на знания. Как видно из рис. 5.1. и 5.2, этап синтеза одинаков для обоих подходов: текстовый генератор преобразует структурное или концептуальное представление в естественно-языковую аннотацию. Некоторые системы предоставляют пользователю возможность управлять получаемыми «выжимками» методом указания, и не предполагают этапа генерации, при условии, что исходные тексты предоставляются наряду с их кратким изложением.

Упражнения:

1. Найдите на сайте http://news.yandex.ru и http://news.google.com новости на русском и английском языках объёмом 1000–2000 знаков.

2. Сделайте автореферат английской новости в MS Word и объясните, какие методы используются для реферирования в MS Word и для создания кратких аннотаций новостей на сайте http://news.google.com

3. Сделайте автореферат русской новости в программе TextAnalyzer и объясните, какие методы используются для реферирования в TextAnalyzer и для создания кратких аннотаций новостей на сайте http://news.yandex.ru

Вопросы к семинару

1. Инвертированный индекс.
2. Обработка булевых запросов.
3. Определение лексикона терминов.
4. Словопозиции с координатами и фразовые запросы.

Литература к семинару

1. Маннинг, К. Введение в информационный поиск: пер. с англ // Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце. — М.: ООО «И.Д. Вильямс», 2011. — С. 23–68.

Глава 6. Криптография

Криптография — это одна из древнейших наук, лежащая на стыке лингвистики и математики. Также как и при автоматическом реферировании в криптографии большое внимание уделяется сжатию информации, которое необходимо для более качественного зашифрования. Рука об руку с зашифрованием идёт криптоанализ, без методов которого немыслима дешифровка древних текстов. Что касается современных языков, то автоматический перевод текста можно рассматривать как дешифрование текста перевода, а исходный текст — как шифротекст, зашифрованный текст перевода с помощью особого шифра.

6.1. Основные понятия

Основным видом криптографического преобразования является шифрование. Под **шифрованием** понимают процесс преобразования открытой информации в зашифрованную (шифротекст) или процесс обратного преобразования зашифрованной информации в открытую. Процесс преобразования открытой информации в закрытую получил название **зашифрования**, а обратный процесс получения открытого текста или информации — **расшифрования** (Левин 2001, с.3–5).

Методом шифрования (шифром) называется совокупность обратимых преобразований открытой информации в закрытую посредством алгоритмов шифрования. Появление ЭВМ и компьютерных систем инициировало процесс разработ-

ки новых шифров, учитывающих возможности как зашифрования/расшифрования информации, так и атаки на шифр.

Атака на шифр (криптоанализ) — это процесс расшифрования закрытой информации без знания ключа и, возможно, при отсутствии сведений об алгоритме шифрования (Левин 2001, с.3–5).

Современные методы шифрования должны отвечать следующим требованиям:

1. Стойкость шифра против попыток взлома (криптостойкость) должна быть такой, чтобы вскрытие его могло быть осуществлено только путём решения задачи полного перебора ключей.
2. Криптостойкость обеспечивается не секретностью алгоритма шифрования, а секретностью ключа (Петров 2000, с. 24).
3. Шифротекст не должен существенно превосходить по объёму исходную информацию.
4. Ошибки, возникающие при шифровании, не должны приводить к искажению или потере информации.
5. Время шифрования не должно быть большим.
6. Стоимость шифрования должна быть согласована со стоимостью закрываемой информации (Завгородний 2001, с. 134).

Криптостойкость шифра является его основным показателем эффективности. Она измеряется временем и стоимостью средств, необходимых криптоаналитику для получения исходной информации по шифротексту, при условии, что ему неизвестен ключ.

Сохранить в секрете широко используемый алгоритм шифрования практически невозможно. Поэтому алгоритм не должен иметь слабых мест, которыми могли бы воспользоваться взломщики. Если это условие выполняется, то криптостойкость шифра определяется длиной ключа, так как единственный путь вскрытия зашифрованной информации — перебор комбинаций ключа и выполнение алгоритма расшифрования. Таким образом, время и средства, затрачиваемые на криптоанализ, зависят от длины ключа и сложности алгоритма (Петров 2000, с. 24).

6.2. Немного из современной истории криптографии

Выдающиеся результаты применения математических методов в криптографии принадлежат Клоду Шеннону. Разработанные К. Шенноном концепции теоретической и практической секретности (или криптостойкости) позволяют качественно оценивать криптографические качества шифров и пытаться строить идеальные или совершенные шифры.

Центральной в работах К. Шеннона является концепция избыточной языковой информации, содержащейся в текстовых сообщениях. Избыточность означает, что в сообщении содержится больше символов, чем достаточно для передачи содержащейся в нём информации. То есть, такие единицы как артикли, предлоги, различные местоимения можно изъять из текста без потери информации. Шеннон показал, что успех криптоанализа определяется тем, насколько избыточность, со-

держащаяся в открытом тексте «переносится» в шифрованное сообщение. Если шифрование «стирает» избыточность, то восстановить текст сообщения по криптограмме становиться принципиально невозможно.

В работах К. Шеннона важную роль играет величина удельной энтропии *H* на букву текста, другими словами, среднее количество информации, передаваемой буквой открытого текста (Основы криптографии 2001, с. 45, 47–48).

Методы, предложенные в работах К. Шеннона, развили в дальнейшем А.Н. Колмогоров и Р.Г. Пиотровский. Они указали на существенную разницу между значениями *H* для различных функциональных стилей (литературного, разговорного, делового).

Следующая страница в истории криптографии посвящена телефонным шифраторам, которые были разработаны в 30-х гг. XX в. Идея телефонного шифратора была запатентована Д.Х. Роджерсом ещё в 1881 г., спустя 5 лет после изобретения А.Г. Беллом телефона. Идея состояла в передаче телефонного сигнала по нескольким цепям поочередными импульсами в некоторой быстро меняющейся последовательности. Линии должны были располагаться на довольно большом расстоянии друг от друга, чтобы исключить возможность одновременного подключения к ним. Подключение же к одной из них позволяло бы слышать только отдельные неразборчивые сигналы.

В более поздних работах предлагались различные преобразования уже самой речи: инверсия, смещение, или деление диапазона частот, шумовые маскировки, временные приостановки частей сигнала, а так же различные комбинации перечисленных преобразований (Основы криптографии 2001, с. 49).

Во второй половине XX века, вслед за развитием вычислительной техники, появились электронные шифраторы, разработка которых потребовала серьёзных теоретический исследований во многих областях прикладной и фундаментальной математики.

В 70-х гг. XX в. произошло два события, серьёзно повлиявшие на дальнейшее развитие криптографии. Во-первых, был принят первый стандарт шифрования данных (DES), постулирующий принцип Керкгоффса (стойкость шифра определяется лишь секретностью ключа) в криптографии. Во-вторых, после работы американских математиков У. Диффи и М. Хеллмана была создана новая «отрасль» криптографии — криптография с открытым ключом. Оба эти события были рождены потребностями развивающихся средств коммуникаций, в том числе и глобальных и локальных компьютерных систем.

В 1978 г. Р. Райвестом, А. Шамиром и Л. Адлеманом была предложена система RSA с реально реализованным открытым ключом. В ней для зашифрования и расшифрования использовались совершенно разные ключи, причём ключ зашифрования был открытым и всем известным. Надёжность этого метода опирается на трудность факторизации больших чисел. Р. Райвест и его коллеги заявляли, что для факторизации 200-значного числа понадобится 4 миллиарда лет работы компьютера с быстродействием 106 операций в секунду (Левин 2002, с. 138).

В связи с асимметричным использованием ключей стал использоваться термин асимметричная шифрсистема, в то время как традиционные шифрсистемы стали

именоваться симметричными. Наряду с системой открытого шифрования У. Диффи и М. Хеллман предложили идею открытого распределения ключей, позволяющую избавиться от защищённого канала связи при рассылке криптографических ключей (Столлингс 2001, с. 82).

Сегодня именно электронные шифраторы представляют подавляющую долю средств шифрования.

6.3. Классификация методов криптографической обработки информации

Под криптографической защитой информации понимают такое преобразование исходной информации, в результате которого она становится недоступной для ознакомления и использования лицами, не имеющих для этого полномочий.

Известны различные подходы к классификации методов криптографического преобразования информации. По виду воздействия на исходную информацию методы криптографической защиты делятся на четыре группы (рис. 6.1) (Завгородний 2001, с. 134).

Процесс шифрования заключается в проведении обратимых математических, логических, комбинаторных и иных преобразований исходной информации, в результате которых зашифрованная информация представляет собой хаотический набор букв, цифр, других символов и двоичных кодов. Для шифрования информации используется алгоритм преобразования и ключ. Как правило, алгоритм для определённого метода шифрования остаётся неизменным. Исходными данными для алгоритма шифрования служит информация, подлежащая зашифровке, и ключ шифрования. Ключ содержит управляющую информацию, которая определяет метод преобразования на определённых шагах алгоритма и величины операндов, используемые для реализации процесса шифрования.

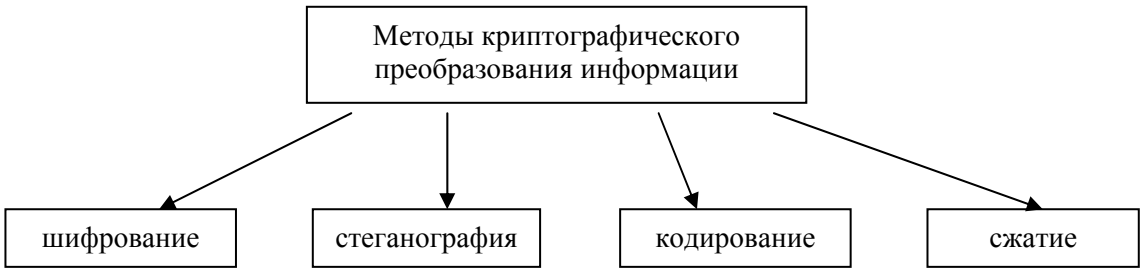

Рис. 6.1

В отличие от других методов шифрования, методы стеганографии позволяют не только скрыть смысл хранящейся или передаваемой информации, но и сам факт хранения или передачи закрытой информации. В основе всех методов стеганографии лежит маскирование закрытой информации среди открытых файлов.

Одним из методов скрытой передачи информации является простой метод скрытия файлов в операционной системе MS DOS. За текстовым открытым файлом записывается скрытый двоичный файл, объём которого много меньше текстового файла. В конце текстового файла помещается отметка EOF (End Of File).

При обращении к этому текстовому файлу стандартными средствами операционной системы считывание прекращается по достижении метки EOF, и скрытый файл остаётся недоступен. Для двоичных файлов никаких меток в конце не предусмотрено. Конец такого файла определяется при специальной обработке, при которой выдаётся информация о длине файла в байтах. Доступ к скрытому файлу может быть получен, если файл открыть как двоичный. Скрытый файл может быть зашифрован. Если кто-то случайно обнаружит такой файл, то он будет рассмотрен просто как сбой в работе системы.

