ГАОУ ВО «Дагестанский государственный университет народного хозяйства»

.

Рашидова Зарема Джаруллаховна

Кафедра информатики

«Информационные технологии в лингвистике»

(курс лекций)

Махачкала – 2018

Составитель – Рашидова Зарема Джаруллаховна, старший преподаватель кафедры информатики ДГУНХ.

Внутренний рецензент – Гереева Тату Рашидовна**,** кандидат экономических наук, доцент кафедры «Прикладная математика и информационные технологии» ДГУНХ.

Внешний рецензент – Везиров Тимур Гаджиевич, доктор педагогических наук, профессор кафедры методики преподавания математики и информатики Дагестанского государственного педагогического университета.

Рашидова З. Д., Курс лекций по дисциплине «Информационные технологии в лингвистике»– Махачкала: ДГУНХ , 2018.- 242с.

Печатается по решению Учебно-методического совета Дагестанского государственного университета народного хозяйства.

Оглавление

Аннотация

Дисциплина предназначена для студентов факультета иностранных языков, обучающихся по направлению подготовки 035700 «Лингвистика». Лингвистика является одной из самых компьютеризированных и математизированных гуманитарных наук, что подтверждается наличием таких областей языкознания, как математическая(комбинаторная и квантитативная) лингвистика, лингвостатистика, лексикостатистика, компьютерная лингвистика. Современные лингвистические исследования требуют применения информационных технологий и математических методов обработки информации для выявления сущностей лингвистических явлений. Основываясь на положениях «математика является универсальным языком науки и мощным средством решения прикладных задач» и «математика и информатика должны работать на профессиональную подготовку будущего специалиста, быть ее органичной частью», необходимо рассматривать информационные технологии лингвистическим компонентом профессионального образования лингвистов. Таким образом, математическая и информационно-технологическая подготовка лингвистов является актуальнейшей задачей современного лингвистического образования.

В рамках курса студенты должны уметь соотнести понятийный аппарат дисциплин теоретической и прикладной лингвистики с реальными фактами и явлениями профессиональной деятельности, уметь использовать теоретические положения для решения практических профессиональных задач.

Целями изучения дисциплины являются:

- получение студентами знаний, умений соотношения понятийного аппарата дисциплин теоретической и прикладной лингвистики с реальными фактами и явлениями профессиональной деятельности, умения использовать

теоретические положения для решения практических профессиональных задач.

- формирование представления о современных методах получения, обработки и хранения информации;
- формирование представления о применении современных информационных технологий в языкознании и лингвистическом анализе;
- формирование у студентов представления о принципах построения математических моделей обработки информации и о границах применимости компьютерных и количественных методов в лингвистике и филологии;
- формирование понимания сущности математической обработки информации в гуманитарных исследованиях и умений применения на практике ряда количественных методов, получивших признание в гуманитарных исследованиях;
- ознакомление с достижениями и возможными перспективами «математизации и компьютеризации» теоретического и прикладного языкознания.

В результате изучения дисциплины студенты должны:

1. Приобрести способность распознавать различные виды информационных систем и технологий;
2. Понимать основные возможности и тенденции развития информационных технологий и систем;
3. Иметь навыки работы в рамках отдельных информационных технологий;
4. Иметь способность применять полученные знания для решения типовых задач выбора и применения информационных технологий и систем;
5. Понимать основные концепции управления информационными системами и технологиями и уметь применять их на практике;
6. Иметь навыки проектирования базы данных к конкретной информационной системе.

Основными объектами изучения в курсе являются:

- история и становление компьютерной лингвистики как научного направления;

- основные проблемы и задачи, решаемые КЛ;

- основные направления исследований;

- теоретические и практические результаты взаимодействия и взаимовлияния информатики и лингвистики;

В результате обучения у студентов должны быть сформированы следующие представления:

- о месте компьютерной лингвистики в кругу других лингвистических дисциплин;

- о языке как основном средстве передачи информации и его функциях;

- о применении теоретических исследований языка в прикладных разработках: в сфере компьютерного моделирования и анализа текста; в сфере "обработки естественного языка" и машинного перевода; в сфере компьютерной лексикографии и корпусной лингвистики;

- о конкретных разработках в названных выше областях;

Студенты должны выработать навыки:

- работы с базовыми для филологического исследования программными продуктами;

- применения существующих программных продуктов в лингвистическом (филологическом) исследовании;

- использование ресурсов Интернет (теоретических и прикладных) в лингвистическом исследовании;

- использования новых источников материала в лингвистическом исследовании;

- применения новых методик сбора и обработки материала.

Тема 1. Основные понятия прикладной лингвистики.

Лингвистические модели.

1.Сущность прикладной лингвистики как особого подхода к языковым явлениям.

2. Различные наименования области прикладной лингвистики. Универсальные прикладные проблемы.

3. Типы лингвистических моделей и экспериментальные методы в лингвистике. Общенаучный метод моделирования и специфика его применения в лингвистике.

1.Сущность прикладной лингвистики как особого подхода к языковым явлениям.

Информационно-коммуникационные технологии не являются новыми в лингвистической науке. Целая область языкознания – прикладная лингвистика – оформилась как естественное «приложение» знаний о законах функционирования и эволюции языка к решению проблем других наук и практической деятельности. У прикладной лингвистики в создании и применении информационно-коммуникационных технологий есть собственная сфера.

В языкознании всегда присутствовали три глобальных исследовательских направления:

- теоретическое (объяснение языковых систем и процессов)
- описательное (конкретное описание языковых явлений)
- прикладное (совершенствование языковой системы).

В рамках последнего направления сформировалась научная дисциплина, которая получила название прикладной лингвистики. Ее отличает подход к языку как к деятельности, а не мертвому продукту.

Прикладная лингвистика (ПЛ)- это комплексная научная дисциплина, изучающая язык в различных ситуациях его применения и разрабатывающая методы совершенствования языковых систем и языковых

процессов, учение о методах решения разнообразных практических задач с использованием знаний о языке учение о совершенствовании языковой способности человека и общества в целом. Термин прикладная лингвистика появился в конце 20 гг. 20 в., когда была осознана необходимость строгого научного решения прикладных задач с использованием методов формального лингвистического анализа письменных и акустико-лингвистического анализа устных сообщений.

За рубежом под прикладной лингвистикой обычно понимают самую традиционную область применения лингвистической науки: совершенствование методов преподавания языка (лингводидактику). В отечественной традиции прикладная лингвистика – это компьютерная лингвистика, т. е. дисциплина, разрабатывающая лингвистические аспекты компьютеризации, прежде всего – информационно-коммуникационные технологии.

Основные задачи ПЛ:

1. автоматическое распознавание и синтез речи
2. автоматические методы переработки текстовой информации
3. создание автоматизированных систем информационного поиска
4. составление автоматических словарей и систем машинного перевода
5. разработка методов автоматического аннотирования, реферирования и перевода
6. разработка экспертных систем
7. лингвистическое обеспечение АСУ(автоматизированные системы управления)
8. стандартизация научно-технической терминологии

Хотя в качестве отдельной научной дисциплины прикладная лингвистика существует с 20-х годов прошлого века, стимулом для ее

бурного развития стало создание искусственного интеллекта и формирование всемирной паутины – сети Интернет.

С появлением электронно-вычислительных машин, роботов и компьютеров область приложения лингвистики существенно расширяется.

Во-первых, лингвисты вместе со специалистами по теории информации занимаются решением проблемы автоматического распознавания речи и интерпретации содержания речевого сообщения. Со сложностью распознавания речи каждый сталкивался в случае телефонного разговора в условиях помех, когда мы вынуждены переспрашивать, строить догадки о том, что слышим. Каждое речевое сообщение можно интерпретировать по-разному. Количество возможных интерпретаций намеренно сужают, когда пишут инструкции по применению, юридические акты или другие документы.

Во-вторых, при создании систем искусственного интеллекта необходимы сведения о способах воплощения знания средствами национального языка, сведения о вербализуемых (т. е. поддающихся передаче с помощью речи) и невербализуемых знаниях. Примером невербализуемых – или трудно вербализуемых – знаний являются схемы и карты. Также необходимы алгоритмы общения человека с системой, которые выстраивают на основе изучения речевой коммуникации. Разумеется, решением такого рода проблем занимаются лингвисты.

В-третьих, решение проблемы анализа и синтеза речи системами искусственного интеллекта невозможно без знания акустических характеристик речевого сигнала. Акустические характеристики каждого звука речи уникальны, поскольку любой звук мы можем произнести вслух или шепотом, любой звук произносят мужчины, женщины и дети. Физические характеристики – частота основного тона, сила, спектр – каждый раз меняются. Лингвисты, работающие в области фонетики, установили, что для каждой единицы языка фонетического уровня специфичны не конкретные характеристики, а соотношение частот колебания воздуха в

полостях глотки и рта при произнесении. В систему искусственного интеллекта необходимо ввести «портреты» языковых единиц, на которые она будет ориентироваться при синтезе речи. При этом каждый «портрет» получается довольно расплывчатым, диффузным; «портреты» перетекают один в другой, как черты лица родственников. Словом, задача не из легких. И поэтому до сих пор не найдено ее удовлетворительное решение.

В-четвертых, лингвистические знания нужны для разработки систем тестирования каналов связи. Между прочим, знаменитый петербургский фонетист Лев Рафаилович Зиндер разработал речевой материал, который использовал для тестирования системы связи Юрий Гагарин во время первого в истории человечества космического полета. В-пятых, интеллектуальные системы необходимо «научить» идентифицировать говорящего, т. е. установить его пол, возраст, наличие или отсутствие акцента, его эмоциональное состояние и т. п. Все это весьма существенно для криминалистики и охранных систем.

В-шестых, интеллектуальные системы должны «уметь» распознавать графические образы – рукописный текст, автограф на банковской карте и т. п. Эта задача предполагает не только инвентаризацию способов написания каждой буквы, но и описание закономерностей сочетания букв в словах.

Не будем продолжать подробный перечень задач, стоящих перед прикладной лингвистикой. В заключение упомянем только о необходимости решения проблемы хранения и переработки больших текстовых массивов.

Как видим, можно выделить такие основные сферы: традиционная методика, перевод (машинный перевод и распознавание речи на различных этапах овладения иностранным языком), речевая коммуникация, обеспечение надежности систем связи, общение с разными системами искусственного интеллекта. Добавим: в традиционной методике появилась такая форма работы, как тест, – обучающие тренажеры, контрольные работы, итоговые экзамены. Для формализации знаний, необходимых для создания теста, обеспечения процедуры тестирования и проверки результатов требуются

специалисты в прикладной лингвистике. Однако наиболее актуальная сфера применения лингвистических знаний – общение с разными системами искусственного интеллекта. Необходимо обеспечивать диалог компьютера с его пользователем, во-первых, и подготовить тексты на естественных языках для хранения, обработки и распространения информации при помощи телекоммуникаций, во-вторых.

Диалог подразумевает способность машины не только «понимать» команды на естественном языке, но и синтезировать собственные сообщения при ответе на запросы пользователя. Разумеется, речь не идет о том, чтобы «разговаривать» с компьютером по-русски или по-английски (по-польски, по-французски, по-грузински…), хотя уже существуют и такие программы. Огромную прикладную ценность имеют программы смысловой обработки сообщений, установления смысловых связей между различными сообщениями, определения актуальной для пользователя информации и оптимального режима взаимодействия с пользователем. Собственно говоря, каждый, кто пытался разыскивать необходимую информацию в Интернете, работал в системе Windows, представляет, как важно обеспечить «взаимопонимание» человека и компьютера. В создании программ, обеспечивающих диалог с компьютером, принимают участие специалисты в области прикладной лингвистики.

В сети Интернет широко представлена информация о прикладной лингвистике: вузах и кафедрах, ученых и лабораториях, специализированных изданиях. Популярность изданий обусловлена необходимостью применять достижения современной науки о языке для решения практических задач. При поддержке Российского Фонда Фундаментальных Исследований создан электронный журнал «Прикладная и экспериментальная лингвистика».

2. Различные наименования области прикладной лингвистики. Универсальные прикладные проблемы.

исетевой журнал «Кругозор»), в которых доступным языком на простых примерах излагаются как основные положения, так и проблемы науки. Однако понятие прикладная лингвистика шире, чем компьютерная.

Прикладная лингвистика - это комплексная научная дисциплина, изучающая язык в различных ситуациях его применения и разрабатывающая методы совершенствования языковых систем и языковых процессов.

Лингвистика входит в ядро складывающегося в настоящее время комплекса когнитивных наук, объединяемых по их интересу к проблемам организации, представления, обработки и использования знаний.

Синонимы ПЛ:

- *Компьютерная лингвистика* (машинная лингвистика) - дисциплина, которая разрабатывает лингвистические аспекты компьютеризации.
- *Вычислительная лингвистика*

Термин к*омпьютерная лингвистика* шире термина *вычислительная лингвистика,* так как задает общую ориентацию на использование компьютеров для решения разнообразных научных и практических задач, никак не ограничивая способы решения этих задач. Термин же вычислительная лингвистика может пониматься более узко, так как даже при широкой трактовке понятия вычисление за его пределами остаются такие стороны решения лингвистических задач, как, например, *представление знаний, организация банков языковых данных, психолингвистические аспекты взаимодействия человека и компьютера* и др. Т. о. можно считать, что термин компьютерная лингвистика (по своей внутренней форме) шире, чем вычислительная лингвистика. Английский эквивалент computational linguistics может переводиться и как компьютерный и как вычислительный (как и русском компьютер - синоним ЭВМ).

- *Структурная лингвистика*

Термин структурная лингвистика – это совокупность взглядов на язык и методов его исследования, в основе которых лежит понимание языка как знаковой системы с четко выделенными структурными элементами (единицами языка, их классами и пр.) и стремление к строгому (как в точных науках) формальному описанию языку. Свое название СЛ получила благодаря особому вниманию к структуре языка, которая представляет собой сеть отношений (противопоставлений) между элементами языковой системы, упорядоченных и находящихся в иерархической зависимости в пределах определенных уровней. Структурное описание языка предполагает такой анализ реального текста, который позволяет выделить обобщенные инвариантные единицы (схемы предложений, морфемы, фонемы) и соотнести их с конкретными речевыми сегментами на основе строгих правил реализации Эти правила определяют границы допустимого варьирования языковых единиц в речи. В зависимости от уровня анализа правила реализации формулируются как правила позиционного распределения конкретных, например, принцип дополнительной дистрибуции в фонологии и морфологии (дистрибутивный анализ), или как трансформационные правила в синтаксисе (при трансформационном анализе) регулирующие переход от инвариантной глубинной структуры предложения к множеству ее реализации. На базе СЛ развилась *порождающая грамматика (генеративная лингвистика*); идеи структурного анализа во многом определили постановку и решение задач, связанных с *машинным переводом*; СЛ открыла дорогу для широкого проникновения в лингвистику мат. методов (*математическая лингвистика*). На СЛ оказали влияние: Сепир, Блумфилд. Ф.де Соссюр, один из создателей и ведущих теоретиков - Якобсон; у нас - Реформатский (знаковая теория языка), Ревзин (общая теория моделирования), Холодович; практическое применение методов СЛ: Апресян, Арутюнова, Гак, Зализняк, Звегинцев, Мельчук, Успенский и др.

- *Математическая лингвистика* - математическая дисциплина, предметом которой является разработка формального аппарата для описания

строения естественных и некоторых искусственных языков. Возникла в 50 годы 20 в.; одним из главных стимулов появления математической лингвистики послужила назревшая потребность в уточнения основных лингвистических понятий. Методы МЛ имеют много общего с с методами мат. логики - мат. дисциплины, занимающейся изучением строения мат. рассуждений, - и в особенности таких ее разделов, как теория алгоритмов и теория автоматов.

- *Контрастивная лингвистика* (сопоставительная лингвистика) – сопоставительное изучение двух, реже нескольких языков для выявления их сходств и различий на всех уровнях языковой структуры с целью типологической классификации языков. Как правило, контрастивная лингвистика оперирует материалами на синхронном срезе языка. КЛ появилась и интенсивно развивалась в 50 гг. 20 в., однако ее появление подготовили работы Е.Д. Поливанова, Бодуена де Куртенэ, Л.В. Щербы с изложением теоретических основ сравнения родного и ин. языков. В 70 гг. контрастивные исследования в отдельных странах (гл. образом в США) использовали порождающую модель Хомского, с возведением явлений двух сопоставляемых языков к общей глубинной структуре; в наст. время наблюдается отход от этой методики в пользу сруктурно-функционального подхода.

Автоматический перевод - выполняемое на компьютере действие по преобразованию текста на одном ЕЯ в эквивалентный по содержанию текст на другом языке.

Универсальные прикладные проблемы:

- создание и совершенствование алфавитов и письменности (решена полностью и успешно) 3 стадии: (1) появление письменности, (2) книгопечатание, (3) компьютеризация.
- создание систем транскрипции устной речи, систем транслитерации иноязычных слов

- составление словарей (лексикография) (первые словари - глоссарии - комментарии к церковным текстам) составление автоматических словарей, тезаурусов
- унификация и стандартизация научно-технической терминологии
- изучение процессов и создание правил образования новых названий изделий, товаров и т.п.
- устный и письменный перевод, разработка систем машинного перевода, АРМов
- обучение родному и иностранным языкам, разработка соответствующих методик (обучение детей и взрослых, обучение эмигрантов, ...)
- создание и совершенствование ИЯ для записи информации
- автоматическое распознавание и синтез речи
- автоматические методы переработки текстовой информации
- создание автоматизированных систем информационного поиска
- составление автоматических словарей и систем машинного перевода
- разработка методов автоматического аннотирования, реферирования и перевода
- разработка экспертных систем
- лингвистическое обеспечение АСУ
- проблемы языка и пола (politically correct non-sexist language)
- создание систем стенографии, систем письма для слепых
- лечение речевых расстройств
- анализ дискурса

3. Типы лингвистических моделей и экспериментальные методы в лингвистике. Общенаучный метод моделирования и специфика его применения в лингвистике

Метод моделирования - центральный исследовательский метод в науке.

Моделирование в науке - это выяснение свойств какого-либо предмета при помощи построения его модели.

Моделью можно назвать образ какого-либо объекта, используемый в определенных условиях в качестве его заместителя (фотография в паспорте - модель человека).

Свойства моделей:

- условность
- образ может быть не только материальным, но и мысленным и передаваться посредством знаковой системы
- моделью может быть не только образ, но и прообраз оригинала
- модель чаще всего является *гомоморфной* оригиналу (то есть многим элементам оригинала соответствует меньшее количество элементов модели в отличие от изоморфизма).

Собственно лингвистические модели:

- модели речевой деятельности, процессуальный модели (самые сложные)
- модели языковой системы, языковой структуры (тоже очень сложные)
- модель памяти и др.

Лингвистическое моделирование необходимо предполагает использование *абстракции и идеализации*. Отображая релевантные существенные (с точки зрения исследования) свойства оригинала и отвлекаясь от несущественных, модель выступает как некоторый абстрактный идеализированный объект. Всякая модель строится на основе *гипотезы* о возможном устройстве оригинала и представляет собой функциональный *аналог* оригинала. что позволяет переносить знания с модели на оригинал. Критерием адекватности модели является эксперимент.

В идеале модель должна быть *формальной* (т.е. в ней должны быть в явном виде и однозначно заданы исходные объекты, связывающие их отношения и правила обращения с ними) и обладать *объяснительной силой*

(т.е. не только объяснять факты или данные экспериментов, необъяснимые с точки зрения уже существующей теории, но и предсказывать неизвестное раньше, хотя и принципиально возможное поведение оригинала, которое позднее должно подтверждаться данными наблюдения или экспериментов).

Содержание термина "модель" в современной лингвистике в значительной степени охватывалось ранее термином "теория" (особенно Ельмслевым). Считается, что наименования модель заслуживает лишь такая теория. которая достаточно эксплицитно изложена и в достаточной степени формализована (в идеале каждая модель должна допускать реализацию на ЭВМ).

Конструирование модели - не только одно из средств отображения языковых явлений, но и объективный практический критерий проверки истинности знаний о языке. В единстве с другими методами изучения языка моделирование выступает как средство углубления познания скрытых механизмов речевой деятельности, его движения от относительно примитивных к более содержательным моделям, полнее раскрывающим сущность языка.

Внутри языка как системы существует принцип моделирования: одни его подсистемы моделируют другие, например, система письменной речи является моделью устной речи; внутри письменной речи мы имеем дело с несколькими моделями (печатной, рукописной); план выражения является моделью плана содержания.

Метод моделирования обычно опирается на знаковые систем, но язык - сам знаковая система, т.е. слова мы моделируем при помощи слов.

Главная *цель моделирования в лингвистике* - это моделирование целостной языковой способности человека.

Ключевое значение для прикладной лингвистики имеет структурная лингвистика. Структурный подход к анализу языкового материала позволяет формализовать такой весьма расплывчатый и сложный феномен, как

естественный язык. В результате язык и его функционирование представлены посредством формальных моделей.

Понятие лингвистической модели возникло в структурной лингвистике, но вошло в научный обиход в 60-70 гг. 20 в. с возникновением мат. лингвистики и проникновением в лингвистику мат. методов.

Модель включает некоторый инвентарь языковых единиц, взаимоотношения между ними, способ их организации в микросистему. Макросистемой является язык как целостное образование, включающее единицы различных уровней: фонетического, морфологического, лексического, синтаксического.

Теперь, что касается прикладных методик и их характерных черт. Прикладные модели отличаются определенным упрощением, огрублением языковой реальности, но это не значит, что они игнорируют реальную сложность моделируемого объекта. Методология прикладного исследования должна учитывать многоаспектность, многоуровневость, открытость языкового механизма.

Методология - совокупность общих принципов, определяющая способ исследования какого-либо явления; определяет взгляд на объект, как к нему подойти; философские принципы исследования явлений.

Метод - определенный тип способа исследования, определяемый инструментами, которые используются при изучении объекта исследования (метод компьютерного моделирования, статистический метод).

Собственно лингвистические методы:

1. дистрибутивный метод
2. трансформационный метод
3. метод компонентного анализа
4. метод различительных признаков

Методика - конкретный способ исследования, определяемый целью исследования; может объединять несколько методов (методика построения ассоциативных тезаурусов).

Характерные черты прикладных методик

1. ведущая роль метода моделирования
2. экспериментальный характер прикладных методик
3. применение точного метаязыка
4. формализованность самих операций исследования (хотя результат может быть приближенным)
5. использование искусственного метаязыка описания
6. комплексное сочетание разных наук.

Экспериментальные методы (ЭМ) в лингвистике - это методы, позволяющие изучать факты языка в условиях управляемых и контролируемых исследователем. Философской основой применения экспериментальных методов в лингвистике является тезис о единстве теоретического и эмпирического уровней познания.

В современной лингвистике термин "экспериментальный метод" не является четким; лингвисты часто говорят об эксперименте (Э) там, где имеет место *наблюдение*, прежде всего наблюдение над текстами (письменными и устными). Существенно, что текст как таковой, будучи данностью не может быть объектом ЭМ; именно поэтому ЭМ не применимы к изучению истории языка, особенностей стиля автора и т.п. в этих случаях следует говорить о наблюдении. *Объектом ЭМ* является человек - носитель языка, порождающий текст, воспринимающий тексты и выступающий как *информант* для исследователя. в лингвистическом эксперименте исследователь может иметь в качестве подобного объекта самого себя или других носителей языка; в первом случае следует говорить об интроспекции, во втором - об объективном эксперименте.

Экспериментальная работа с информантами (нередко в сочетании с наблюдением) непосредственно в среде носителей языка называется обычно *полевой лингвистикой*.

Историю применения ЭМ в лингвистике можно разделить на три периода:

1. Активное освоение ЭМ в фонетике, акцент на сходстве ЭМ в лингвистике и точных науках (труды Богородицкого, Щербы, Матусевича)

2. Осознание ЭМ в лингвистике как важнейшего способа получения данных о живом языке вообще, включая его морфологию. синтаксис, семантику, а также проблемы языковой нормы, языкового общения, патологий речевого развития и т.д. эта научная программа была впервые сформулирована Щербой ("О трояком аспекте языковых явлений и об эксперименте в языкознании")

3. Реализация указанной научной программы, и как следствие углублении методологических разработок (Апресян, Фрумкина). В социолингвистике и психолингвистике ЭМ занимают доминирующие место.

Последовательное применение ЭМ в исследовании языка и речевых процессов сделало необходимым использование статистических методов при планировании эксперимента и обработке результатов (лингв. статистика). существенно. что лингвист, изучающий речевое поведение человека, имеет дело с объектом, равным ему самому по сложности. В силу этого отношение исследователь - объект в лингвистике превращается в симметричное отношение между двумя исследователями: информант может иметь свою теорию об экспериментаторе и соответственно изменять свое поведение в процессе эксперимента, что может негативно повлиять на результаты Э. Особой сферой использования ЭМ являются машинные эксперименты, проверяющие адекватность формализованных действующих моделей языка.

Процесс эксперимента:

- общая задача
- рабочая гипотеза
- формальные выводы, изменения
- новые гипотезы

Цель эксперимента - проверка гипотез. Человек не должен знать целевую установку экспериментатора.