Для скрытой передачи информации в звуковых и графических файлах используется несколько другой метод. Графическая и звуковая информация представляется в числовом виде. Так, например, в графических объектах наименьший элемент может представляться одним байтом. В младшие разряды определённых байтов изображения в соответствии с алгоритмом криптографического преобразования помещаются биты скрытого файла. Если правильно подобрать алгоритм размещения скрытого файла, то человеческим глазом будет невозможно отличить полученное изображение от исходного. Комплексное использование стеганографии и шифрования многократно повышает сложность задачи обнаружения и раскрытия конфиденциальной информации.

Содержанием процесса кодирования информации является замена смысловых конструкций исходной информации (слов, предложений) кодами. В качестве кодов могут использоваться сочетания букв, цифр, букв и цифр. При кодировании и обратном преобразовании используют специальные таблицы и словари. Кодирование информации целесообразно применять в системах с ограниченным набором смысловых конструкций. Недостатками кодирования конфиденциальной информации является необходимость хранения и распространения кодировочных таблиц, которые необходимо часто менять, чтобы избежать раскрытия кодов статистическими методами обработки перехваченных сообщений.

Сжатие информации может быть отнесено к методам криптографического преобразования информации с некоторыми оговорками. Целью сжатия является сокращение объёма информации. В то же время сжатая информация не может быть прочитана или использована без определённых обратных преобразований. Но, учитывая доступность методов сжатия и обратного преобразования (архиваторы ZIP, RAR и т.д.), метод сжатия не может рассматриваться как надёжный метод криптографического преобразования информации. Даже если держать в секрете алгоритмы, они сравнительно легко могут быть раскрыты статистическими методами обработки. Поэтому сжатые файлы конфиденциальной информации подвергаются последующему шифрованию. Для экономии времени целесообразно совмещать процесс сжатия и шифрования информации.

6.4. Вскрытие шифров перестановки

Возьмём пример шифровки двойной перестановки. Пусть имеется шифровка *АЗЮЖЕ СШГТООИПЕР*, которая так укладывается в таблицу 4х4 (рис. 6.2).

Рассматривая маловероятные сочетания букв, криптоаналитику легко будет найти истинную последовательность столбцов. Так, сочетание *ГТ* в 3 строке шифровки указывает на то, что после 1 столбца с малой вероятностью следует 2 столбец. Рассчитаем статистически, какой столбец, скорее всего, следует за 1. Для этого можно воспользоваться таблицей логарифмов вероятностей биграмм русского текста (табл. 6.1). Вероятность следования одного столбца за другим равна произведению вероятностей биграмм в строках этих столбцов. Поскольку в таблице даны логарифмы биграмм, то их достаточно суммировать, а потом выбрать сочетание столбцов с максимальной вероятностью. Для вероятностей следования за первым столбцом 2, 3 и 4 имеем выражения:

	1	2	3	4
1	А	З	Ю	Ж
2	Е		С	Ш
3	Г	Т	О	О
4	И	П	Е	Р

Рис. 6.2

$$p(1–2) = p(АЗ) \cdot p(Е\ ) \cdot p(ГТ) \cdot p(ИП) = 7+9+0+5 = 21$$
$$p(1–3) = p(АЮ) \cdot p(ЕС) \cdot p(ГО) \cdot p(ИЕ) = 6+8+8+8 = 30$$
$$p(1–4) = p(АЖ) \cdot p(ЕШ) \cdot p(ГО) \cdot p(ИР) = 7+5+8+7 = 27$$

В нашем случае наиболее вероятно, что после столбца 1 следует столбец 3. Для такой небольшой таблицы шифрования, которую мы имеем, можно перебрать все варианты перестановок — их всего лишь 24. В случае большого числа столбцов целесообразно оценить вероятности пар сочетаний разных столбцов и решить задачу оптимизации, которая укажет перестановку столбцов, дающую фрагменты естественного текста с наибольшей вероятностью. В данном случае наилучший результат достигается при расстановке столбцов (2413), что примерно вдвое по вероятностной оценке достовернее ближайшей к ней по вероятности расстановки (4132). После того, как столбцы шифровки расставлены, нам не составит труда правильно расставить и её строки по смыслу фрагментов текста (рис. 6.3).

	2	4	1	3
1	З	Ж	А	Ю
2		Ш	Е	С
3	Т	О	Г	О
4	П	Р	И	Е

Рис. 6.3

Текст в ней уже читается, и, расставив строки в порядке (4123), получим расшифровку: ПРИЕЗЖАЮ ШЕСТОГО.

Вопросы к семинару

1. История русской тайнописи.
2. Шифр «Магический квадрат».
3. Метод Юлия Цезаря.
4. Квадрат Полибия.
5. Метод Гронсфельда.
6. Метод Порта.
7. Симметричный алгоритм DES.
8. Ассиметричный алгоритм RSA.

Таблица 6.1

Логарифмы вероятностей биграмм

	АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ_	Процент
А	27867777477788376782667755000067 9	71
Б	71101622605635727507054105572203 5	13
В	80504803716756846666030130082004 8	38
Г	60116500604544807006001200000000 4	10
Д	81634810704717846527133340064045 7	25
Е	55678666477889658893365656001155 9	73
Ж	60006721705027101213000000002000 2	8
З	84626411615566715006002100262004 6	14
И	66766857777688557881577763010067 9	64
Й	00303000003654000660001230000000 8	11
К	81511652712705807667006010000000 7	29
Л	84121861804416700336300311068078 6	31
М	75722801704476851316100000073006 8	32
Н	90336811906017800576525300085046 7	50
О	28888677687887678883256765001525 9	89
П	70000804703614849456201000045304 4	28
Р	91644860805266842667354242074016 7	48
С	64625720707866875696351550056138 7	43
Т	82714800806456938846000402178015 8	60
У	34466765336556067771550636000074 8	19
Ф	60000500600220604035400000010000 2	4
Х	43300400301105605313002000100000 8	6
Ц	50600600700000300004000000050000 5	6
Ч	70100800706106201073000130013000 4	12
Ш	50000600703303403034000000004000 5	3
Щ	60000700600002002004000000004010 1	3
Ъ	00000400000000000000000000000003 2	1
Ы	14735715175562155560070541010001 8	18
Ь	01000307106047100640000161000062 8	12
Э	00400100026521020170430000000000 1	4
Ю	05002012041000000370000617001030 7	6
Я	01525625022365014470044304000064 9	18
_	78978758838689998987767851121826 0	138

Упражнения:

1)

	1	2	3	4
1	А	Т	О	
2	Б	У	С	Б
3	Г	Е	С	О
4	Я	Н	Д	

2)

	1	2	3	4
1	Е	К	Н	
2	Е	И	Т	С
3	И	К	И	Ж
4	П	И	Н	Р

Глава 7. Системы автоматического синтеза речи

Система автоматического синтеза речи (АСР, англ. Text-to-Speech, TTS) — это компьютеризированная система, которая может произносить любой текст вне зависимости от того, введён ли он в компьютер оператором или отсканирован и распознан системой оптического распознавания символов (ОРС, англ. Optical Character Recognition, OCR).

Такие системы отличаются от любых других говорящих машин (таких как, к примеру, магнитофон) в том смысле, что нас интересует автоматическое произнесение новых предложений. Системы, которые просто склеивают изолированные слова или части предложений, называемые *системами с речевым ответом* (*Voice Response Systems*), применимы только тогда, когда используется ограниченный словарь (несколько сотен слов) и ограниченная структура (к примеру, в объявлениях о прибытии поездов на вокзале).

Система АСР используется в тех случаях, когда невозможно (и бесполезно) записать и хранить все слова языка. Таким образом, систему АСР можно определить как *систему автоматической генерации речи посредством графемно-фонемной транскрипции предложений.*

7.1. Сферы применения АСР

Сферы применения систем АСР очень разнообразны. Приведём несколько примеров.

1. **Средства телекоммуникаций**. Системы АСР можно использовать для получения текстовой информации по телефону. В качестве текстов могут быть как простые сообщения, такие как важные культурные события (в кинотеатрах, театрах и т.д.), так и огромные базы данных, в которых трудно хранить записанную речь. Запросы к таким информационно-поисковым системам могут быть организованы посредством голоса пользователя (при помощи системы распознавания речи) или клавиатуры телефона (с тональным набором). К примеру, в некоторых телефонных компаниях организованы дополнительные сервисы: чтение электронных сообщений или факсов по телефону, телефонная релейная связь (передача текстовых сообщений, генерируемых по голосу отправителя и синтезируемых в голос для получателя), и автоматизированная телефонная справочная служба. В большинстве таких служб естественность компьютерного голоса не столь важна.

2. **Обучение языку**. На сегодняшний день системы АСР практически не применяются в этой области, так как их низкое качество не удовлетворяет требованиям таких задач. Если же используется качественный синтезатор речи, то большинство текстовой информации должно дополнительно сопровождаться интонационными управляющими тегами (например, для различения слов *за́мок* и *замо́к*).

3. **Помощь инвалидам**. С помощью специально спроектированных клавиатур и программ быстрого составления предложений речь может быть синтезирована за несколько секунд, позволив немым общаться с другими людьми. К примеру, астрофизик Стивен Хокинг (Stephen Hawking) читает таким образом все свои лек-

ции. Слепые могут получать информацию с помощью устройств, в которых сочетаются системы АСР и ОРС.

4. **Говорящие книги и игрушки**.

5. **Голосовой контроль**. В некоторых случаях произнесённая информация более эффективна, чем текстовые сообщения. Она обращает на себя больше внимания, в то время как можно получать другую информацию из визуальных источников. Данная идея привела к встраиванию речевых синтезаторов в системы контроля и измерения.

6. **Мультимедиа, человеко-машинное общение**.

7.2. Устройство системы АСР

На рис. 7.1 показана самая общая схема системы АСР. Она состоит из модуля обработки естественного языка (ОЕЯ), преобразующего текст в фонетическую транскрипцию, с необходимой интонацией и ритмом, и модуля обработки цифрового сигнала (ОЦС), который преобразует символьную информацию в речь (Dutoit 1997).