Типы экспериментов:

- моделирующие эксперименты (в социолингвистике): порождается ряд гипотез, отбираются социальные параметры, которые варьируются
- имитационные эксперименты (лабораторные) - имитация усеченной действительности
- натурные эксперименты включают условия, позволяющие демонстрировать поведение, максимально похожее на реакцию в аналогичной естественной ситуации.

Типы методов (по количеству информантов):

- индивидуальный
- межгрупповой
- многоуровневый, многофакторный

Экспериментальные методы в семантике (Хофман):

- **ассоциативный эксперимент** - испытуемому дается слово-стимул и предлагается реагировать на это слово первым пришедшим в голову словом или словосочетанием
- **метод семантического дифференциала** (экспериментальная семантика) - один из методов построения субъективных сем. пространств (градуированные оценочные шкалы)
- **метод классификации** (в психолингвистике - испытуемым предлагается разбить материал на произвольное количество классов.)

Модель в лингвистике- искусственно создаваемое реальное или вымышленное устройство, воспроизводящее, имитирующее своим поведением поведение оригинала в лингвистических целях.

Типы лингвистических моделей*:*

1. по охвату структуры языка:

- **общие (глобальные)** стремятся охватить весь язык: <VG> (vocabulary, grammar)
- **частные:** фонетическая модель русского языка, модель системы гласных

2. по типологическому статусу:

- **универсальные** стремятся охватить все языки мира: <VG>
- **специфические** характерны для определенного языка или группы языков: мягкость - твердость согласных рус. языка (не действует в англ., франц.)

3. по гносеологическому статусу:

- **модели языка**
- **модели лингвистических знаний** различные фонетические школы
- **модели деятельности лингвиста**

4. по отраженному аспекту языка и речевой деятельности:

Модели различаются не только по направленности на определенный объект, но и по используемым средствам моделирования (алгоритму или исчислению)

Алгоритм - строгая последовательность предписывающих правил

Исчисление - множество разрешающих правил (порядок выполнения не важен)

- **анализирующие модели** моделируют процесс понимания, используют логическое средство алгоритм
- **синтезирующие модели** моделируют процесс вербализации, смысла речевого отрезка
- **порождающие модели** автор Хомский объект моделирования - множество правильных речевых отрезков составляются правила различения приемлемого и неприемлемого; логический средство - исчисление; не служат выражением смысла; на выходе - цепочки элементов (грамм. правильных предложений)
- **собственно структурные модели** основа всех остальных объект моделирования - структура языка как таковая; логический аппарат - логика отношений и классов. Пример: грамматический словарь Залезняка

5. по конечной цели исследования

- **теоретические**
- **описательные**
- **прикладные**

6. по используемым методам
   - **математические модели**
   - **психологические модели**
   - **социологические модели**
7. по функциональному статусу
   - **абстрактно обобщающие модели**
   - **действующие**
8. по используемым материальным средствам
   - **графические**
   - **символьные**
   - **компьютерные**

Частная модель обычно входит в набор частных моделей, описывающий определенный уровень языка:

1. **фонологический уровень**
2. **морфологический уровень**
3. **синтаксический уровень**
4. **лексико-семантический уровень**

При использовании информационных технологий в лингвистике выделяют также следующие типы моделей.

1. ***Структурные модели*** служат для изучения и описания внутреннего строения некоторого объекта. Например, такая модель строится, если необходимо изучить систему согласных какого-либо языка или устройство речевого аппарата человека.

2. ***Функциональные модели*** позволяют изучать поведение некоторого объекта, течение некоторого процесса или же этапы реализации некоторого явления. Например, функциональная модель строится, если необходимо смоделировать процесс со

здания некоторого текста человеком. Такая же модель создается для объяснения процесса перевода текста с одного языка на другой.

3.***Динамические модели*** создаются при необходимости найти объяснение некоторых процессов или явлений в их временном развитии. Так, если требуется узнать, как со временем менялось произношение некоторого слова, строят динамическую модель такого процесса.

Особая роль в лингвистике отводится функциональным моделям, позволяющим раскрыть суть функционирования языка, механизма производства и восприятия речи и текста. Нельзя заглянуть в мозг человека и посмотреть, как в нем осуществляются операции с буквами, звуками, словами, предложениями при всевозможном использовании языка. Поэтому для решения таких задач в рамках функциональных моделей выделяют ***воспроизводящие инженерно-лингвистические модели*** (ВИЛМ). Они представляют собой компьютерные системы, поведение которых, с одной стороны, имитирует поведение реальных лингвистических объектов, а с другой стороны, позволяет хотя бы частично воспроизвести эти реальные объекты.

Как отмечалось выше, существуют разные способы формализованного описания объекта, процесса или явления: формулы, таблицы, графики, схемы, наборы предложений естественного языка и т.д. Все эти способы составляют основу алгоритмического решения задач с помощью ПК.

Основные теоретические требования к модели:

1. полнота модели - способность отражать все факты, на которые она рассчитана, на охват которых она претендует

2. простота - удобство, использования как можно меньшего числа средств (символов, правил) для достижения поставленной научной цели

3. объяснительная сила - способность модели вскрывать причины наблюдаемых фактов и предсказывать новые факты (например. модели исторического изменения слова; системы машинного перевода в очень малой степени объяснительные)

4. адекватность - свойство максимальной похожести на моделируемый объект, на оригинал, можно свести к объяснительной силе или теоретико-множественному соответствию

5. экономность - экономичное использование энергетических и временных ресурсов при применении модели

6. точность - возможность выполнения операций представляемым моделью формальным аппаратом

7. эстетические свойства - красота модели

Прикладные критерий: главное - удобство модели. Для моделирования языка очень важны логические средства реализации модели (компьютерное воплощение модели).

Тема 2. Основные составляющие информационных технологий.

1. Структура информационных технологий
2. Теоретические основы информационных технологий.
3. Будущее информационных технологий.
4. Алгоритмы и их свойства.
5. Элементы языка программирования Turbo Pascal.

1. Структура информационных технологий.

В составе современных информационных технологий можно выделить следующие составляющие:

1) теоретические основы информационных технологий;

2) методы решения задач информационными технологиями;

3) средства решения задач, используемые в информационных технологиях:

а) аппаратные средства;

б) программные средства.

2. Теоретические основы информационных технологий.

Теоретическую основу информационных технологий составляют важнейшие понятия и законы информатики. В свою очередь понятие «информатика» тесно связано с понятием «информация».

Слово **информация** (от лат. *informatio* — разъяснение, изложение) в обычном житейском понимании обозначает некоторые сведения о внешнем и внутреннем мире, которые мы используем для регулирования своего поведения. Более строго это понятие раскрывается разными способами. Выберем один из них, который более всего подходит к рассматриваемым в пособии лингвистическим задачам, и определим информацию как определенным образом связанные сведения, данные, понятия, отраженные в нашем сознании и изменяющие наши представления о реальном мире.

Информация обладает разными свойствами. Наиболее важными из них являются: ценность, достоверность, полнота, актуальность, логичность, компактность. ***Ценность*** информации определяется тем, насколько она важна для позволяющих получателю информации достичь своей цели. ***Актуальность*** информации определяется необходимостью ее немедленного использования для достижения какой-либо цели. ***Компактность*** информации — способность представить ее в наиболее сжатом виде. Понятия ***достоверность*** и ***логичность*** информации не требуют особых пояснений.

Выделяют различные виды информации. При этом для ее классификации по видам разработано много подходов, использующих разнообразные признаки и особенности информации. Так, в зависимости от того, какими органами чувств воспринимается информация, ее делят на визуальную, аудиальную (звуковую, фонетическую), аудиовизуальную, тактильную. По направленности информации всем членам общества или каким-то его группам различают информацию массовую, предназначенную для всех членов общества, и специальную — для специалистов в различных областях науки, техники, культуры, производства. Специальную информацию подразделяют на научную, техническую, производственную, эстетическую и т.п.

В каждом виде специальной информации выделяют подвиды. Например, в зависимости от области науки и научной информации выделяют информацию физическую, математическую, биологическую, лингвистическую и т.д. Так, **лингвистической информацией** называют множество определенным образом связанных сведений, данных, понятий о языке и правилах его функционирования, отраженных в нашем сознании и влияющих на наше речевое поведение.

Слово «информатика» также не имеет единого определения. С современной точки зрения **информатика** — это наука о законах и методах получения, хранения, передачи, распространения, преобразования и использования информации в естественных и искусственных системах с применением компьютера.

В зависимости от вида информации выделяют различные типы информатики. Так, различают информатику социальную, экономическую, научную, научно-техническую, статистическую, биологическую, медицинскую и т. п.

Наука, изучающая законы и методы организации и переработки с помощью компьютера лингвистической информации, называется лингвистической информатикой. Вспомнив, что понимается под

лингвистической информацией, можно сказать, что *объектом исследования* лингвистической информатики будет структура слов, словосочетаний, предложений, текстов. ,

Ее интересуют правила, объединяющие нижестоящие языковые единицы в вышестоящие, правила перевода предложений и текстов, способы построения рефератов и аннотаций, пути обучения языкам и целый ряд других вопросов, связанных с языком и речью.

3. Будущее информационных технологий

Философы, психологи и другие специалисты отмечают, что в будущем социально защищенным может считаться лишь тот человек, который способен гибко перестраивать направление и содержание своей деятельности в связи со сменой технологий или требований рынка. Чтобы подготовить такого человека, необходимо заменить традиционную технологию получения новых знаний более эффективной организацией познавательной деятельности обучаемых в ходе учебного процесса. Это можно сделать с использованием современных информационных технологий. Именно они могут продемонстрировать обучаемому тот факт, что любой информационный ресурс представляет реальную ценность лишь в том случае, когда к нему организован соответствующий доступ. С их помощью будущих специалистов можно научить правильной организации хранения информации и выбору адекватных форм ее представления. «Интеллектуальная собственность, представленная в цифровом формате, станет главной "валютой" XXI века».

Технология получения и распространения новых знаний уже сейчас неотделима от Internet. Формируется инфраструктура, объединяющая в единое целое глобальные и местные телекоммуникационные каналы, радио, телевидение, телефонные линии связи (одна планета — одна сеть). Все это не просто создает сверхинтеллект, а

формирует новые измерения сознания, феномен сверхпсихологических изменений в личности человека. Широкое распространение в недалеком будущем получат видеоконференции и дистанционное обучение. К концу XXI века ученые изучат принципы работы мозга на уровне отдельных нейронов и научатся обращаться с ним, как со сложным электронным объектом. Это даст новый толчок к созданию систем искусственного интеллекта, систем автоматического порождения текстов, их перевода, реферирования и т.д.;

4. Алгоритмы и их свойства.

С точки зрения современной психологии **задача** в самом общем понимании — это некоторая цель, поставленная в конкретных условиях и требующая исполнения, решения. Примерами интеллектуальных задач являются следующие: 1) решить полное квадратное уравнение $ax^2 + b\,x + c = 0$; 2) составить таблицу значений x^2, x^3 и 1/x величины *x,* меняющейся с некоторым шагом *к* от некоторого начального значения *n* до некоторого конечного значения *m;* 3) найти среди группы русских глаголов те, которые употреблены в инфинитиве; 4) составить реферат научного текста; 5) перевести текст с английского языка на русский и т.д. Чтобы решить задачу, необходимо знать ее ***начальные условия,*** а также ***метод или способ*** ее решения.

Алгоритмы являются объектом систематического исследования пограничной между математикой и информатикой научной дисциплины, примыкающей к математической логике – *теории алгоритмов*.

Само слово «алгоритм» происходит от algorithmi – латинской формы написания имени великого математика IX века аль-Хорезми, который сформулировал правила выполнения арифметических действий.

Алгоритм – это точно определенная последовательность действий, которые необходимо выполнить над исходной информацией, чтобы получить решение задачи.

Алгоритм решения задачи, заданный в виде последовательности команд на языке вычислительной машины (в кодах машины), называется **машинной программой.**

Команда машинной программы (иначе, машинная команда) – это элементарная инструкция машине, выполняемая ею автоматически без каких-либо дополнительных указаний и пояснений.

Графическое представление алгоритмов.

Алгоритм, составленный для некоторого исполнителя, можно представить различными способами: с помощью графического или словесного описания, в виде таблицы, последовательностью формул, записанным на алгоритмическом языке (языке программирования). Остановимся на графическом описании алгоритма, называемом блок-схемой. Этот способ имеет ряд преимуществ благодаря наглядности, обеспечивающей, в частности, высокую «читаемость» алгоритма и явное отображение управления в нем.

Прежде всего определим понятие блок-схемы. Блок-схема – это ориентированный граф, указывающий порядок исполнения команд алгоритма.

Понятие исполнителя алгоритма.