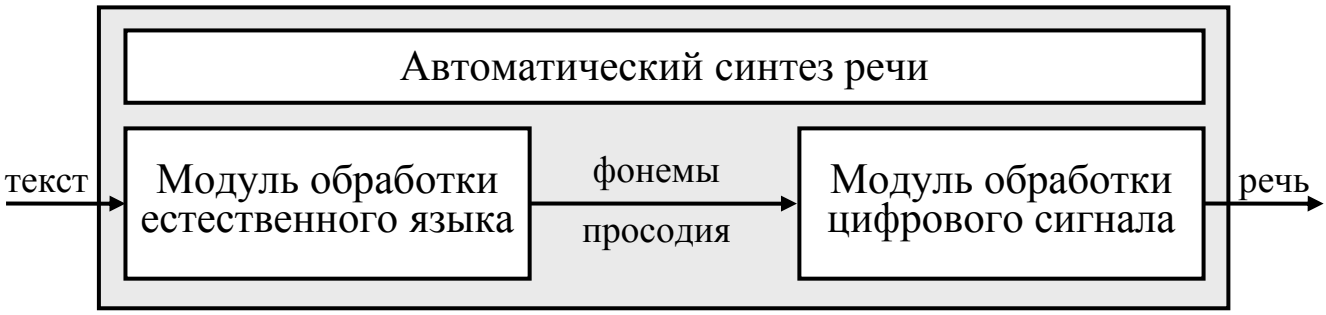

Рис. 7.1

7.2.1. Модуль обработки естественного языка

Общая схема устройства модуля ОЕЯ показана на рис. 7.2. Кроме ожидаемого компонента транскрибирования букв и блока генерации просодии, в данном модуле имеется также морфо-синтаксический анализатор, необходимый для нужд генерации речи высокого качества. Преобразование предложения в последовательность частей речи и дальнейшее его представление в виде синтаксического дерева необходимо, по крайней мере, по двум причинам:

1. Правильная фонетическая транскрипция для некоторых слов может быть записана, только если известны их часть речи и отношения зависимости между другими словами.

2. Естественная просодия очень зависит от синтаксиса. Семантика и прагматика также играет в этом большую роль, но так как эти аспекты ещё мало изучены, существующие системы АСР в основном опираются только на синтаксис. Хотя в некоторых системах предусмотрено полное разрешение многозначности.

7.2.2. Анализ текста

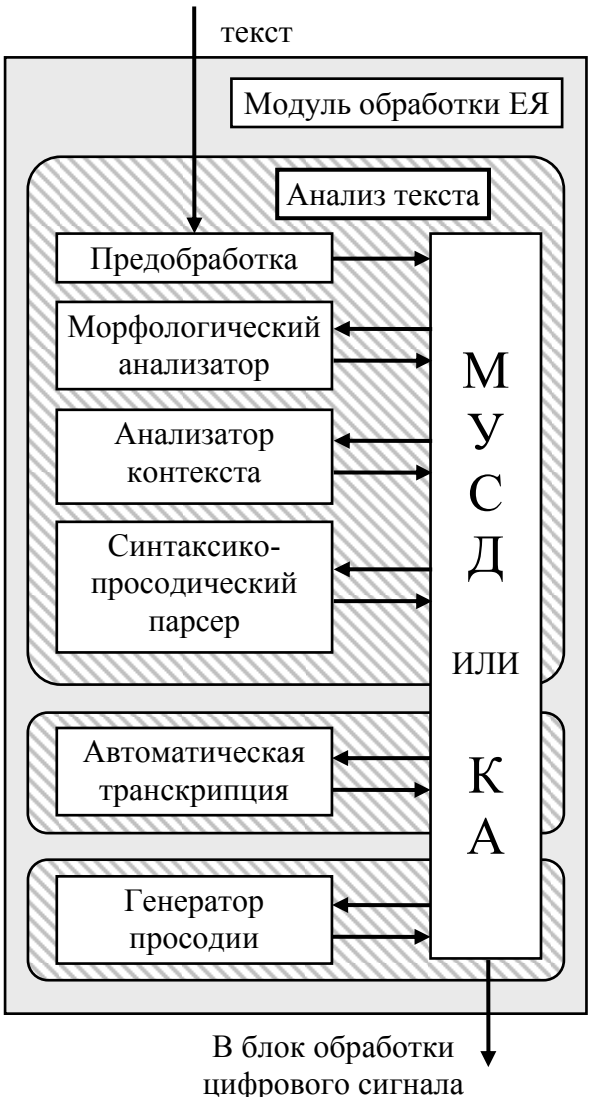

Рис. 7.2

Блок анализа текста (рис. 7.2) состоит из следующих компонентов:

1. Модуль предобработки разделяет текст на слова. Здесь выявляются числа, сокращения, акронимы и идиоматичные выражения, которые преобразуются в полнословные эквиваленты. На данном этапе возникает проблема многозначности пунктуационных знаков (например, определение границ предложений), которая обычно решается с помощью элементарных регулярных грамматик.

2. Модуль морфологического анализа определяет все возможные части речи слова. Словоизменительные, словообразовательные и сложные слова раскладываются на морфемы при помощи простых регулярных грамматик, которые используют лексиконы основ и аффиксов.

3. Модуль анализа контекста позволяет сократить список возможных частей речи слова до небольшого набора гипотез высокой вероятности, выводимых из частей речи соседних слов. Это достигается либо за счёт *n-грамм*, которые описывают локальную синтаксическую зависимость в виде вероятностного конечного автомата или многослойного персептрона (т.е. нейронной сети), либо за счёт локальных нестохастических грамматик, созданных экспертами-лингвистами или автоматически полученных из тестового набора данных при помощи деревьев решений (*classification and regression tree,* CART).

4. Наконец, синтаксико-просодический парсер просматривает оставшееся пространство поиска и определяет структуру текста (организацию клауз и фраз), которая больше всего похожа на её ожидаемую просодическую реализацию.

7.2.3. Автоматическая транскрипция

Использование только словарей произношения не может помочь в записи транскрипции слова:

1. Словари произношения содержат только транскрипцию лемм. В них не даны все морфологические варианты слов (мн.ч., ж.р., склонения, спряжения), которые описываются *морфофонологией*.

2. В словаре может быть несколько омографов, имеющих разное произношение (*record:* ['rekɔ:d] или [rɪ'kɔ:d]) или омоформ (*read*: ['ri:d] или ['red]).

3. В задачу морфофонологии входит также описание правил для таких случаев образования новых слов из лемм, как феномен изменения произношения согласных в таких словах как *electricity* из леммы *electric,* редукция *t* в слове *softness* ['sɔfnɪs] и т.д., которые невозможно полностью перечислить в словаре.

4. Слова в окружении других слов звучат не так, как по-отдельности.

5. Наконец, не все слова можно найти в словаре произношения: неологизмы, собственные имена, которые должны произноситься по типу существующих слов.

Ясно, что необходимо использовать первичный морфосинтаксический (для 1–3), синтаксический или метрически анализ предложения (для 4), модуль морфологии и/или модуль нахождения графемной аналогии между словами (для 5).

Таким образом, модуль автоматической транскрипции можно организовать поразному, но часто выделяют стратегии, основанные на использовании словаря или правил, хотя существуют и промежуточные подходы.

Стратегия использования словаря подразумевает хранение большинства фонологической информации в лексиконах (рис. 7.3). Для уменьшения размера словаря его обычно разбивают на словари основ и аффиксов, а произношение поверхностных форм вычисляется на основе словоизменительных, словообразовательных морфофонемных правил и правил словосложения, которые описывают, как фонетическая транскрипция морфемных составляющих изменяется при образовании слов. Морфемы, не найденные в словаре, транскрибируются по правилу. После того, как каждому слову приписана фонемная транскрипция, выполняется последующая фонетическая обработка для учёта коартикуляционного феномена. Такой подход используется в системе MITalk. Её словарь, включающий около 12 000 морфем, покрывает 95% входных слов. Компания AT&T Bell Laboratories также использует этот подход в своей системе ACP, но её словарь увеличен до 43 000 морфем.

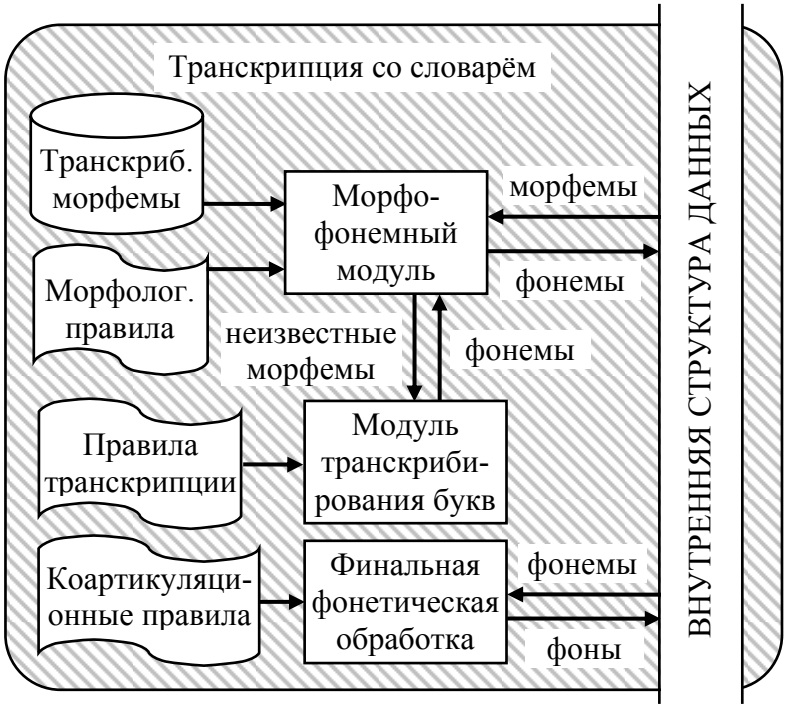

Рис. 7.3

Другая стратегия применяется в транскрипционных **системах, основанных на правилах** (рис. 7.4). В ней большинство фонологической компетенции словарей переносится во множество правил преобразования графема-фонема. В словаре хранятся только слова-исключения. Так как многие исключения являются очень частотными словами, достаточно небольшой словарь исключений может покрыть слова большинства текстов. К примеру, в английском языке 2000 слов достаточно, чтобы покрыть 70% слов текста.

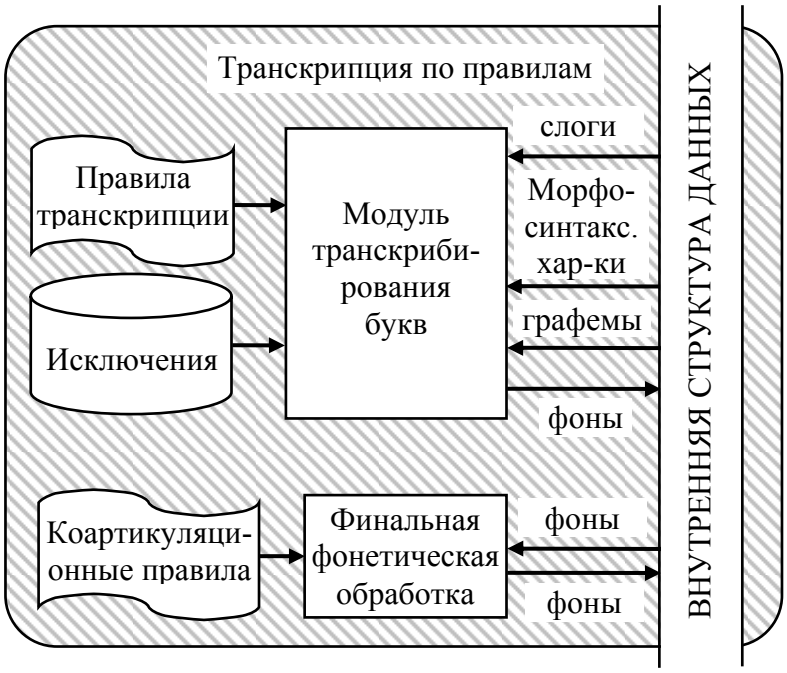

Рис. 7.4

Выбор подхода очень часто зависит от языка.

7.3. Модуль обработки цифрового сигнала

Существует два способа преобразования фонем в звук:

- явно, в виде набора правил, которые формально описывают влияние одних фонем на другие;
- неявно, путём хранения примеров фонетических переходов и коартикуляций в базе данных отрезков речи и использования их, в конечном счёте, вместо фонем.

Указанные способы лежат в основе двух подходов к акустическому синтезу речи: различаются *синтез по правилам* и *синтез на основе конкатенации* (или *компилятивный синтез*).

7.3.1. Акустический синтез по правилам

Синтезаторы данного типа реализуются обычно в виде формантных синтезаторов и используются в основном фонетистами и фонологистами. К примеру, синтезатор Klatt (Klatt 80) широко используется, т.к. позволяет изучать характеристики естественной речи путём аналитического прослушивания синтезированной речи. Практика показывает, что для получения достаточно естественной речи требуется около 60 параметров, характеризующих передаточную функцию речевого тракта, т.е. задающих частоту, амплитуду и ширину формант и антиформант речевых звуков. Кроме того, необходимы правила для параметров, которые описывают динамически меняющиеся характеристики голосового и шумовых источников звука.

Создание формантного синтезатора по правилам требует большой подготовительной работы, связанной с акустическим анализом значительных речевых массивов и формулировкой нужных правил (на это требуется обычно несколько лет).

Синтезированная речь, получаемая с помощью современных формантных синтезаторов, часто сопровождается гудением или жужжанием, что снижает её естественность. В то же время формантный синтез по правилам даёт возможность имитировать разные голоса, отслеживать изменения в стиле произношения и тембре голоса.

Формантный синтез описанного типа используется в целом ряде систем АСР, из которых наиболее известны MITALK, DECTALK для английского языка и многоязычная система синтеза INFOVOX (Кодзасов 2001).

7.3.2. Акустический синтез на основе конкатенации

В синтезаторах этого направления грубая первичная основа акустического сигнала создаётся на основе *конкатенации* (склейки) акустических образцов, которые хранятся в памяти компьютера в виде особой акустической базы данных.

Подготовка базы данных

Сначала выбираются такие сегменты речи, чтобы избежать дальнейших трудностей при их конкатенации. В качестве них часто выбираются дифоны (отрезки, начинающиеся в середине одного звука и заканчивающиеся в середине соседнего), полуслоги и трифоны (которые отличаются от дифонов тем, что включают полностью центральный фон). Во всех случаях решающим является требование сохранить без изменения переходные участки, на которых происходит фонетическое взаимодействие звуков. Нежелательные позиции в словах (внутри ударных слогов или слишком редуцированных контекстах) не берутся во внимание. Затем записывается корпус речи и сегментируется либо вручную при помощи визуальных средств представления цифрового сигнала, либо автоматически с последующей корректировкой (рис. 7.5). Результаты записываются в акустическую базу данных в виде названия сегмента, формы волны, продолжительности и позиций внутренних сегментов. Например, при записи дифонов также необходимо указывать границу между фонами, чтобы можно было изменить продолжительность одного полуфона, не затрагивая длину другого.

Зачастую сегменты впоследствии приводятся к параметрическому виду, последовательности векторов параметров, выдаваемых *речевым анализатором.* В такой форме они хранятся в *параметрической базе данных отрезков речи*. Преимущества использования речевой модели в следующем:

- Удачно выбранные речевые модели позволяют сократить размер базы данных звуков.
- Некоторые модели разделяют сигнал на источник звука и фильтра (голосовой тракт), что очень помогает при операциях, непосредственно предшествующих синтезу: подбору просодии и конкатенации сегментов.

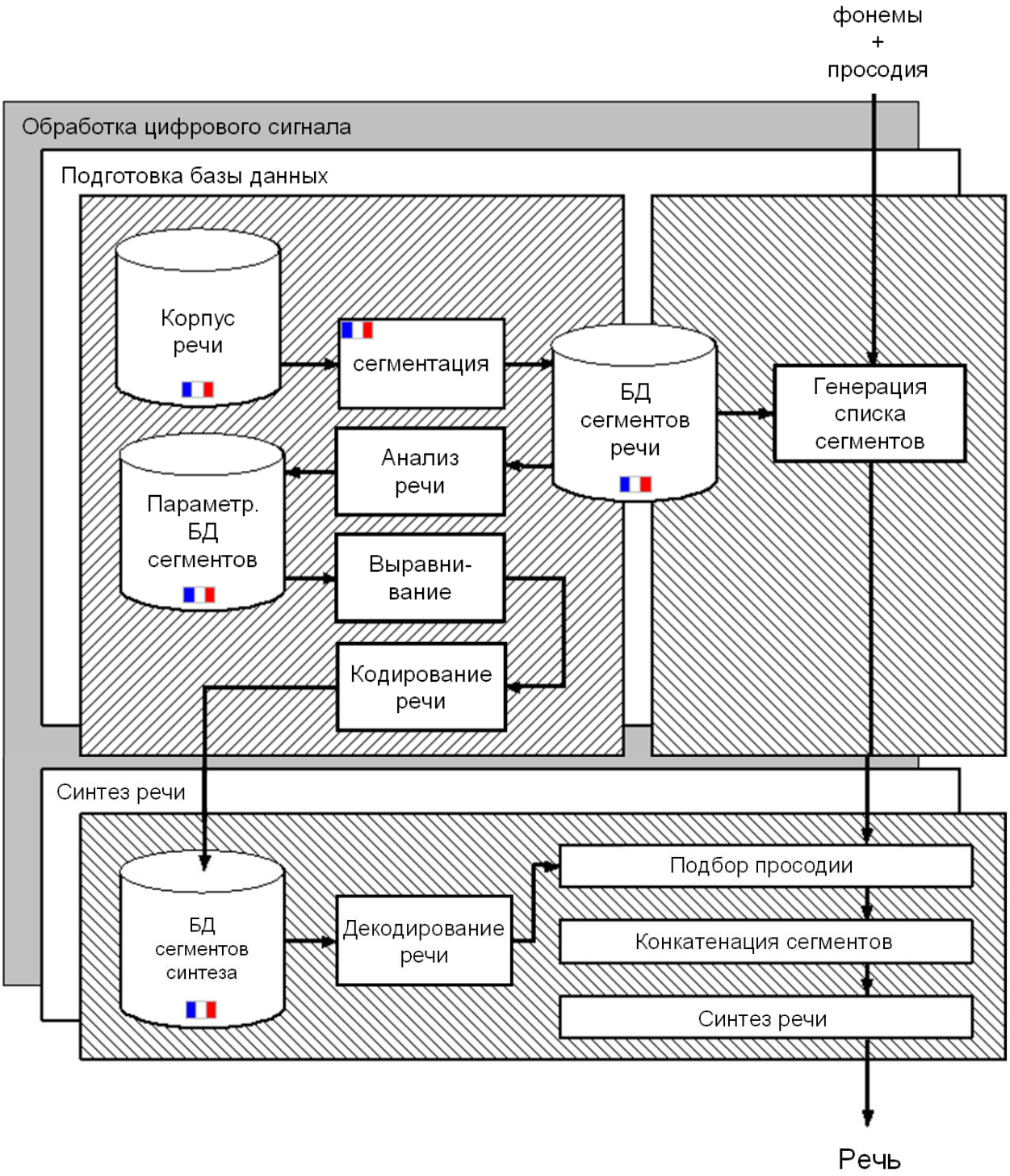

Рис. 7.5. Общая схема компилятивного синтезатора. Верхний левый блок соответствует стадии разработки (т.е. выполняется только один раз). Другие блоки соответствуют операциям, выполняемым во время синтеза. Операции и данные, зависимые от конкретного языка отмечены флагом.

Параметрическое представление отрезков речи позволяет легко изменять высоту тона, длину волны и огибающую спектра, что нельзя сделать с просто записанными отрезками речи.

Так как отрезки речи были получены из разных слов, т.е. из разных фонетических контекстов, при конкатенации их амплитуда и тембр могут не совпадать. Это

можно исправить при создании *базы данных сегментов синтеза* с помощью выравнивания спектра амплитуды на концах сегментов, оставляя без изменения сами сегменты. На практике же уровни громкости плавно корректируются в начале и конце сегмента, а громкости различных фонов одной фонемы приводятся к её среднему значению. В отличие от амплитуды разница в тембре снимается во время синтеза для каждой пары сегментов, а не выравнивается для всех сегментов сразу, оставляя естественное фонетическое различие в коартикуляциях. На практике выравнивание уровней амплитуды можно проводить как до анализа речи, так и после (т.е. либо на необработанных отрезках, либо на параметрах речи). После того, как сформирована параметрическая база данных сегментов, можно приступить собственно к синтезу речи.

Синтез речи

Первоначально последовательность сегментов выводится из фонемного входа синтезатора в блоке генерации списка сегментов, который служит интерфейсом между модулями ОЕЯ и ОЦС. Как только просодические такты правильно соотнесены с каждым сегментом, *модуль подбора просодии* запрашивает в базе данных сегментов синтеза параметры этих сегментов и приводит их в соответствие один за другим с требуемой просодией. В *блоке конкатенации сегментов* производится подгонка сегментов, производится простая интерполяция параметров, приблизительно соответствующая сглаживанию акустических переходов между звуками. Полученная последовательность параметров передаётся в *блок синтеза сигнала*, в котором производится операция обратная анализу речи — синтез речи (Dutoit 1997).

Упражнения:

1. Расставить управляющие теги в абзаце текста на русском, английском и испанском языках для корректировки произношения синтезатора речи.

Глава 8. Системы автоматического распознавания речи

8.1. Метод Байеса

Представьте себе ситуацию, когда нужно интерпретировать последовательность фонов [nɪ], когда они встречаются после слова *I* в начале предложения. Всего слов, которые могут быть произнесены как [nɪ] 7! А именно: *the, neat, need, new, knee, to,* и *you*.