Понятие исполнителя невозможно определить с помощью какой-либо формализации. Исполнителем может быть человек, группа людей, робот, станок, компьютер, язык программирования и т.д. Важнейшим свойством, характеризующим любого из этих исполнителей, является то, что исполнитель умеет выполнять некоторые команды. Так, исполнитель-человек умеет выполнять такие команды, как «встать», «сесть», «включить компьютер» и т.д., а исполнитель – язык программирования Бейсик –

команды PRINT, END, LIST и другие аналогичные. Вся совокупность команд, которые данный исполнитель умеет выполнять, называется системой команд исполнителя. (СКИ).

В качестве примера рассмотрим исполнителя – робота, работа которого состоит в собственном перемещении по рабочему полю (квадрату произвольного размера, разделенному на клетки) и перемещении объектов, в начальный момент времени находящихся на «складе» (правая верхняя клетка).

Одно из принципиальных обстоятельств состоит в том, что исполнитель не вникает в смысл того, что он делает, но получает необходимый результат. В таком случае говорят, что исполнитель действует формально, т.е. отвлекается от содержания поставленной задачи и только строго выполняет некоторые правила, инструкции.

Это – важная особенность алгоритмов. Наличие алгоритма формализует процесс решения задачи, исключает рассуждение исполнителя. Использование алгоритма дает возможность решать задачу формально, механически исполняя команды алгоритма в указанной последовательности. Целесообразность предусматриваемых алгоритмом действий обеспечивается точным анализом со стороны того, кто составляет этот алгоритм.

Введение в рассмотрение понятия «исполнитель» позволяет определить алгоритм как понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленной цели. В случае исполнителя-робота мы имеем пример алгоритма «в обстановке», характеризующегося отсутствием каких-либо величин. Наиболее же распространенными и привычными являются алгоритмы работы с величинами – числовыми, символьными, логическими и т.д.

Свойства алгоритмов.

Алгоритм должен быть составлен таким образом, чтобы исполнитель, в расчете на которого он создан, мог однозначно и точно следовать командам алгоритма и эффективно получать определенный результат. Это

накладывает на записи алгоритмов ряд обязательных требований, суть которых вытекает, вообще говоря, из приведенного выше неформального толкования понятия алгоритма. Сформулируем эти требования в виде перечня свойств, которым должны удовлетворять алгоритмы, адресуемые заданному исполнителю.

1. Одно из первых требований, которое предъявляется к алгоритму, состоит в том, что описываемый процесс должен быть разбит на последовательность отдельных шагов. Возникающая в результате такого разбиения запись представляет собой упорядоченную совокупность четко разделенных друг от друга предписаний (директив, команд, операторов), образующих прерывную (или, как говорят, дискретную) структуру алгоритма. Только выполнив требования одного предписания, можно приступить к выполнению следующего. Дискретная структура алгоритмической записи может, например, подчеркиваться сквозной нумерацией отдельных команд алгоритма, хотя это требование не является обязательным. Рассмотренное свойство алгоритмов называют **дискретностью.**

2. Используемые на практике алгоритмы составляются с ориентацией на определенного исполнителя. Чтобы составить для него алгоритм, нужно знать, какие команды этот исполнитель может понять и исполнить, а какие – не может. Мы знаем, что у каждого исполнителя имеется своя система команд. Очевидно, составляя запись алгоритма для определенного исполнителя, можно использовать лишь те команды, которые имеются в его СКИ. Это свойство алгоритмов будем называть **понятностью.**

3. Будучи понятным, алгоритм не должен содержать предписаний, смысл которых может восприниматься неоднозначно, т.е. одна и та же команда, будучи понятна разным исполнителям, после исполнения каждым из них должна давать одинаковый результат.

Запись алгоритма должна быть настолько четкой, полной и продуманной в деталях, чтобы у исполнителя не могло возникнуть потребности в принятии решений, не предусмотренных составителем алгоритма. Говоря иначе, в алгоритмах недопустимы также ситуации, когда после выполнения очередной команды алгоритма исполнителю неясно, какая из команд алгоритма должна выполняться на следующем шаге.

Отмеченное свойство алгоритмов называют **определенностью** или **детерминированностью.**

4. Обязательное требование к алгоритмам – **результативность.** Смысл этого требования состоит в том, что при точном исполнении всех предписаний алгоритма процесс должен прекратиться за конечное число шагов и при этом должен получиться определенный результат. Вывод о том, что решения не существует – тоже результат.

5. Наиболее распространены алгоритмы, обеспечивающие решение не одной конкретной задачи, а некоторого класса задач данного типа. Это свойство алгоритма называют **массовостью**. В простейшем случае массовость обеспечивает возможность использования различных исходных данных.

Итак, алгоритм решения задачи имеет ряд обязательных свойств:

- Дискретность – разбиение процесса обработки информации на более простые этапы (шаги выполнения), выполнение которых компьютером или человеком не вызывает затруднений,
- Определенность алгоритма – однозначность выполнения каждого отдельного шага преобразования информации,
- Выполнимость – конечность действий алгоритма решения задач, позволяющая получить желаемый результат при допустимых исходных данных за конечное число шагов,
- Массовость – пригодность алгоритма для решения определенного класса задач.

Виды алгоритмов.

Алгоритмы бывают трех основных видов, которые и являются базовыми при написании программ.

Первый тип – *линейный* алгоритм; такой, в котором все действия выполняются в строгом порядке, последовательно, одно за другим. Типичный жизненный пример такого алгоритма – рецепт пирога.

Второй тип – *разветвляющийся* алгоритм; такой, в котором выполняются те или иные действия в зависимости от выполнения или невыполнения некоего условия. Пример из жизни – правило перехода улицы по светофору. Если горит красный – стоим, если горит зеленый – идем.

Наконец, третий тип – *циклический* алгоритм; такой, в котором присутствуют повторяющиеся действия с какой-либо изменяющейся величиной, так называемым параметром. Пример – колка дров. Берем полено – ставим на попа, колем топором, берем втрое полено и т.д., пока поленья не закончатся, и эта работа нам не надоест.

Примеры.

В приведенных ниже последовательностях каждый следующий элемент получен по некоторому строгому алгоритму. Разгадав его, продолжите ряд:

- а, в, д, ё, з, й; {л – буквы русского алфавита идут через одну}
- 1, 2, 4, 8, 16, 32; {64 – каждое предыдущее число умножается на 2}
- 1, 4, 9, 16, 25; {36 – квадраты натуральных чисел}
- 1, 1, 2, 3, 5, 8, 13, 21, 34; {55 – так называемые числа Фибоначчи, где каждое последующее образуется суммой двух предыдущих}
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 0, 1, 1, 1, 2, 1; {3 – перечисляется последовательность чисел, но двузначные числа представляются в виде составляющих их цифр: 1,0 – 10; 1,1 – 11; 1,2 – 12 и т.д.}

- победа, обеда, беда, еда; {да – каждое последующее слово образуется отсечением первой буквы от предыдущего}
- о, д, т, ч, п, ш, с, в, д, д . { о – первые буквы числительных, начиная с одного: о – один, д –два, т – три и т.д.}

5. Элементы языка программирования Turbo Pascal.

Таблица 1. Запись арифметических выражений на ТР.

Математическая функция	Запись на языке Pascal
сложение	+
вычитание	–
модуль числа x	Abs (x)
$\sqrt{x}$	Sqrt (x)
x^2	Sqr (x)
e^x	Exp (x)
sin x	Sin (x)
cos x	Cos (x)
ln x	Ln (x)
умножение	*
деление	/
величина числа пи	pi
целая часть числа: [x]	int (x)
дробная часть числа: {x}	frac (x)
целая часть от деления A/B	A div B
дробная часть от деления A/B	A mod B
округление числа x до целого	Round(x)

Арифметическое выражение – это последовательность констант, переменных, стандартных функций, разделенных знаками арифметических операций и круглыми скобками. Порядок действий в арифметическом выражении:

1. скобки;
2. стандартные функции;
3. *, /, DIV, MOD – слева направо;
4. +, – слева направо;

Например, арифметическое выражение 5 * 8 / 100 * 2 – 3 * 1e2 / (5 * 4) выполняется в следующем порядке:

5 * 4 = 20;
5 * 8 = 40;
40 / 100 = 0.4;
0.4 * 2 = 0.8;
3 * 1e2 = 300.0;
300.0 / 20 = 15.0;
0.8 - 15.0 = -14.2.

Примеры записи на Pascal арифметических выражений:

1) $7{,}9 * 10^5 \ \sin X^2$ = 7.9e5 * SIN(X * X);

2) $10^{-4} \cos^2 X - \sqrt{3 \lg Z}$ = 1e-4 * SQR(COS(X)) – SQRT(3 * LN(Z) / LN(10));

3) $tg19° + 2^{-3X}$ = SIN(PI * 19 / 180) / COS(PI * 19 / 180) + EXP(–3 * X * LN(2));

Типы данных в ТР.

Переменные-это величины, которые могут последовательно изменяться в процессе выполнения программы и для обращения, к которым нужно указать их имя.

Имя переменной (идентификатор) – последовательность букв латинского алфавита и цифр количеством не более 8 и начинающаяся с буквы. В имени переменной может встречаться подчеркивание “_”.

В языке Turbo Pascal соответствующие заглавные и строчные буквы в именах и служебных словах не различаются, так, например, следующие три имени переменных одинаковы: index, INDEX, Index.

В качестве имен переменных нельзя использовать служебные слова (например: PROGRAM, BEGIN, END и т.д.) и желательно не использовать названия функций (например: SIN, COS, TRUE, REAL...). Переменные, так же как и константы могут быть целого типа (INTEGER), вещественного (REAL), логического (BOOLEAN), символьного(CHAR).

Все переменные в Pascal характеризуются своим типом. Тип переменных характеризует множество значений, которые может принимать

переменная, а также множество операций, которые можно производить над переменной.

Язык Turbo Pascal является типизированным или статическим. Это означает, что тип переменной описывается в начале каждой программы и не может быть изменен в процессе ее выполнения.

Тип данных определяет: 1. количество отведенных в памяти байтов. 2. способ представления значения переменной в памяти. 3. набор операций допустимых над этой переменой.

Типы данных принято делить на простые (или базовые) и структурированные.

К простым типам в языке Pascal относятся:

- Целый;
- Вещественный;
- Логический;
- Символьный;
- Перечисляемый;
- Тип-диапазон.

Структурированные типы описывают наборы однотипных или разнотипных данных. Типы данных, образующих набор, в свою очередь могут быть как простыми, так и структурированными. Структурированный тип может иметь неограниченное количество уровней структурирования.

К стандартным структурированным типам относятся:

- Массив;
- Множество;
- Запись;
- Строка;
- Файл.

Все типы, кроме вещественного, также называются дискретными.

Целый тип

Имеется пять целых типов, различающихся допустимым диапазоном значений и размером занимаемой оперативной памяти. Целые типы обозначаются идентификаторами INTEGER, BYTE, SHORTINT, WORD, LONGINT; их характеристики:

Таблица 2.Диапазоны значений целых типов.

Целый тип	Диапазон значений	Размер памяти
INTEGER	-32768…32767	2 байта
WORD	0…65535	2 байта
BYTE	0…255	1 байт
SHORTINT	-128…128	1 байт
LONGINT	-214783648…214783647	4 байта

Для работы с целыми типами данных используются константы MAXINT, MININT и арифметические функции +, -, *, DIV, MOD, ABS, SQR, а также функции DEC, INC, ORD, PRED, SUCC, TRUNC, ROUND, рассмотренные ниже.

MAXINT = -32768, MININT = 32767.

Вещественный тип

Эта группа типов означает множества вещественных значений в различных диапазонах. Turbo Pascal поддерживает четыре различных вещественных типа. Они именуются идентификаторами REAL, SINGLE, COMP, DOUBLE, EXTENDED.

Таблица 3.Диапазоны значений вещественных типов.

Вещественный тип	Диапазон значений	Число цифр мантиссы	Размер памяти
REAL	2.9e-39…1.7e38	11 – 12	6 байт
SINGLE	1.5e-45…3.4e38	7 – 8	4 байта
DOUBLE	5.0e-324…1.7e308	15 – 16	8 байт

EXTENDED	3.4e-493…1.1e493	19 – 20	10 байт
COMP	-2e63… 2e63	Нулевое	8 байт

Замечание: хотя тип COMP считается вещественным типом, он содержит только целые числа из весьма значительного диапазона, которые представляются в вычислениях как вещественные (с нулевой мантиссой).

Для работы с вещественными типами данных используются операции +, -, *, /, ABS, SQR, ARCTAN, COS, SIN, SQRT, EXP, LN, FRAC, INT.

Логический тип BOOLEAN

Имеется два значения логического типа, представляющие логические значения (TRUE - истина и FALSE - ложь). Над значением булевского типа допустимы операции сравнения, причем для установления порядка принято, что

FALSE < TRUE.

Для работы с логическими типами используются логические функции NOT, OR, XOR, AND.