Системы автоматического распознавания речи (АРР) часто используют архитектуру, в которой каждое произношение ставится в соответствие со всеми возможными вариантами интерпретации, которые затем сохраняются в базе со значениями вероятности. Таким образом, не нужно генерировать список возможных кандидатов (как это делается, к примеру, в программах проверки орфографии).

Необходимо выбрать такое слово из списка кандидатов, чья априорная вероятность и условная вероятность максимальны в соответствии с формулой (8.1), где *y* представляет собой последовательность фонов (в данном случае [nɪ]) и *w* — слово-кандидат (*the, new* и т.д.).

$$\hat{w} = \underset{w \in W}{\operatorname{argmax}} \overbrace{P(y|w)}^{\text{условная}} \overbrace{P(w)}^{\text{априорная}} \tag{8.1}$$

Одним из простых методов генерации вероятности произношения являются правила вероятности. Правила вероятности были впервые предложены для задачи распознавания речи Уильямом Лабовым (Labov 1969). Идея заключается в том, чтобы ассоциировать правила произношения с вероятностями. Затем мы можем использовать эти правила вероятности для обработки лексикона и сгенерировать вероятности для различных возможных поверхностных форм.

К примеру, рассмотрим правило назализации, которое объясняет, почему *the* может быть произнесено как [nɪ]. Начальный [ð] произносится [n], если предыдущее слово оканчивается на [n] или иногда на [m]:

$$[0{,}15]\ \text{ð} \rightarrow \text{n} \ /\ [+\text{nasal}]\ \#__ \tag{8.2}$$

Значение [0,15] слева от правила — это вероятность. Она может быть сгенерирована при использовании достаточно большого корпуса, к примеру, затранскрибированной части корпуса Switchboard. Пусть *ncount* — число раз, когда фонема [ð] заменяется фоном [n] в начале слова, если предыдущая слово заканчивается на носовой звук (91 в корпусе Switchboard). Пусть *envcount* — число появлений фонемы [ð] (вне зависимости от того, в качестве какого фона она реализуется), когда предыдущее слово заканчивается на носовой звук (617 в корпусе Switchboard). Тогда получаемая вероятность:

$$P(\text{ð} \rightarrow \text{n} \ /\ [+\text{nasal}]\ \#__) = \text{ncount/envcount} = 91/617 = 0{,}15$$

Аналогичным образом мы можем создать правила вероятности для ассимиляции и опущения фонем, которые объясняют произношение [nɪ] для других слов. В

табл. 8.1 приведены некоторые правила произношения и соответствующие им вероятности, вычисленные на корпусе Switchboard.

Таблица 8.1

Слово	**Фонетическое явление**	**Правило**	**P**
the	назализация	ð → n / [+nasal] #__	0,15
neat	опущение конечного *t*	t → 0 / V __ #	0,52
need	опущение конечного *d*	d → 0 / V __ #	0,11
new	продвижение *u* вперёд	u → i / __ # [y]	0,36

Теперь нам необходимо посчитать априорную вероятность *P(w)* для каждого слова. Для орфографической проверки слов в качестве значения можно использовать относительную частоту слова в большом корпусе текстов. Например, слово, найденное 44 000 раз в корпусе размером 44 млн. слов получит оценку вероятности 44 000 / 44 000 000 = 0,001. Для задачи распознавания речи давайте посчитаем априорную вероятность на корпусах письменной и звучащей речи.

Корпус Брауна содержит 1 миллион словоформ из 500 элементарных выборок разных письменных жанров (газет, романов, научно-технической литературы и т.д.). Он был создан в университете Брауна в 1963–64 гг. (Kučera 1967). Корпус Switchboard содержит 1,4 млн. словоформ затранскрибированных телефонных разговоров. В табл. 8.2 показаны вероятности наших 5 слов. Каждая вероятность посчитана на основе частот этих двух корпусов, нормализованных относительно количества слов в обоих корпусах (плюс 0,5 * количество слов, т.е. знаменатель равнялся 2 486 075 + 30 836).

Таблица 8.2

Слово (w)	**Частота**	**P(w)**
knee	61	0,000024
the	114 834	0,046
neat	338	0,00013
need	1 417	0,00056
new	2 625	0,001

Сейчас мы практически готовы ответить на наш исходный вопрос: какое слово соответствует произношению [nɪ] при условии, что предыдущим словом было *I* в начале предложения. Давайте перемножим априорные и условные вероятности, чтобы получить оценку вероятности. Они показаны в табл. 8.3 в порядке от наиболее вероятного к наименее вероятному слову (вероятность *the* равняется 0, т.к. предыдущий фон отличается от [n] и нет других правил, позволяющих фонеме [ð] реализоваться в виде фона [n]).

Таблица 8.3

Слово (w)	**p(y|w)**	**p(w)**	**p(y|w)*p(w)**
neat	0,52	0,00013	0,000068
need	0,11	0,00056	0,000062
knee	1,00	0,000024	0,000024
the	0	0,046	0

Таким образом, наиболее вероятным получается слово *neat*. Но это неверный ответ, в корпусе Switchboard последовательности [nɪ] после слова *I* соответствует

слово *need*. Чтобы правильно решить эту задачу необходимо знание порядка слов. Носители английского языка знают, что последовательность слов *I need*... более вероятна, чем последовательность *I neat*.... Тем не менее, не стоит отказываться от метода Байеса, чтобы решить эту проблему. Достаточно изменить модель таким образом, чтобы она учитывала вероятности следования слов друг за другом (биграмм).

В действительности, метод Байеса используется в той или иной модификации во всех современных системах APP (Jurafsky 2000).

8.2. Архитектура системы APP

Системы распознавания речи оперируют акустическим сигналом, предполагая, что он является «шумным» вариантом исходного предложения. Для того чтобы декодировать такое предложение с шумом, необходимо рассмотреть все возможные предложения и для каждого вычислить вероятность, генерируя предложения с шумом. Затем необходимо выбрать предложение с максимальной вероятностью (рис. 8.1).

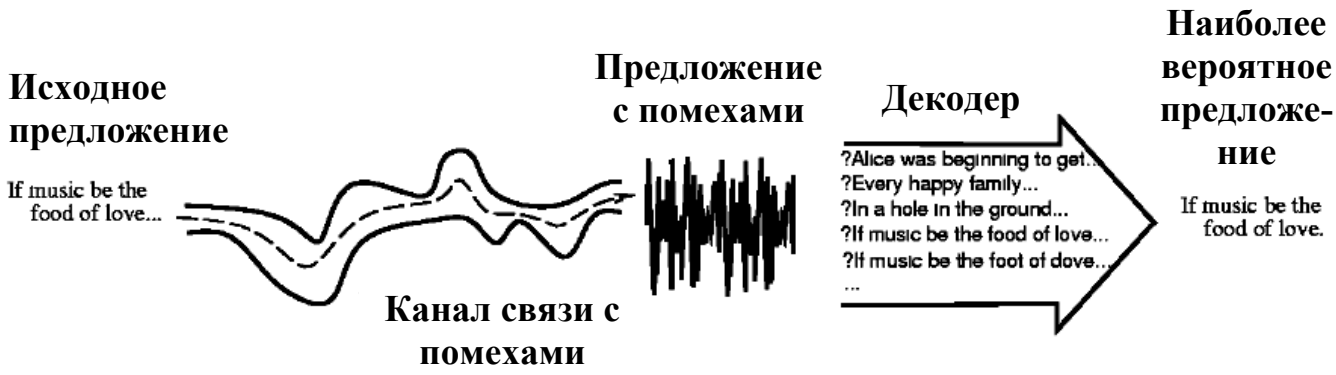

Рис. 8.1

Модель канала с помехами применяется ко всему предложению. Современные системы распознавания речи ищут предложения из огромного пространства потенциальных исходных предложений и выбирают предложение с максимальной вероятностью, генерируя предложения с шумом. Для этого они должны работать с моделями, которые выражают вероятность предложений, реализованных в виде определённой цепочки слов (N-грамм); моделями, которые выражают вероятность слов, реализованных в виде определённой цепочки фонов (скрытая марковская модель, СММ); и моделями, которые выражают вероятность фонов, реализованных в виде акустических или спектральных признаков (модель Гаусса или многослойный персептрон — искусственная нейронная сеть).

При использовании модели канала с помехами требуется решить 2 проблемы:

1. Чтобы выбрать предложение, более всего соответствующее сигналу с шумом, необходимо выбрать метрику для лучшего соответствия. Т.к. речь так разнообразна, входное акустическое предложение никогда не будет в точности соответствовать никакой модели для этого предложения.

2. Необходим эффективный алгоритм, который бы искал исходное предложение не среди всех возможных предложений, а среди тех, у которых есть шанс совпасть с исходным. Это проблема декодировки или проблема поиска. Для решения этой проблемы применяется 2 подхода:

- декодер Витерби или декодер динамического программирования;
- алгоритм эвристического поиска пути на графе решений A* (произносится «A со звёздочкой») или стековый декодер.

Цель такой архитектуры АРР можно сформулировать следующим образом: найти наиболее вероятное предложение из всех предложений языка L, соответствующее входному акустическому сигналу O.

Можно рассматривать акустический сигнал O как последовательность «символов» или «наблюдений» (8.2): например, если разбить весь сигнал на отрезки по 10 мс и представить каждой отрезок в виде вектора действительных чисел, соответствующих интенсивности или набору частот этого отрезка. Индекс каждого отрезка o_i соответствует смещению во времени:

$$O = o_1,o_2,o_3,\ldots,o_t \tag{8.3}$$

Аналогично, можно рассматривать предложения как последовательности слов:

$$W = w_1,w_2, w_3,... ,w_n \tag{8.4}$$

Оба предположения являются упрощениями. Например, разбить предложение на слова не всегда представляется возможным одним единственным способом: иногда требуется трактовать несколько слов как одну единицу, иногда — одно слово как несколько единиц. Обычно при распознавании речи за слово принимается словоформа (после приведения всех словоформ к нижнему регистру): слово *oak* будет отличаться от слова *oaks*, но вспомогательный глагол *can* («*can you tell me…*») будет идентичен существительному *can* («*I need a can of…*»). В последнее время, однако, разработчики более сложных систем АРР начали использовать информацию о морфемах и частях речи.

Тогда задачу АРР можно выразить в виде формулы (8.5):

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} P(W|O) \tag{8.5}$$

Выражение (8.5) гарантирует нахождение подходящего предложения *W*, однако, чтобы решить это уравнение, необходимо преобразовать его по методу Байеса (8.6).