Символьный тип CHAR

Значениями символьного типа являются символы из множества ASCII (американский стандартный код для обмена информацией). Это множество состоит из 256 различных символов, упорядоченных определенным образом, и содержит символы заглавных и строчных букв, цифр, и других символов включая специальные управляющие символы.

Символьный тип – это любой символ в кавычках, например:

' + ' , ' A ' , ' 8 ' , ' - ' .

Символьные переменные упорядочены соответственно их порядковым номерам в ASCII символ, например: ‘7’ < ‘8’, ‘C’ < ‘D’. Русские прописные буквы не упорядочены по алфавиту.

Для работы с символьным типом используются функция ORD, описанная ниже.

Ограниченный тип (тип-диапазон)

Ограниченный тип получается из дискретного типа путем ограничения допустимых значений. Тип, на который накладываются ограничения, называется базовым типом. Описание ограниченного типа:

VAR <имя переменной>: <нижняя граница>..<верхняя граница>;

Здесь <нижняя граница> и <верхняя граница> должны принадлежать одному базовому типу.

Например:

```
VAR
MONTH:  1..12;   {ограниченный тип, базовый тип - целый}
CHAS:    0..24;  {ограниченный тип, базовый тип - целый}
BUKVA:  'A'..'Z'; {ограниченный тип, базовый тип -символьный}
CIFRA:   '0'..'9';  {ограниченный тип, базовый тип - символьный}
```

Внимание! Ограничение вещественного типа не допускается.

Всякий раз, когда заранее известно, что значения некоторой переменной лежат внутри некоторого диапазона, следует использовать в программе ограниченный тип, поскольку при выполнении программы происходит автоматическая проверка выхода значений переменных за допустимый диапазон.

Перечислимый тип

Перечислимый тип задается путем явного перечисления всевозможных значений, при этом обращение к элементу возможно по его имени.

В общем случае для перечислимого типа:

<тип> = (<значение>, <значение>, …, <значение>);

Перечислимые типы считаются дискретными типами. Над значениями перечислимых типов определены операции сравнения (считается, что значения перечислимого типа указаны в списке в порядке возрастания). Кроме того, допускается образование ограниченных типов из перечислимых. Например:

```
VAR
```

DEN: (PN, WT, SR, CH, PT, SB, WS); {перечислимый тип}

RAB_DEN: PN..PT; {ограниченный тип, базовый тип - перечислимый}

WT < SR, SR > WT, WS > SB, PN = PN, PT <> SB.

Для дискретных величин можно использовать функции PRED, SUCC:

PRED(<аргумент>) - предыдущее значение аргумента.

SUCC(<аргумент>) - последующее значение аргумента.

Примеры:

RRED(8) = 7; PRED('D') = 'C';
SUCC(8) = 9; SUCC('D') = 'E';
PRED(TRUE) = FALSE; PRED(SR) = WT;
SUCC(FALSE) = TRUE; SUCC(SR) = CH.

2) Pascal содержит группу стандартных функций, специально предназначенных для преобразования типов. Эти функции можно рассматривать как своего рода унарные операции, операндом которых является значение одного типа, а результатом – значение другого типа.

Операнд (аргумент) функции записывается в круглых скобках, а имя функции (идентификатор) ставится перед скобками, общий вид:

<функция> (<аргумент>)

Например: CHR(s).

Таблица 4.Стандартные функции преобразования:

Функция	Тип аргумента	Тип результата	Что делает
ODD	Целый	Булевский	Принимает значение TRUE, если аргумент нечетный, FALSE – если четный
TRUNC	Вещественный	Целый	Отсекает дробную часть числа
ROUND	Вещественный	Целый	Округляет вещественное число до ближайшего целого числа
ORD	Дискретный	0..255	Ставит в соответствие ASCII
CHR	0..255	Символьный	Выдает символ из ASCII с заданным в аргументе порядковым номером

DEC	Целый	Целый	Уменьшает значение аргумента на 1
INC	Целый	Целый	Увеличивает значение аргумента на 1
FRAC	Вещественный	Вещественный	Возвращает дробную часть аргумента
INT	Вещественный	Вещественный	Возвращает целую часть аргумента

Примеры:

TRUNC(7.59) = 7;	ROUND(-8.3) = -8;
ROUND(7.59) = 8;	DEC(25) = 24;
DEC(8) = 7;	TRUNC(-0.5) = 0;
TRUNC(-8.3) = -8;	ROUND(-0.5) = -1;
ORD(8) = 8;	INT(1.83) = 1.0;
ORD(126) = 126;	INT(0.001) = 0.0;
FRAC(1.001) = 0.001;	ODD(25) = TRUE;
ORD('0') = 48;	ORD('z') = 122;
ORD('b') = ORD('a')+1;	ORD('2') = ORD('1') + 1;
CHR(48) = '0';	CHR(ORD('z')) = 'z';

Формат программы

Раздел описаний

PROGRAM <имя программы>;

<раздел меток> (LABEL);

<раздел описаний> (VAR);

<раздел констант> (CONST);

<раздел типов> (TYPE);

<раздел функций и процедур>;

BEGIN

<раздел операторов>;

END.

Простейший оператор ввода и вывода

К операторам ввода относятся:

Read (<список переменных через запятую>);

Readln (<список переменных через запятую>);

Readln;

Второй отличается от первого тем, что после ввода переводит курсор на новую строку. Третий оператор используется для организации паузы.

К операторам вывода относятся:

Write (<список вывода>);

Writeln (<список вывода>);

Writeln;

В списке переменных можно писать строковые константы (последовательность символов в апострофах) и даже выражения (выводятся их значения). Действие операторов аналогично соответствующим операторам вывода.

Линейным называется алгоритм, в котором результат получается путем однократного выполнения заданной последовательности действий при любых значениях исходных данных. Операторы программы выполняются последовательно, один за другим, в соответствии с их расположением в программе.

Условный оператор.

Для организации разветвлений в программах используют оператор условного перехода (оператор условия или условный оператор), который имеет следующий вид:

IF <условие> THEN <оператор 1> ELSE <оператор 2>;

Здесь IF /если/ – служебное слово;

THEN /то/ – служебное слово;

ELSE /иначе/ – служебное слово;

<условие> – это логическое (или булевское) выражение, которое может принимать одно из 2-х значений: или TRUE (истина), или FALSE (ложь). В зависимости от того истинно или ложно значение логического выражения, выполняется или оператор 1 после THEN, или оператор 2 после ELSE.

Часть оператора со служебным словом ELSE может отсутствовать. В этом случае при ложности логического выражения ничего не делается и выполнение условного оператора заканчивается. Например, рассмотрим часть программы:

```
READ(X);
IF X * 2 > 17 THEN K := 1  ELSE K := 2;
K := 3 * K;
WRITE(K);
```

Если ввести число x=15, то логическое выражение x*2>17 будет истинным (т.к. 15*2>17) и выполняется оператор k:=1. Затем выполняется оператор k:=3*k и в окно OUTPUT будет выведено число 3.

Если при повторном выполнении ввести, например, число 5, то x=5 и результатом логического выражения x*2>17 будет ложь (т.к. 5*2<17), выполняется k:=2, следующий за словом ELSE. Далее выполняется оператор присваивания k:=3*k и в окно OUTPUT будет выведено значение 6.

Часто по смыслу алгоритма после служебных слов THEN и ELSE бывает необходимо записать не один, а несколько операторов. В этом случае следует использовать составной оператор, который имеет следующий вид:

BEGIN <оператор>; ... ; <оператор> END

Примеры условных операторов:

```
IF X < 0 THEN Y := -X ELSE Y := X ;
IF Z >= 0 THEN F := SQRT(Z) ;
IF (X <= 0) AND (Y >= 0) THEN WRITELN('ТОЧКА ВО 2 ЧЕТВЕРТИ');
IF A < B THEN BEGIN C := A; A := B; B := C END;
```

Оператор выбора CASE

Оператор выбора CASE может быть использован вместо нескольких условных операторов If, если требуется сделать выбор более чем из двух возможностей.

Структура этого оператора в Турбо Паскале:

CASE <ключ выбора> OF

C1: <оператор 1>;

C2: <оператор 2>;

.....

CN: <операторN>;

[else < оператор 0>;]

end;

Здесь <ключ выбора> – это выражение порядкового типа, в зависимости от значения которого принимается решение;

Оператор перехода GOTO.

Оператор безусловного перехода goto служит для прерывания естественного хода выполнения программы. Следующим выполняется оператор, помеченный меткой, которая указана в данном операторе перехода. Один оператор может помечаться несколькими метками.

Goto < метка>;

<метка> – это целое число без знака или идентификатор, обязательно описанный в разделе описания меток (LABEL).

Циклические программы.

В программах, связанных с обработкой данных или другими сложными операциями, часто выполняются циклически повторяющиеся действия.

Цикл представляет собой последовательность операторов, которая выполняется неоднократно.

В языке программирования Pascal имеется стандартный набор из трёх разновидностей цикла – цикл со счётчиком (инструкция for), цикл с предусловием (инструкция while) и цикл с постусловием (инструкция repeat). При изучении циклов часто возникают затруднения при выборе оператора цикла.

Оператор цикла с параметром FOR.

Оператор цикла For организует выполнение одного оператора заранее известное число раз. Существует 2 варианта:

возрастающий: For i:=<начало> to <конец> do

begin

<оператор>

end.

убывающий: For i:=<начало> downto <конец> do

begin

<оператор>

end.

где i – параметр цикла, переменная порядкового типа, которая принимает последовательные значения от начала до конца, <начало> – выражение, определяющие начальное значение параметра цикла, <конец> – выражение, определяющее конечное значение параметра цикла, <оператор> – выполняемый оператор, может быть составным.

Оператор цикла с предусловием WHILE.

Оператор цикла с предусловием имеет вид:

WHILE <условие> DO <оператор>;

Условие проверяется перед каждым выполнением тела цикла (<оператора>). Если условие принимает значение TRUE, то тело цикла выполняется, если значение FALSE, происходит выход из цикла. <Оператор> может быть составным. Цикл с предусловием используется для программирования процессов, в которых число повторений оператора цикла не известно, а задается некоторое условие его окончания.

Оператор цикла с постусловием REPEAT

Оператор цикла с постусловием имеет вид:

REPEAT

<операторы>;

UNTIL <условие >;

где repeat-повторять, until- до тех пор, пока…

Этот оператор аналогичен оператору цикла с предусловием, но отличается от него тем, что проверка условия производится после очередного выполнения тела цикла. Это обеспечивает его выполнение хотя бы один раз.

Одномерные массивы.

Массив – это совокупность конечного числа данных одного типа. Массив описывается в разделе переменных:

var <имя массива>: array [n1..n2] of <тип элементов массива>

Для обозначения элементов массива используются имя переменной-массива и индекс.

Для проверки программ удобно задавать случайное заполнение массива с помощью генератора случайных чисел: random(n) – задаёт случайное целое число в диапазоне от 0 до n–1. Процедура randomize позволяет получать разные наборы случайных чисел при каждом запуске программы.

Двумерные массивы.

Массивы, положение элементов в которых описывается двумя элементами, называются двумерными. Их можно представить в виде прямоугольной таблицы или матрицы. Первый индекс элемента массива – номер строки, второй индекс – номер столбца. Рассмотрим матрицу A размером 2*3, то есть в ней будет две строчки по три элемента:A=

a_{11} a_{12} a_{13}

a_{21} a_{22} a_{23}

Каждый элемент имеет свой номер, как у одномерных массивов, но номер состоит из двух чисел - номера строки, в которой находится элемент, и

номера столбца. Существует несколько способов объявления двумерного массива:

Способ 1. В Turbo Pascal двумерный массив можно описать как одномерный, элементами которого являются одномерные массивы.

Например, для матрицы А, приведённой выше:const n=2; m=3;

type. mas1=Array[1..m] of<тип элементов>;

mas2=Array[1..n] of mas1;

Var v: mas1;a: mas2;

Способ 2. Описание массива А можно сократить, исключив определение типа mas2:const n=2; m=3;

type massiv=array[1..n] of array[1..m] of <тип элементов>;

var a: massiv;

Способ 3. Ещё более краткое описание массива А можно получить, указывая имя массива и диапазоны изменения индексов для каждой размерности массива:

const n=2; m=3;

type massiv=array[1..n,1..m] of <тип элементов>ж

var a:massiv;

Формирование двумерного массива можно осуществлять всеми тремя способами, как и для одномерных, то есть ввод с клавиатуры, через генератор случайных чисел или с помощью файла.

Чтобы заполнить случайными числами (от 0 до 9) матрицу n×m и вывести её на экран в виде таблицы, необходимо набрать следующий код:

```
for i:=1 to n do begin
for j:=1 to m do begin
matr[i, j]:=random (10);
write (matr[i, j]:3);
end;
writeln;
```

end;

Работа с символами.

Основные сведения о символьных величинах.