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{8.6}$$

Подставив выражение (8.6) в формулу (8.5), получим (8.7):

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} \tag{8.7}$$

Посчитать вероятности в правой части формулы (8.7) намного проще, чем посчитать вероятность *P(W|O)*. Например, априорная вероятность цепочки слов, *P(W)*, считается с использованием n-gram языка, т.е. вероятностей следования слов друг за другом. И мы увидим далее, что посчитать *P(O|W)* также не сложно. Посчитать же вероятность последовательности акустических наблюдений, *P(O)*, довольно сложно. К счастью, мы можем проигнорировать знаменатель, т.к. предполагаем, что рассматриваются все возможные предложения языка, т.е. выражение $\frac{P(O|W)P(W)}{P(O)}$ будет вычисляться для каждого предложения, соответственно знаменатель будет оставаться тем же. Таким образом, мы получаем (8.8):

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} P(O|W)\,P(W) \tag{8.8}$$

Итак, чтобы найти наиболее вероятное предложение *W*, соответствующее последовательности наблюдений *O*, нужно посчитать произведение двух вероятностей для каждого предложения и выбрать предложение, значение произведения вероятностей для которого будет наибольшим.

Априорную вероятность *P(W)* в данном случае называют *моделью языка*, а условную вероятность *P(O|W) — акустической моделью*.

Чтобы посчитать акустическую модель *P(O|W)* предположим, что акустический сигнал *O* — это не последовательность акустических наблюдений, а последовательность фонов *F*.

На рис. 8.2 схематично показана архитектура АРР. Процесс распознавания речи разбивается на 3 этапа. На этапе обработки сигнала осциллограмма разбивается на равные отрезки (обычно по 10, 15 или 20 мс), и для каждого отрезка вычисляются его спектральные характеристики (насколько интенсивен сигнал на каждой частоте). На этапе распознавания фонов или частей слова используются такие статистические методы как искусственные нейронные сети (ИНС) или модель Гаусса для отнесения полученных характеристик к фонемам. Если используется ИНС, то выходом будет вектор вероятностей фонов (например, что данный отрезок соответствует фонеме [p] с вероятностью 0,8, фонеме [b] с вероятностью 0,1, фонеме [f] с вероятностью 0,02 и т.д). На последнем этапе используется лексикон произношений и модель языка (вероятностная грамматика). Для поиска подходящей последовательности слов, имеющей максимальную вероятность, используется алгоритм Витерби или декодер A*.

Рис. 8.2. Схематичная архитектура системы распознавания речи

Глава 9. Искусственные нейронные сети

9.1. Коннекционизм

В 80–90-е гг. XX в. в когнитивной науке был разработан и успешно применён новый компьютерный подход к созданию искусственных интеллектуальных систем — коннекционизм. Он позволил сконструировать принципиально новый тип компьютерных устройств, состоящих из искусственных нейронных сетей, которые обеспечили возможность моделировать некоторые процессы познания живых существ (включая человека) и их интеллектуальные способности. Оказалось, что искусственные нейронные сети, использующие принцип параллельной и распределённой обработки информации, с гораздо большей степенью адекватности воспроизводят выявленные нейробиологами механизмы функционирования мозга. Например, наличие в организации нейронов промежуточных, «скрытых» слоев, при участии которых происходит внутренняя переработка поступающих извне сигналов, способность определённым образом соединённых групп нейронов к постепенному изменению своих свойств по мере получения новой информации (т.е. обучению) и т.д. (Меркулов 2004а, с. 45–46).

По-видимому, функционирование компьютерных устройств, состоящих из искусственных нейронных сетей, во многих отношениях действительно напоминает работу нашего правополушарного, пространственно-образного мышления, ведь оно способно параллельно перерабатывать значительное число одновременно поступающих на «вход» единиц когнитивной информации (Меркулов 2004б, с. 248–251).

Из коннекционистских моделей и методов обучения сетей, в частности следует, что репрезентация когнитивной информации в мозгу скорее не локализована в отдельных нейронах или нейронных узлах, а распределена в системе. Обучение искусственных нейронных сетей показало, что каждая распределённая репрезентация является паттерном, действующим через все модули, так как граница между простыми и сложными репрезентациями отсутствует. Поскольку ни один индивидуальный модуль не кодирует какой-либо символ, распределённые репрезентации являются субсимвольными. Если, например, моделируется действие каждого нейрона с числом, то действие мозга в целом может быть тогда представлено как гигантский вектор (или список) чисел. И вход в мозг из сенсорных систем, и его выход к индивидуальным мышечным нейронам также могут быть обработаны как векторы того же самого типа (Меркулов 2004а, с. 47–48).

Лучше всего искусственные нейронные сети адаптированы к обработке информации, касающейся ассоциаций, к когнитивным проблемам, которые возникают в случае параллельно действующих противоречивых команд, — например, распознавание объектов, планирование, координирование движений, оценка тонких статистических паттернов, оперирование нечеткими понятиями и т.д. (Меркулов, 2004а, с. 47). Такие системы достаточно устойчивы и выдают приемлемый результат даже при получении искажённой информации.

Дискуссия между «традиционалистами» и «коннекционистами» неизбежно приводит к признанию необходимости разработки «гибридных» концепций, учитывающих специфику нейронных связей, распространяющейся по ним активации, роли параллельно протекающих процессов, взаимодействия процессов переработки «снизу вверх» и «сверху вниз», т.е. приводящих к взаимодействию субсимвольного и символического подходов. При обучении, например, иностранному языку, пребывание в естественной языковой среде облегчает настройку коннекционистских сетевых «фильтров», которые облегчают распознавание речи и её сегментацию, в то же время происходит использование правил уже изученного языка, перенос которых — как успешный, так и неуспешный — происходит в соответствии с традиционными моделями эксплицитного научения (Цепцов 1996, с. 146).

9.2. Как работает мозг

Нервная система и мозг человека состоят из нейронов, соединённых между собой нервными волокнами. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от нашей кожи, ушей и глаз к мозгу, процессы мышления и управления действиями — всё это реализовано в живом организме как передача электрических импульсов между нейронами.

Рассмотрим строение биологического нейрона (рис. 9.1). Каждый нейрон имеет отростки нервных волокон двух типов — *дендриты*, по которым принимаются импульсы, и единственный *аксон*, по которому нейрон может передавать импульс. Аксон контактирует с дендритами других нейронов через специальные образования — *синапсы*, которые влияют на силу импульса.

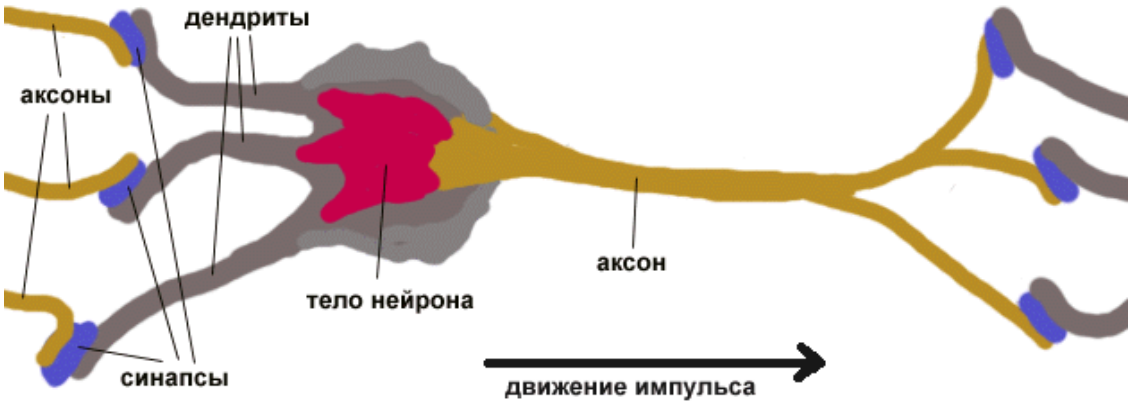

Рис. 9.1

Можно считать, что при прохождении синапса сила импульса меняется в определённое число раз, которое мы будем называть *весом синапса*. Импульсы, поступившие к нейрону одновременно по нескольким дендритам, суммируются. Если суммарный импульс превышает некоторый порог, нейрон возбуждается, формирует собственный импульс и передает его далее по аксону. Важно отметить,

что веса синапсов могут изменяться со временем, а значит, меняется и поведение соответствующего нейрона. Нетрудно построить математическую модель описанного процесса (Нейронные сети).

9.3. Математическая модель нейронной сети

Искусственные нейронные сети представляют собой устройства параллельных вычислений, состоящие из множества взаимодействующих простых процессоров. Такие процессоры обычно исключительно просты, особенно в сравнении с процессорами, используемыми в персональных компьютерах. Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам, и, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи. (Каллан 2001, с. 13)

Следует сделать замечание, что искусственные нейронные сети ориентированы на использование специального аппаратного обеспечения — нейрокомпьютеров, чтобы обеспечить параллельное вычисление состояний нескольких нейронов. В существующих IBM-совместимых компьютерах искусственные нейронные сети представляются с помощью специального программного обеспечения, а не в виде аппаратной реализации, вычисления в них производятся последовательно, что сокращает быстродействие системы.

Нейрон представляет собой единицу обработки информации в искусственной нейронной сети. Нейрон j может иметь несколько входов (дендритов), причём синапсы этих дендритов имеют веса w_{1j}, w_{2j}, w_{3j} … w_{nj} (рис. 9.2). Пусть к синапсам поступают импульсы силы x_1, x_2, x_3 … x_n соответственно, тогда после прохождения синапсов и дендритов к нейрону поступают импульсы $w_{1j}x_1$, $w_{2j}x_2$, $w_{3j}x_3$ … $w_{nj}x_n$. В отличие от синапсов мозга синаптический вес искусственного нейрона может иметь как положительные, так и отрицательные значения.

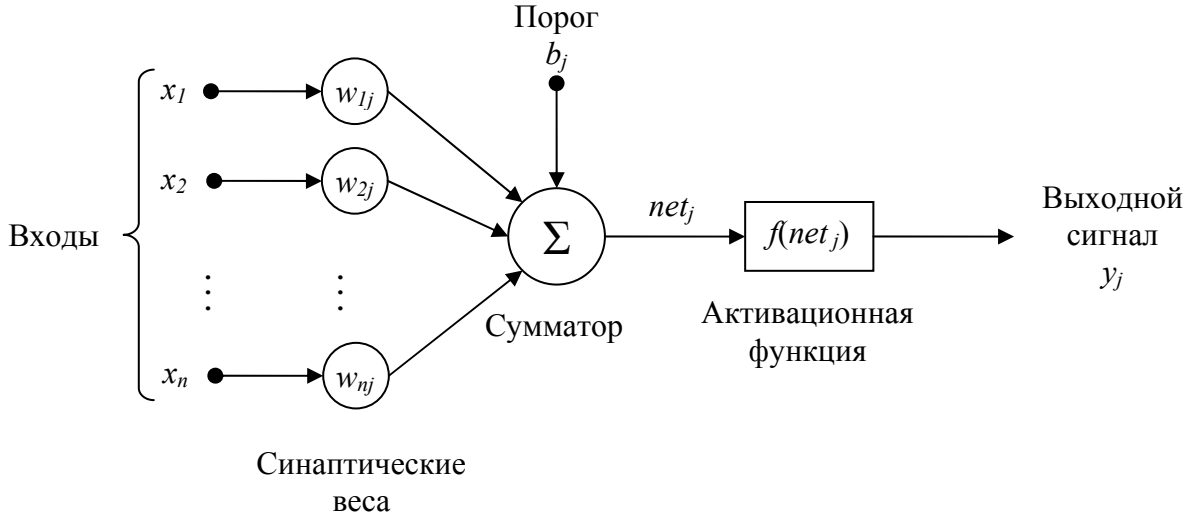

Рис. 9.2 Модель нейрона

Состояние нейрона определяется по формуле (9.1):

$$net_j = \sum_{i=1}^{n} w_{ij} x_i \,, \tag{9.1}$$

где
n — число входов нейрона;
x_i — значение i-го входа нейрона;
w_{ij} — вес i-го синапса.