С помощью ЭВМ можно решать весьма разнообразные задачи обработки текстов: от составления платежных ведомостей до автоматической верстки газет. Для того, чтобы ЭВМ могла обрабатывать тексты, она должна уметь оперировать не только с числами, но и со словами.

Познакомимся с основными приемами обработки текста на компьютере.

Будем полагать, что текст - это произвольная последовательность символов некоторого алфавита. Алфавитом может служить любое множество символов, например (Q, 1,2, .,.), (A,B, C…).

Строкой символов, или символьной (строковой, текстовой) константой, будем называть последовательность символов, заключенных в кавычки.

Строка символов может состоять из одного или нескольких символов, а также не содержать ни одного символа (пустая строка, или строка нулевой длины - максимальная длина текстовой строки 255символов.

Мы знаем, что для обработки данных того или иного типа используются переменные. Тип переменной определяется типом данных, которые она представляет.

В Turbo Pascal 7.0 для работы с символами используются два типа переменных:

-символьный тип данных;

-строковый тип данных;

Символьный тип данных (Char).

Описание: идентификатор char, {var S: char)

Диапазон значений значением переменной этого типа может быть любой символ -это буквы, цифры, знаки препинания и каждому символу алфавита соответствует индивидуальный числовой код от 0 до 255.

Примечание: наиболее распространенной международной согласованной системой кодирований всех символов является система ASCII. Символы с кодами от 0 до 127 представляют так называемую основную таблицу кодов ASCII. Эта часть идентична на всех IBM-совместимых компьютерах. Коды с символами от 128 до 255 представляют национальную часть.

Так как символы языка упорядочены, то к символьным данным применимы операции сравнения. Операция сравнения осуществляется следующим образом: из двух символов меньше тот, который встречается в таблице ASCII раньше.

В Turbo Pascal 7.0 значения для переменных типа char задаются апострофах: ch=’*’; a:=’3’; letter:=’G’;

Строковый тип данных (string)

Как правило, одно целое число или один символ занимают в памяти ЭВМ два байта. В то же время для изображения символа достаточно одного байта. С целью экономии памяти машины при использовании символьных данных в языке Паскаль введено понятие строки. Строкой называется последовательность символов определенной длины. Элементы строки хранятся по два в двух байтах памяти ЭВМ.

Переменные типа string могут быть объявлены следующим образом:

Var strl: string[30]; str2: string.

1.Стандартные функции, процедуры для работы с символьными величинами.

Операция сложения (конкатенация) символьных величин.

Операция сложения (конкатенация) позволяет строить из двух символьных строк третью, состоящих из символов первой, за которой следуют символы второй. Обозначается эта операция знаком"+".

Пример:

Описываем строковые переменные. .

var strl, str2, str3: string[20].

Присваиваемое значение строки заключается в апострофы. Присвоим первым двум следующие значения, а третья будет равна их склеиванию:

Strl:='У Егорки';

str2:='всегда отговорки';

str3:=str1+’’+str2.

Строка str3 имеет значение 'У Егорки всегда отговорки’. В данном примере итоговая строка может состоять, максимум из 20 символов, если она будет состоять из большего числа, то будут взяты в качестве значения только первые 20, а остальные рассматриваться не будут.

Сравнение*' ..

Turbo Pascal 7.0 позволяет выполнять операции, сравнения двух строк. Сравнение происходит посимвольно слева направо: сравниваются коды соответствующих символов до тех пор, пока не нарушится равенство и сразу делается вывод о знаке неравенства. Две строки называются равными, если они равны по длине и совпадают посимвольно.

'Balkon'<balkon'
(Ord('B')<Ord('b'));
‘ba1коп'> 'balken'
(Ord('o')>0rd('e'));

"balkon" > "balk" длина первой строки больше 'balkon' > 'balk' Можно использовать любые сравнения (>, <, =, <>, >=, <=) и их комбинации в условных операторах. Их результат - это одно из двух значений: В Turbo Pascal 7.0 True или False.

В Turbo Pascal 7.0 для доступа к отдельному символу в строке необходимо указать имя строки и в квадратных скобках номер позиции элемента (символа) в строке. При этом по отношению к отдельному символу строки возможны все те же операции, что и к переменной типа Char.

2. Стандартные процедуры и функции языка Turbo Pascal 7.0.

Turbo Pascal 7.0 предоставляет в распоряжение пользователя целый ряд встроенных функций и процедур, предназначенных для обработки строк. Рассмотрим наиболее важные из них.

Длина строки.

Под длиной строки понимается количество введенных символов, но она не может превышать максимально возможной длины (в описательной части). Это значение можно определить при помощи функции, результат которой целое число, равное количеству символов.

Пример.

```
Length(str)
str1: = 'ABCDEFGH';
str2: = 'Мама мыла раму';
k1:=Length(strl);
k2:=Length(str2).
```

В результате значения целых переменных будут равны: kl=8, k2=14.

Копирование.

Функция copy(str, n, т) в Turbo Pascal 7.0 - копирует т символов строки str, начиная с n-го символа, при этом исходная строка не меняется. Можно результат этой функции присваивать другой строке или сразу выводить его на экран.

Пример.

```
    str1:=’ABCDEFGH’;
str2: ="abcdefgh";
str3:=copy(strl,4,3);
writeln(str3);
writeln(copy(str2, 4, 3));
```

Значение переменной Str3='DEF'. А на экране будут выведены следующие строки:

DEF и def.

Удаление.

В Turbo Pascal 7.0 для этого используется процедура Delete (str, n,m), которая вырезает из строки str m символов, начиная с п-го, таким образом сама строка изменяется.

Пример.

Дан фрагмент программы:

```
strl:='ABCDEFGH';
delete(str1, 3, 4);
writeln(str1);
```

После выполнения этих операторов из строки будут удалены четыре символа, начиная с третьего, то есть строка будет такой: Str1='ABGH'.

Замена (Вставка).

Строка - символьная переменная, в которой требуется произвести изменения;

начало - начальная позиция в исходной строке, с которой производится замена;

[длина] - количество символов из символьного выражения. Если длина опущена, то производится замена полным содержимым символьного выражения;

символьное выражение - представляет новые символы (может быть переменной, константой или выражением).

В Turbo Pascal 7.0 это можно сделать, применяя процедуру lnsert(Strl,Str2,n) - вставка строки Strl в строку Str2, начиная с п-го символа, при этом первая строка остается такой же, как и была, а вторая получает новое значение.

Пример.

```
Strl = ‘ABCDEFGH’;
Str2:=’abcdefgh’;
Insert(str1,str2,3);
```

В результате выполнения данной процедуры строка будет такой str2='abABCDEFGHcdefgh'

Подстрока.

В Turbo Pascal 7.0 существуют функции, определяющие позицию подстроки в строке. Результат этих функций - целоечисло, и оно определяет номер первого элемента, с которого начинается первое вхождение подстроки. Если такой подстроки нет, то значение функции равно 0.

Пример.

```
posstr1:='cde';
str2:='ABCDEFGH';
kl:=pos(str1, str2);
k2:=pos(str2,str1);
```

В результате получим: k1=3 k2=0.

Числа и строки.

Надо заметить, что число 13 и строка 13 - это не одно и то же. Для работы с числами и строками применяются две процедуры.

В Turbo Pascal 7.0 Str(N,Str1) - переводит числовое значение N в строковое и присваивает результат строке Str1, причем можно переводить как целые числа, так и вещественные.

Пример:

```
str(n,str1)
str(1234, strl)
- после выполнения strl ='1234';
```

Существует обратная операция, переводящая строковое значение в числовое.

В Turbo Pascal функция val(str, N, К) - переводит строковое значение в числовое, если данная строка действительно является записью числа (целого

или вещественного), то значение К=0, а N - это число, иначе К будет равно номеру символа, в котором встречается первое нарушение записи числа N.

Пример.

valfstr, n, к)

val(‘1245',n,k) п=1234, k=0;

Функции преобразования типов.

Иногда в программах возникает необходимость по коду определить символ и, наоборот, по символу определить его код. Для этого используют функции:

CHR(X:BYTE):SHAR

Эти функции возвращают символ, соответствующий ASCII -коду числа X.

Пример.for i = 0 to 255 do

writelnt(i,’’, chr$(i));

Для определения кода по символу используют функцию ORD.

ORD.

readln(str);

writeln(ord(str))

Функции UCASE$ и UPCASE.

В Turbo Pascal 7.0 функция UPCASE(str) - преобразует символ из строчного в прописной. Эта функция, в отличие от предыдущей, рассчитана на обработку только одного символа. Она не преобразует символы кириллицы.

Пример.

upcase

ch = upcase$('n') -выполнено

ch = upcase$(‘язык') -не выполнено

ch = upcase$('n') -не выполнено

ch =upcase$(‘?')-не выполнено

ch = upcase$(‘until') -не выполнено

Работа с файлами.

Файлы классифицируются по двум признакам:

По методу доступа- последовательный и прямой доступ.

По типу (логической структуре) – типизированные, текстовые, нетипизированные. Типизированные и нетипизированные файлы при записи кодируются, текстовые записываются напрямую, их можно прочитать любым текстовым редактором. Любой файл необходимо связать с соответствующей файловой переменной.

Операции над файлами в Turbo Pascal.

Assign(<имя переменной-файла>,<имя внешнего файла> - процедура устанавливает соответствие между файловой переменной и внешним файлом.

Append(File) – процедура, открывает существующий файл для добавления в него новой информации.

Rewrite(File) – процедура, создаёт новый файл, связанный с файловой переменной File. Если такой файл уже существует, то он удаляется и на этом месте создаётся новый пустой файл.

Reset(File) – процедура, открывает существующий физический файл, который был связан с файловой переменной File.

Close(File) – процедура, закрывает открытый файл.

Так как, по определению, число элементов файла не задаётся заранее, то в языке Turbo Pascal введён признак конца файла. Это логическая функция EOF<имя переменной - файла>. Она используется для определения, достигнут ли конец файла или ещё нет. Для определения конца файла используется оператор цикла.

Множества.
Множество – это набор элементов одинакового типа, которые рассматриваются как единое целое. Элементы множества не пронумерованы, поэтому нельзя обратиться к отдельному элементу множества по его индексу. Поэтому множества используются в тех задачах, где порядок следования элементов не имеет значения.

Для объявления множественного типа используется словосочетание set of(множество из), например: type A= Set of B; здесь A – множественный тип, B – базовый тип элементов множества.

Записи.
При решении задач обработки большого количества данных используются массивы. Но при работе с массивами основное ограничение заключается в том, что все элементы массива должны быть одного типа. Для работы с комбинациями данных разных типов в Pascal применяется комбинированный тип данных – запись.

Запись – это структурированный тип данных, состоящий из фиксированного числа компонентов одного или нескольких типов, называемые полями записи. В отличие от массива, компоненты записи могут быть разного типа. Чтобы можно было ссылаться на тот или иной компонент записи, каждое её поле имеет своё имя. Записи можно объявлять следующим образом.

Сначала объявляется тип записи:

```
Type
   имя типа = record
      имя поля1: тип поля1;
      имя поля2: тип поля2;
      ……
      имя поляN: тип поля N
end;
```

Затем объявляются переменные соответствующего типа.

```
Var
имя переменной: имя типа;
```

Например, данные об автомобилях:

```
Type avto=record
  Number: integer;              {номер автомобиля}
  Marka: string [20];       {марка автомобиля}
  Fio: string [40];         {Ф.И.О. владельца}
  Address: string[60];      {адрес владельца}
 End;
Var m,v: avto;
```

Контрольные вопросы.

1. Что понимается под информатизацией общества?
2. Что такое информационные технологии?
3. Какими основными свойствами характеризуются информационные технологии?
4. Какие подходы к определению информации вы знаете?
5. Этапы развития информационных технологий.
6. Информационное общество. Модели и проблемы информатизации общества.
7. Определите понятие и характеристики автоматизированной информационной технологии.
8. Как соотносятся информационная технология и информационная система?
9. Назовите основные характеристики новой информационной технологии.
10. Какова цель информационной технологии?
11. По каким признакам классифицируют информационные технологии?
12. Что представляет собой технологический процесс обработки информации?

13. Что такое этапы и технологические операции?
14. Назовите основные этапы технологического процесса обработки информации.
15. Какие технологические операции различают по содержанию и последовательности преобразования информации? Охарактеризуйте их.
16. Что понимается под программным обеспечением?
17. Какие программные средства относятся к базовому программному обеспечению?
18. Какая основная функция выполняется базовым программным обеспечением?
19. Укажите назначение и функции основных групп прикладного программного обеспечения.
20. Какие ППП относятся к классу универсальных?
21. Какие ППП относятся к классу проблемно-ориентированных?
22. Что такое математическое обеспечение ИС?
23. Что относится к средствам математического обеспечения?
24. Перечислите основные группы экономико-математических методов.
25. Что понимают под организационным обеспечением ИС?
26. Что представляет собой лингвистическое обеспечение ИС?
27. Что включается в состав правового обеспечения ИС?
28. Перечислите и охарактеризуйте виды информации.
29. Перечислите и охарактеризуйте свойства информации.
30. Какие существуют подходы к измерению количества информации?
31. Что такое лингвистическая информация?
32. Что является объектом исследования лингвистической информатики?
33. Что такое алгоритм?
34. Что такое машинная программа?
35. Графическое представление алгоритмов.
36. Одномерные массивы.

37. Двумерные массивы.
38. Свойства алгоритмов.
39. Виды алгоритмов.
40. Типы данных в ТР.
41. Что такое переменная?
42. Простые типы данных в Турбо Паскаль.
43. Операторы ввода и вывода.
44. Условный оператор.
45. Оператор выбора CASE
46. Как выполняется операция сложения (конкатенация) символьных величин?

Тема 3. Базы и банки данных.

1) Понятие банка данных.
2) Классификация баз данных.
3) Определения, классификация, характеристика MSAccess.
4) Свойства полей и типы данных
5) Типы данных:
6) Типы связей между данными
7) Объекты БД.

1. Понятие банка данных.

Информация - сведения, неизвестные ранее получателю информации, пополняющие его знания, подтверждающие или опровергающие положения и соответствующие убеждения. Информация носит субъективный характер и определяется уровнем знаний субъекта и степенью его восприятия. Информация извлекается субъектом из соответствующих данных.

Знания - совокупность фактов, закономерностей и эвристических правил, с помощью которых решается поставленная задача. Последовательность операций обработки данных называют информационной технологией (ИТ). В силу значительного количества информации в современных задачах она должна быть упорядочена. Существует два подхода к упорядочению.

Данные связаны с конкретной задачей (технология массивов) - упорядочение по использованию. Вместе с тем алгоритмы более подвижны (могут чаще меняться), чем данные. Это вызывает необходимость переупорядочения данных, которые к тому же могут повторяться в различных задачах.

В связи с этим предложена другая, широко используемая технология баз данных, представляющая собой упорядочение по хранению.

Под базой данных (БД) понимают совокупность хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений. Целью создания баз данных, как разновидности информационной технологии и формы хранения данных, является построение системы данных, не зависящих от принятых алгоритмов (программного обеспечения), применяемых технических средств и физического расположения данных в ЭВМ; обеспечивающих непротиворечивую и целостную информацию при не регламентируемых запросах. БД предполагает многоцелевое ее использование (несколько пользователей, множество форм документов и запросов одного пользователя).

База знаний (БЗ) представляет собой совокупность БД и используемых правил, полученных от лиц, принимающих решения (ЛПР).

Наряду с понятием «база данных» существует термин «банк данных», который имеет две трактовки.

В настоящее время данные обрабатываются децентрализовано (на рабочих местах) с помощью персональных компьютеров (ПК). Первоначально же использовалась централизованная обработка на больших ЭВМ. В силу централизации базу данных называли банком данных и потому часто не делают различия между базами и банками данных.

Банк данных - база данных и система управления ею (СУБД). СУБД (например, FoxPro) представляет собой приложение для создания баз данных как совокупности двумерных таблиц.

Таким образом, **банк данных -** это система специальным образом организованных баз данных, программных, технических, языковых и организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

Банк данных - это совокупность баз данных созданных в интересах решения задач по определенным направлениям науки и техники. Часто банком данных называют комплекс, состоящий из совокупности информационных баз данных, программного обеспечения для работы с ними - СУБД, соответствующего комплекса технических средств и обслуживающего персонала.

Любой банк данных в своем составе содержит следующие два основных компонента: БД является как дата - логическое представление информационной модели предприятия и СУБД, с помощью которой реализуются централизованное управление данными, хранимыми в базе, доступ к ним и поддержание их в состоянии соответствующем состоянию предметной области.

Центральную роль в функционировании банка данных выполняет система управления базой данных (СУБД) – это пакет программ, обеспечивающий поиск, хранение, корректировку данных, формирование ответов на запросы. Система обеспечивает сохранность данных, их

конфиденциальность, перемещение и связь с другими программными продуктами.

Банк данных – это не только интегрированное хранение данных, но и идея отделения описания данных от программ их обработки, интерфейс между которыми обеспечивается СУБД.

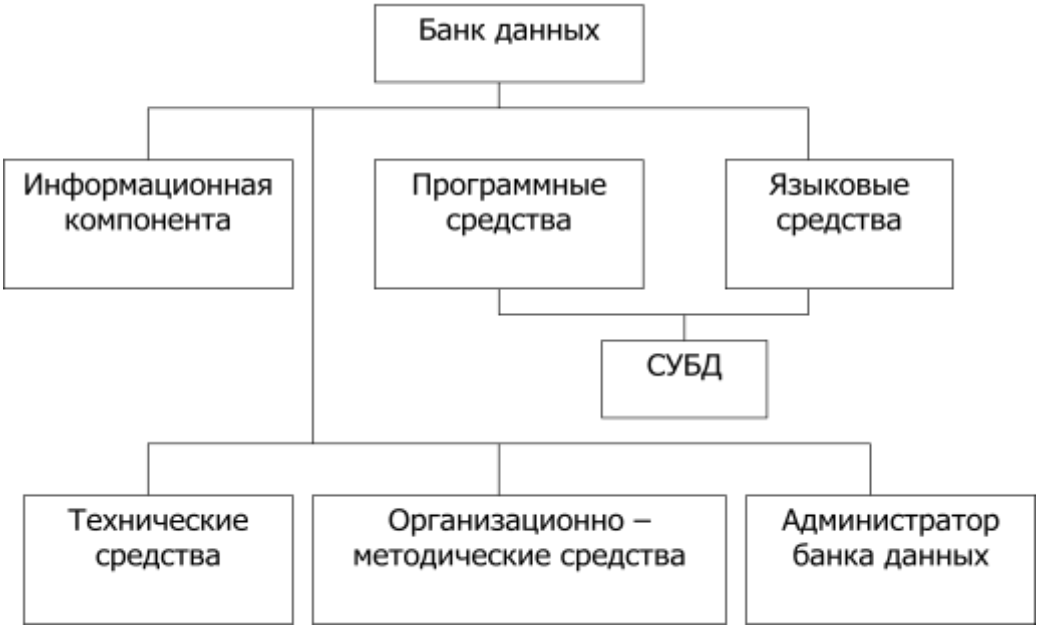


Рис.1 Компоненты банка данных

Информационная компонента:

База данных – это поименованная совокупность взаимосвязанных данных, находящихся под управлением СУБД.

Комплекс программных и языковых средств:

СУБД – сложный комплекс, обеспечивающий взаимодействие всех частей информационной системы при ее функционировании. Сюда входят организация ввода, обработка и хранение данных, а также средства настройки системы и ее тестирования. Языковые средства обеспечивают интерфейс пользователя с БД.

Технические средства: компьютеры, устройства ввода и отображения выводимой информации;

Организационно-методические средства: инструкции, методические и регламентирующие документы, предназначенные для различных пользователей, имеющих доступ к информации.

Администратор банка данных: группа специалистов, обеспечивающих создание, функционирование и развитие банка данных.

Уровни представления данных:

Логический (даталогический) – уровень математической модели, условное представление данных как системы объектов и связей между ними | программно-аппаратные средства СУБД

Физический (внутренний) – уровень программно-аппаратной реализации хранения данных;

Внешний (концептульный)– визуальное представление данных, с которым работает конечный пользователь | языки управления базами данных (SQL)

2.Классификация баз данных.

Рис.2. Классификация БД.

1) по модели представления данных:

иерархическая БД – база данных, в которой связь между элементами осуществляется по типу подчинения и схематично изображается в виде дерева. Иерархия начинается с корневого узла. Каждый узел имеет только одного «предка» и N «потомков».

(+) простота и однозначность представления, легкость адресации

(–) существенная зависимость от программно-аппаратных средств

Пример: дерево папок Windows, каталог ресурсов Интернет

сетевая БД, возможно существование любых взаимосвязей между объектами. Если изобразить эту модель графически, то получится набор узлов на плоскости, связанных линиями со стрелками.

(+) теоретически возможны сколь угодно сложные связи между объектами;

(–) сложность реализации, существенная зависимость от программно-аппаратных средств

Пример: службаWWW – документы, произвольно связанные ссылками.

реляционная БД, представление данных в виде системы взаимосвязанных таблиц. Каждый объект системы описывается в виде таблицы с набором свойств (атрибутов), а взаимосвязь между объектами – связями между таблицами.

(+) простота; относительная независимость от программных и аппаратных средств;

(–) существенная зависимость скорости обработки от объема БД

Использование: все существующие СУБД

2) Классификация БД по организации хранения данных и обращения к ним:

локальные (персональные),

сетевые (интегрированные),

распределенные базы данных.

3) Классификация БД по типу хранимой информации:

документальные,

фактографические,

лексикографические.

Среди документальных БД различают **библиографические, реферативные** и **полнотекстовые.**

К лексикографическим БД относятся различные словари (классификаторы, многоязычные словари, словари основ слов и т. п.).

Реляционные СУБД осуществляют:

работу с базой данных через экранные формы;

организацию запросов на поиск данных с помощью специальных языков запросов высокого уровня;

генерацию отчётов различной структуры данных с подведением промежуточных и окончательных итогов;

вычислительную обработку путём использования встроенных функций, программ, написанных с использованием языков программирования и макрокоманд.

Терминология реляционных СУБД

Поле (атрибут) – свойство описываемого объекта;

Запись (кортеж) – значение атрибута объекта;

Таблица (отношение) – совокупность записей с заполненными значениями атрибутов;

Структура БД (схема отношения) – совокупность информации о полях таблицы;

Понятие ключа БД.

Для идентификации каждой записи в таблице используется уникальный маркер, который называют **первичным ключом.**

Внешний ключ – поле, содержащее ссылку на поле первичного ключа в другой таблице.

Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

Стандартные требования к СУБД – правила Кодда

уникальность записей;

неупорядоченность записей и полей;

атомарность значений атрибутов (нормализованное представление «поля-записи»)

Метод нормализации отношений

Нормализация – это разбиение таблицы на несколько, обладающих лучшими свойствами при обновлении, включении и удалении данных.

3.Определения, классификация, характеристика MSAccess.

Цель любой информационной системы – обработка данных об объектах реального мира. В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира, в какой- либо предметной области. Создавая базу данных ,пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, если данные структурированы.

Структурирование – это введение соглашений о способах представления данных, сочетанием признаков. Сделать это возможно, если данные структурированы.

Неструктурированными называют данные, записанные, например, в текстовом файле.

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а так же специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут работать по-разному с разными объектами и предоставляют различные функции, и средства большинство СУБД опирается на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятие, приемы и методы на весь класс СУБД. В качестве такого учебного объекта выберем СУБД MicrosoftAccess, входящую в пакет MicrosoftOffice.

4.Свойства полей и типы данных

Полем обычно называется столбец в таблицах баз банных. В каждом поле хранится определенный тип информации. Например, в таблице адресов акционеров в одном из полей может находиться имя, в другом - почтовый индекс, а в третьем название улицы или дома.

Очень мощное средство MicrosoftAccess - способность хранить в поле не только текст или числа - но и объекты OLE - например, рисунки или иные документы различных приложений Windows. Свойства поля определяются типом данных, размещаемых в столбце и задаются наряду с его именем при создании таблицы в режиме конструктора. При необходимости, каждое поле можно описать.

Размер текстового поля по своей длине должен быть достаточным для самого длинного значения или осмысленной аббревиатуры. Однако не рекомендуется задавать размер поля необоснованно большим - это плохо сказывается на эффективности работы СУБД и проводит к неоправданно длинным фразам.

Размер поля и объем дискового пространства не связаны между собой. Даже если длину поля установить несколько избыточной, это не влияет на размер файла базы данных. Например, если Вы определите длину поля в 50 символов, а вводить в поле будете лишь 2 буквы, MicrosoftAccess будет хранить данные кусками не в 50, а в 2 символа. Это возможно благодаря применению в программе специального метода последовательного индексированного доступа (SequentialMicrosoftAccessMethod) для хранения данных.

Размер числового поля или поля счетчика позволяет указать количество значимых цифр, вводимых в это поле чисел. Формат поля можно видоизменять. Это позволяет по-разному представлять число в поле. Для изменения формата выделенного поля применяется бланк свойств Формат поля режима конструктора, путем выбора нужного формата числа из раскрывающихся доступных значений.

Количество десятичных знаков - определяет точность числовых значений в числовых и денежных полях. Этот параметр, как и предыдущие,

является одним из свойств поля и задается аналогичным образом. Однако, что бы использовать точность, предусмотренную для данного типа числового поля, необходимо указать это поле в бланке свойств, для этого в строке Формат поля надо открыть список и выбрать режим Авто. Но если для поля установлен Стандартный формат чисел (General), свойство Число десятичных знаков не работает.