Используются и другие формы комбинирования входящих импульсов, и другим часто встречающимся методом является рассмотрение квадрата разности между значением веса синапса и значением импульса, с последующим суммированием таких разностей для всех входов данного нейрона.

Затем определяется значение аксона нейрона (активность) по формуле (9.2):

$$y_j = f(net_j + b_j), \tag{9.2}$$

где f — некоторая *активационная* функция, которая ограничивает амплитуду выходного сигнала нейрона до определённого диапазона, поэтому эта функция называется также *функцией сжатия*. Обычно активность нейрона лежит в интервале [-1, 1] или [0, 1];

b_j — *пороговый элемент* (bias). Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации, и также как и синаптические веса изменяется во время обучения ИНС (Луканин 2010).

Наиболее часто в качестве активационной функции используется так называемый *сигмоид*, который имеет следующий вид:

$$f(net_j) = \frac{1}{1 + \exp(-net_j a)}. \tag{9.3}$$

При уменьшении параметра a сигмоид становится более пологим, вырождаясь в горизонтальную линию на уровне 0,5 при $a = 0$. При увеличении a сигмоид всё больше приближается к функции единичного скачка (рис. 9.3) (Стариков).

С точки зрения строения ИНС можно выделить три основных типа таких сетей (Павленко):

- *полносвязные* (рис. 9.4, а);
- *многослойные* или *слоистые* (рис. 9.4, б);
- *слабосвязные* (с локальными связями) (рис. 9.4, в).

При решении задач анализа с использованием нейронных сетей не предполагают использование какой-либо модели изучаемого объекта. Всё, что в данном случае необходимо, — конкретные факты поведения этой системы, содержащиеся в обучающем множестве. Тем не менее, одно простое, но фундаментальное предположение при этом всё-таки используется; это предположение о монотонности пространства решений, которое формулируется так (Некипелов): «Похожие входные ситуации приводят к похожим выходным реакциям системы» (Загоруйко, 1999).

Рис. 9.3

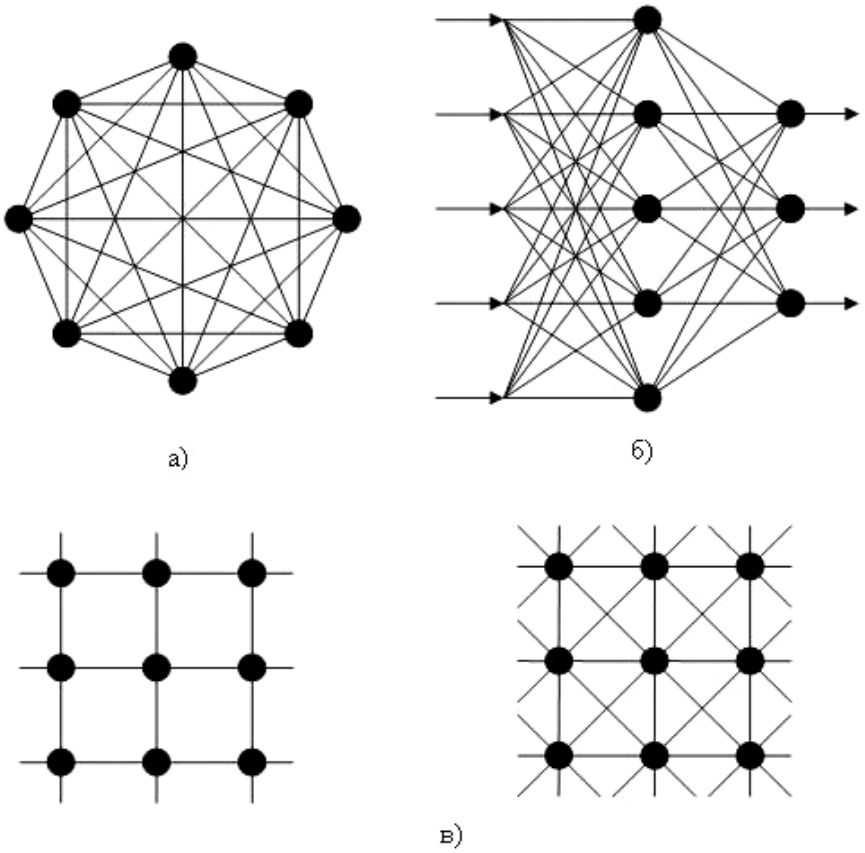

Рис. 9.4

«При таком подходе мы не пытаемся познать систему так глубоко, чтобы уметь предсказывать её реакцию на любые возможные внешние воздействия. Мы знаем лишь одно её фундаментальное свойство: монотонность поведения в окрестностях имеющихся прецедентов. И этого обычно оказывается достаточно для

получения практически приемлемых решений в каждом конкретном случае» (Загоруйко 1999).

9.4. Правило обучения, корректирующее связи

Одно из главных преимуществ нейронных сетей заключается в том, что они предполагают наличие правил, с помощью которых сеть может программироваться автоматически.

Типичной формой обучения является управляемое обучение, когда для каждого набора данных, подающегося в процессе обучения на вход сети, соответствующий выходной набор известен. Обычно в начале обучения весовые коэффициенты устанавливаются равными случайным малым значениям, так что в первый раз при предъявлении сети учебного образца оказывается весьма маловероятным, чтобы сеть произвела верный вывод. Расхождение между тем, что даст сеть, и тем, что для данного учебного набора должно быть получено на самом деле, составляет ошибку, которая может использоваться для корректировки весов. Примером правила коррекции ошибок является дельта-правило, называемое также правилом Видроу-Хоффа (Widrow-Hoff). Если выходной элемент имеет активность (т.е. вывод) y, а истинный вывод должен быть равным t, то ошибка δ задаётся следующей формулой (9.4):

$$\delta = t - y. \tag{9.4}$$

Сигнал, приходящий к выходному элементу, обозначен через x. В соответствии с дельта-правилом, необходимо внести коррекцию Δw, вычисляемую по формуле (9.5):

$$\Delta w = \eta \delta x, \tag{9.5}$$

где η обозначает действительное число, называемое *нормой обучения.* Новый весовой коэффициент устанавливается равным сумме значений старого веса и коррекции:

$$w = w + \Delta w\,. \tag{9.6}$$

В начале обучения весовые коэффициенты устанавливаются равными малым случайным значениям, например, из диапазона [-0.3, +0.3]. В процессе обучения на вход сети подаются образец за образцом, и в результате их обработки весовые коэффициенты корректируются до тех пор, пока для всех вводимых образцов ошибки не станут меньше некоторого приемлемого достаточно малого значения. В завершение процесса сеть тестируется на данных, не представленных в фазе обучения: в результате можно оценить, насколько хорошо сеть работает с данными, которые в процессе обучения были ей неизвестны.

Данные, используемые для обучения нейронной сети, обычно разделяются на две категории: одни данные используются для обучения, а другие — для тестирования. На самом деле реальные качества нейронной сети выявляются только во время тестирования, поскольку успешное завершение обучения сети должно оз-

начать отсутствие признаков неправильной работы сети во время её тестирования. Процесс тестирования разрабатывается так, чтобы в его ходе для данной сети можно было бы оценить её способность обобщать полученные знания. Обобщение в данном случае означает способность сети правильно выполнять задачу с данными, которые оказываются хотя и аналогичными данным, предъявлявшимся сети в процессе обучения, но всё же отличными от них (Калан 2001, с. 25–28).

Вопросы к семинару

1. Архитектура сети NETtalk, особенности кодирования букв и фонем.
2. Алгоритм обучения и исходные данные для обучения.
3. Отличия и сходства обучения сети NETtalk и развития речи у детей.
4. Семантико-синтаксическое представление информации с помощью простых рекуррентных сетей.
5. Автоматическая рубрикация документов с использованием самоорганизующихся карт Кохонена. Формализмы WebSOM и TopSOM.
6. Автоматическая обработка ЕЯ с использованием субсимвольных нейронных сетей.

Литература к семинару

1. Плешко, В.В. TopSOM: визуализация информационных массивов с применением самоорганизующихся тематических карт / В.В. Плешко, А.Е. Ермаков, Г.В. Липинский // Информационные технологии. — 2001. — № 8. — С. 8–11.
2. Browsing digital libraries with the aid of self-organizing maps / K. Lagus, S. Kaski, T. Honkela, T. Kohonen // *Proceedings of the Fifth International World Wide Web Conference WWW5*, May 6–10, Paris, France, volume Poster Proceedings, 1996. — P. 71–79.
3. Miikkulainen, R. Natural Language Processing with Subsymbolic Neural Networks / Risto Mikkulainen // A. Browne (editor), Neural Network Perspectives on Cognition and Adaptive Robotics. Institute of Physics Publishing, 1997.
4. Sejnowski, T.J. NETtalk: a parallel network that learns to read aloud / T.J. Sejnowski, C.R. Rosenberg, Cognitive Science, 1986. — 14. — P. 179–211.
5. Stolcke, A. Learning Feature-based Semantics with Simple Recurrent Networks / Andreas Stolcke. — ICSI Technical Report tr-90-015, April 1990. — ftp://ftp.icsi.berkeley.edu/pub/techreports/1990/tr-90-015.pdf

Библиографический список

1. Беляева, Л.Н. Автоматический (машинный) перевод / Л.Н. Беляева, М.И. Откупщикова // Прикладное языкознание: учебник. — СПб.: Изд-во С.-Петербург. ун-та, 1996. — С. 360–388.
2. Богданов, В.В., Реферирование / В.В. Богданов // Прикладное языкознание: учебник. — СПб.: Изд-во С.-Петербург. ун-та, 1996. — С. 389–398.
3. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. — СПб.: Питер, 2000.
4. Завгородний, В.И. Комплексная защита информации в компьютерных системах: учебное пособие / В.И. Завгородний. — М.: Логос; ПБОЮЛ Н.А. Егоров, 2001. — С. 133–139.
5. Загоруйко, Н.Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко. — Новосибирск: Изд-во Института математики, 1999.
6. Каллан, Р. Основные концепции нейронных сетей: пер. с англ. / Р. Каллан. — М.: Издательский дом «Вильямс», 2001.
7. Кодзасов, С.В. Общая фонетика / С.В. Кодзасов, О.Ф. Кривнова. — М.: РГГУ, 2001. — 592 с.
8. Левин, М. Криптография: Руководство пользователя / М. Левин. — М.: Познавательная книга плюс, 2001. — С. 3–4, 84–115.
9. Левин, М. E-mail «безопасная»: Взлом, «спам» и хакерские атаки на системы электронной почты в Интернет / М. Левин. — М.: Майор, 2002. — С.138–155.
10. Леонов, В.П. О методах автоматического реферирования (США, 1958–1974) / В.П. Леонов // Научно-техническая информация. Сер. 2. — 1975. — № 6.
11. Луканин, А.В. Инструментарий прикладного лингвиста / А.В. Луканин // Современные направления прикладной лингвистики: материалы I Студенческой научно-практической конференции, Челябинск, 19 апреля 2008 г. — С. 4–13. — http://ol.susu.ru/konf_tpl2008.html.
12. Луканин, А.В. Лингвистическая синергетика и нейросетевой подход (по поводу книги Р.Г. Пиотровского) / А.В. Луканин // НТИ Серия 2. — 2008. — №3. — С. 30–33.
13. Луканин, А.В. Нейросетевой и фреймовый подходы к компьютерной обработке лингвистических сетей // Памяти проф. Р.Г. Пиотровского: межвуз. сб. Вып. 9 / под ред. М.А. Марусенко. — СПб.: Филологический факультет СПбГУ, 2010. — (Серия «Древняя и Новая Романия»). — С. 129–144.
14. Меркулов, И.П. Когнитивная модель сознания / И.П. Меркулов // Эволюция. Мышление. Сознание. (Когнитивный подход и эпистемология). — М.: Канон+, 2004. — С. 35–64.
15. Меркулов, И.П. Мышление как информационный процесс / И.П. Меркулов // Эволюция. Мышление. Сознание. (Когнитивный подход и эпистемология). — М.: Канон+, 2004. — С. 228–260.
16. Минский, М. Фреймы для представления знаний: пер. с англ. / М. Минский. — М.: Энергия, 1979. — 152 с.

17. Михайлов, А.И. Научные коммуникации и информатика / А.И. Михайлов, А.И. Черный, Р.С. Гиляревский. — М.: Наука, 1970.
18. Нейронные сети. — http://www.uran.donetsk.ua/~masters/2001/kita/baginsky/diss/libr.htm
19. Некипелов, Н. Введение в RBF сети / Н. Некипелов. — http://www.basegroup.ru/neural/rbf.htm.
20. Основы криптографии: учебное пособие / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин. — М.: Гелиос АРВ, 2001. — С.8–54.
21. Павленко, М. Нервные сети / М. Павленко // Мой компьютер. — 2005. — № 7(334). — http://mycomp.com.ua/text/8465
22. Петров, А.А. Компьютерная безопасность. Криптографические методы защиты / А.А. Петров. — М.: ДМК, 2000. — С.23–28.
23. Пиотровский, Р.Г. Лингвистический автомат (в исследовании и непрерывном обучении): 2-е доп. и испр. изд. / Р.Г. Пиотровский. — СПб.: Изд-во РГПУ им. А.И. Герцена, 2008. — 256 с. — http://www.silod.org/txt/text2.zip
24. Пиотровский, Р.Г. Лингвистическая синергетика: исходные положения, первые результаты, перспективы / Р.Г. Пиотровский. — СПб.: Филологический факультет СПбГУ, 2006. — 160 с.
25. Рекомендации по подготовке исходного текста. — http://www.translate.ru/tips.asp?lang=ru
26. Стариков, А. Нейронные сети — математический аппарат / А. Стариков. — http://www.basegroup.ru/neural/math.htm
27. Столлингс, В. Криптография и защита сетей: принципы и практика: пер. с англ., 2-е изд. / В. Столлингс. — М.: Издательский дом «Вильямс», 2001. — С. 48–80, 177–213.
28. Хан, У. Системы автоматического реферирования / У. Хан, И. Мани // Открытые системы. — 2000. — № 12. — http://www.osp.ru/os/2000/12/178370/
29. Цепцов, В.А. От критики коннекционизма к гибридным системам обработки информации / В.А. Цепцов // Познание. Общество. Развитие / РАН. Ин-т психологии; ред. и сост. Д.В. Ушаков. — М., 1996.
30. Antworth, E.L. PC-KIMMO: a two-level processor for morphological analysis / E.L. Antworth // Occasional Publications in Academic Computing. — Dallas, TX: Summer Institute of Linguistics, 1990. — No. 16.
31. Bauer, L. English word-formation / L. Bauer. — Cambridge: Cambridge University Press, 1983.
32. Dutoit, T. High-Quality Text-to-Speech Synthesis : an Overview / T. Dutoit // Journal of Electrical & Electronics Engineering, Australia: Special Issue on Speech Recognition and Synthesis. — 1997. — vol. 17 n°1. — P. 25–37. — http://tcts.fpms.ac.be/publications/regpapers/1997/ieeea97_td.zip
33. Jurafsky, D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition / D. Jurafsky, J.H. Martin. — New Jersey: Prentice Hall, 2000.
34. Klatt, D.H. Software for a cascade / parallel formant synthesizer / D.H. Klatt // J. Accoust. Soc. AM. — 1980. — Vol 67. — P. 971–995.

35. Kleene, S.C. Representation of events in nerve nets and finite automata. Tech. rep. RM-704 / S.C. Kleene. — RAND Corporation. RAND Research Memorandum. — 1951. — 101 p.
36. Kleene, S.C. Representation of events in nerve nets and finite automata / S.C. Kleene //Automata Studies / C. Shannon, J. McCarthy (Eds.). — Princeton, NJ: Princeton University Press, 1956. — P. 3–41.
37. Kučera, H. Computational analysis of present-day American English / H. Kučera, W.N. Francis. — Providence, RI: Brown University Press, 1967.
38. Labov, W. Contraction, deletion, and inherent variability of the English copula / W. Labov // Language. — 1969. — № 45(4). — P. 715–762.
39. McCulloch, W.S. A logical calculus of ideas immanent in nervous activity / W.S. McCulloch, W. Pitts // Bulletin of Mathematical Biophysics. — 1943. — P. 115–133. (Русский перевод: Маккаллок, У.С. Логическое исчисление идей, относящихся к нервной деятельности. Автоматы / У.С. Маккаллок, У. Питтс. — М.: ИЛ, 1956).
40. Porter, M.F. An algorithm for suffix stripping / M.F. Porter // Program. — 1980. — № 14(4). — P. 130–137.
41. Rabin, M.O. Finite automata and their decision problems / M.O. Rabin, D. Scott // IBM Journal of Research and Development. — 1959. — 3(2). — P. 114–125.
42. Saggion, H. Видео-курс Hands-on Natural Language Processing for Information Access Applications / H. Saggion // RuSSIR'2008. — http://videolectures.net/russir08_saggion_nlpiaa/
43. Sproat, R. Morphology and Computation / R. Sproat. — Cambridge: MIT Press, 1993.
44. Torrejón, E. Controlled Translation: A New Teaching Scenario Tailor-made for the Translation Industry / E. Torrejón, C. Rico // 6th EAMT Workshop Teaching Machine Translation. — Manchester, 2002. — P. 107–116. — http://www.mt-archive.info/EAMT-2002-Torrejon.pdf
45. Turing, A.M. On Computable Numbers, with an Application to the Entscheidungsproblem / A.M. Turing // Proceedings of the London Mathematical Society. — 1937. — 2(42). — P. 230–265.

Оглавление

Глава 1. Системы автоматической обработки текста и лингвистический автомат 3
1.1. Лингвистический автомат ... 3
1.2. Человеко-машинная синергетика и самоорганизация в системах АОТ ... 4
1.3. Уровневое построение систем АОТ и ЛА ... 5
1.4. Опознание языка документа и нормализация входного текста ... 5
1.5. Подблок опознания формата текста и его частей, а также определение их жанровой и тематической принадлежности ... 6
1.6. Средства обеспечения живучести ЛА ... 8
Глава 2. Фреймы в системах автоматической обработки текстов ... 9
2.1. Основные определения ... 9
2.2. Типы и структура фреймов ... 10
2.3. Применение фреймов в системах автоматической обработки текстов ... 11
Глава 3. Подъязыки в системах машинного перевода ... 14
3.1. Подъязыки технической документации ... 14
3.2. Предредактирование ... 15
3.3. Постредактирование ... 17
Глава 4. Теория конечных автоматов ... 18
4.1. Моделирование овечьего языка с помощью конечного автомата ... 19
4.2. Формальные языки ... 20
4.3. Недетерминированные конечные автоматы ... 21
4.4. Распознавание как поиск ... 23
4.5. Построение лексикона на основе конечного автомата ... 24
Глава 5. Системы аннотирования и реферирования ... 28
5.1. Подходы и методы автоматического реферирования ... 29
5.2. Экстрактивные методы ... 29
5.2.1. Статистические методы ... 31
5.2.2. Логико-математические методы ... 33
5.2.3. Дистрибутивный метод ... 34
5.2.4. Метод содержательных аспектов ... 35
5.2.5. Метод текстовых связей ... 36
5.3. Методы с опорой на знания ... 36
Глава 6. Криптография ... 38
6.1. Основные понятия ... 38
6.2. Немного из современной истории криптографии ... 39
6.3. Классификация методов криптографической обработки информации ... 41
6.4. Вскрытие шифров перестановки ... 42
Глава 7. Системы автоматического синтеза речи ... 45
7.1. Сферы применения АСР ... 45
7.2. Устройство системы АСР ... 46

7.2.1. Модуль обработки естественного языка ... 46
7.2.2. Анализ текста ... 47
7.2.3. Автоматическая транскрипция ... 47
7.3. Модуль обработки цифрового сигнала ... 49
7.3.1. Акустический синтез по правилам ... 49
7.3.2. Акустический синтез на основе конкатенации ... 50
Глава 8. Системы автоматического распознавания речи ... 53
8.1. Метод Байеса ... 53
8.2. Архитектура системы APP ... 55
Глава 9. Искусственные нейронные сети ... 59
9.1. Коннекционизм ... 59
9.2. Как работает мозг ... 60
9.3. Математическая модель нейронной сети ... 61
9.4. Правило обучения, корректирующее связи ... 64
Библиографический список ... 66