Подпись поля.Это свойство удобно для пояснения данных - например, при заполнении форм или составления отчетов с подведением итогов. Поясняющий текст вводится в строку Подпись поля его содержание, естественно, не зависит от типа данных.

Значение по умолчанию - присваивается в строке Значение по умолчанию. Это упрощает ввод данных в поле: как только курсор окажется в ячейке поля с заданным по умолчанию значением, оно тут же появится. В качестве значения по умолчанию можно применять формулы и выражения с помощью построителя выражений. Для этого необходимо щелкнуть кнопку с многоточием.

Условие на значение - свойство поля, позволяющее ввести ограничения на вводимые в поля данные. Сформировать условие на значение можно, перейдя к соответствующему свойству в бланке свойств и активизировав программу - мастер «Построитель выражений» щелчком кнопки с многоточием.

Число столбцов определяет число столбцов, выводящихся в списке или в раскрывающемся списке поля со списком, или число столбцов, передаваемых в объекты OLE в диаграмме или в рамке свободного объекта. Например, если задать для свойства Число столбцов списка в форме «Сотрудники» значение 3, то можно вывести в одном столбце имена сотрудников, в другом их фамилии, а в третьем коды.

Ширина столбцов определяет ширину столбца в поле со списком или в списке с несколькими столбцами. Его использование также позволяет скрывать столбцы.

Описание определяет текст, содержащий описание объекта, выводящегося в окне базы данных, а также описание отдельных полей таблицы или запроса.

Свойство поля «Тип данных»

Свойство Тип данных определяет тип данных, сохраняемых в поле таблицы. В каждое поле допускается ввод данных только одного типа.

4. Типы данных:

Текстовый(Значение по умолчанию).- Текст или числа, не требующие проведения расчетов, например, номера телефонов. Размер - Число символов, не превышающее минимальное из двух значений: 255 или значение свойства Размер поля (FieldSize). MicrosoftAccess не сохраняет пустые символы в неиспользуемой части поля.

Поле MEMO - Длинный текст или комбинация текста и чисел. Размер - До 65535 символов. (Если поле MEMO обрабатывается через объекты доступа к данным (DAO) и содержит только текст и числа, а не двоичные данные, то его размер ограничивается размером базы данных).

Числовой - Числовые данные, используемые для проведения расчетов. Размер - 1, 2, 4 или 8 байт (16 байт только для кода репликации).

Дата/время - Даты и время, относящиеся к годам с 100 по 9999, включительно. Размер – 8 байт.

Денежный - Денежные значения и числовые данные, используемые в математических расчетах, проводящихся с точностью до 15 знаков в целой и до 4 знаков в дробной части. Размер - 8 байт.

Счетчик - Уникальные последовательно возрастающие (на 1) или случайные числа, автоматически вводящиеся при добавлении каждой новой записи в таблицу. Значения полей типа счетчика обновлять нельзя. Размер - 4 байт (16 байт, если для свойства Размер поля (FieldSize) задано значение кода репликации).

Логический - Логические значения, а также поля, которые могут содержать одно из двух возможных значений (True/False, Да/Нет). Размер - 1 бит.

Поле объекта OLE - Объект (например, электронная таблица MicrosoftAccess, документ MicrosoftWord рисунок, звукозапись или другие данные в двоичном формате), связанный или внедренный в таблицу MicrosoftAccess. Размер - До 1 Гбайт (ограничивается объемом диска).

Гиперссылка - Строка, состоящая из букв и цифр, и представляющая адрес гиперссылки. Адрес гиперссылки может состоять максимум из трех частей: текст - текст, выводимый в поле или в элементе управления; адрес - путь к файлу (в формате пути UNC) или странице (адрес URL), дополнительный адрес - смещение внутри файла или страницы. Чтобы вставить адрес гиперссылки в поле или в элемент управления, выберите команду Гиперссылка из меню Вставка. Для получения дополнительных сведений см. раздел Ввод адреса гиперссылки в режиме формы и в режиме таблицы. Размер - Каждая из трех частей в типе «Гиперссылка» может содержать до 2048 символов.

Мастер подстановок - Создает поле, в котором предлагается выбор значений из списка, или из поля со списком, содержащего набор постоянных значений или значений из другой таблицы. Выбор этого параметра в списке в ячейке запускает мастера подстановок, который определяет тип поля.

Размер - Тот же размер, что и у ключевого поля, используемого в подстановке (обычно 4 байт). Значение данного свойства может быть задано только в верхней половине окна таблицы в режиме конструктора.

Денежный тип данных рекомендуется использовать для полей, в которых планируется хранить числовые значения с одним - четырьмя знаками в дробной части. При обработке значений полей типа «С плавающей точкой (4 байт)» (Single) и «С плавающей точкой (8 байт)» (Double) выполняются вычисления с плавающей точкой. Для значений денежных полей используются более быстрые вычисления с фиксированной точкой.

Для того чтобы указать специальный формат или один из встроенных форматов отображения для поля с типом «Числовой», «Денежный», «Дата/время» или «Логический», следует определить значение свойства Формат поля (Format).

5.Типы связей между данными

Связь – отношение между двумя общими полями двух таблиц.

Тип создаваемой в схеме данных связи зависит от полей, для которых определяется связь:

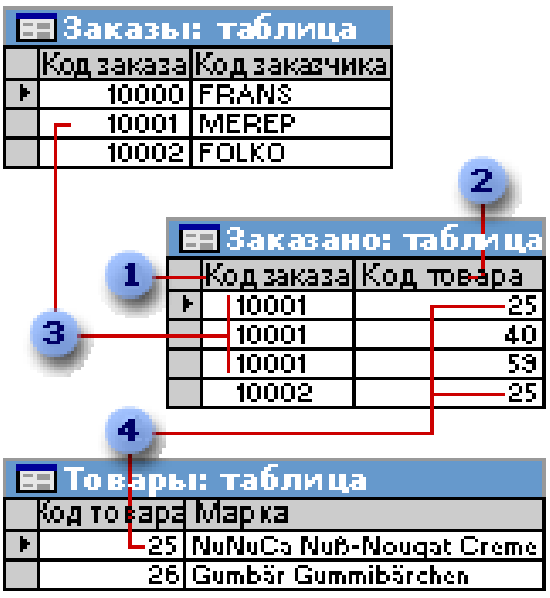
Отношение «один-к-одному» (1:1) – каждая запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Может использоваться для разделения таблиц, содержащих много полей, для отделения части таблицы по соображениям безопасности, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Пример: Студент - № зачетки

Отношение «один-ко-многим» (1:N) – каждой записи в таблице А могут соответствовать несколько записей в таблице В, но не наоборот.

Пример 1: Группа – Студенты

Пример 2: Отношение «многие-ко-многим»



1 - Первичный ключ из таблицы «Заказы»

2 - Первичный ключ из таблицы «Товары»

3 - Один заказ может содержать несколько товаров,

4 - ... а каждый товар может содержаться в нескольких заказах.

Отношение «многие-ко-многим» (N:M) - одной записи в таблице А могут соответствовать несколько записей в таблице В, а одной записи в таблице В несколько записей в таблице А. Этот тип связи возможен только с помощью третьей (связующей) таблицы, первичный ключ которой состоит из двух полей, которые являются внешними ключами таблиц А и В.

Пример 1: Студенты - Курсы, которые они слушают

Пример 2: Отношение «один-ко-многим»

1 - Один поставщик...

2 - ... может поставлять несколько товаров,

3 - ... но у каждого товара может быть только один поставщик.

6.Объекты БД.

7.Объекты БД MSAccess

Кроме таблиц база данных может содержать и другие типы объектов. Привести полную классификацию возможных объектов баз данных затруднительно, поскольку каждая система управления базами данных может реализовать свои типы объектов. Однако основные типы объектов рассмотреть можно.

Таблицы. Это основные объекты любой базы данных. Во – первых в таблицах хранятся все данные, имеющиеся базе, а во вторых таблицы хранят структуру базы (поля, их типы и свойства).

Формы, назначение форм.

Форма – это способ оформления заголовка, области данных, области примечаний таблиц и заголовков. Формы позволяют создавать пользовательский интерфейс для таблиц базы данных. Форма строится на основе таблицы или запроса. При каждом открытии сохраненной формы обновляются данные запроса, на основе которого создается форма. Формы могут быть выведены на экран в трех видах: режим конструктора, режим формы, режим таблицы.

Запросы, назначение запросов.

Запрос – это средство выборки данных из одной или нескольких таблиц. Отбор осуществляется по условию заданному пользователем. Запросы используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным.

Виды запросов

1) Запрос на выборку является наиболее часто используемым типом запроса. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

Запрос на выборку

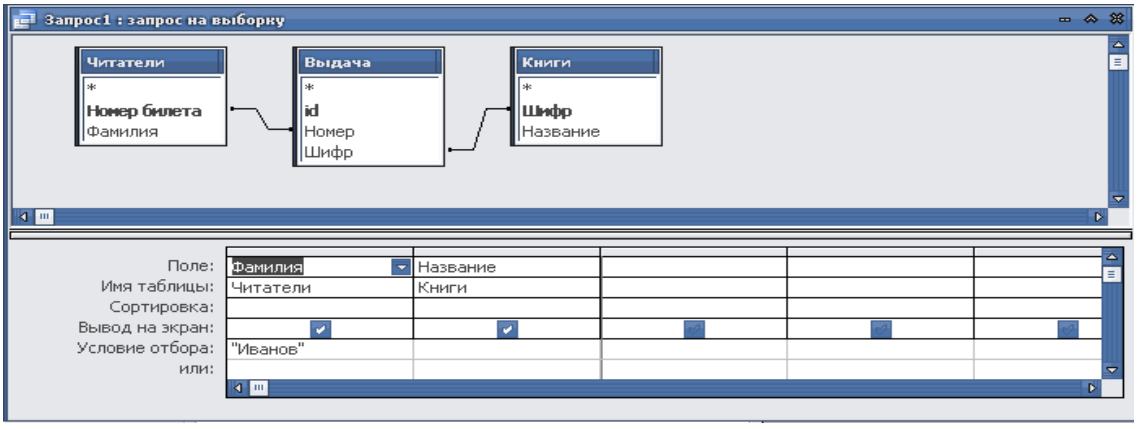


Рис. 3. Бланк запроса на выборку.

2) Запрос с параметрами— это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле. Запросы с параметрами также удобно использовать в качестве основы для форм, отчетов и страниц доступа к данным.

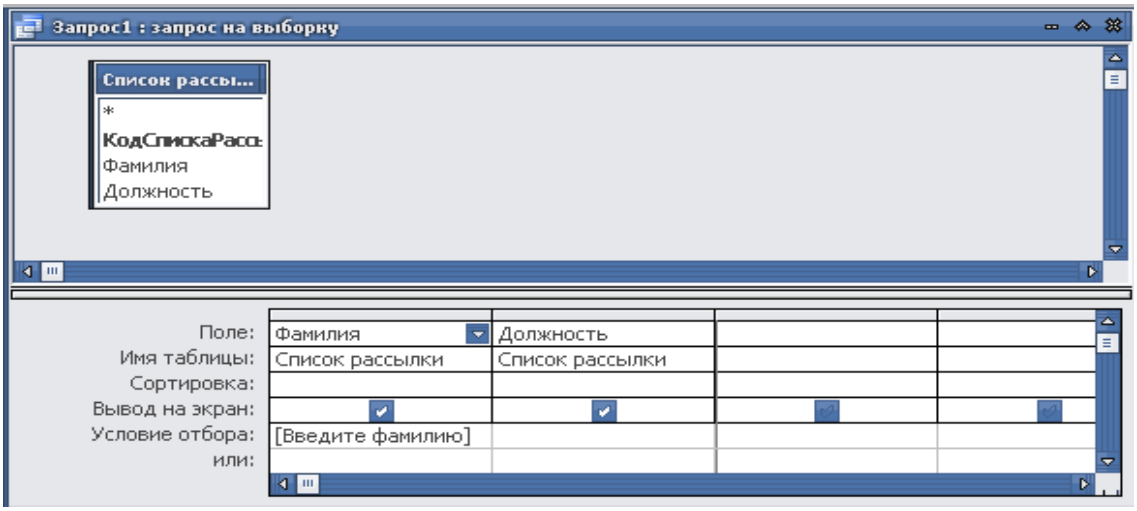


Рис.4. Параметрический запрос

3) Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

4) Запрос на изменение - это запрос, который за одну операцию изменяет или перемещает несколько записей.

Существует четыре типа запросов на изменение:

На удаление записи

На обновление записи

На добавление записей

На создание таблицы.

Запрос на создание таблицы – основан на запросе выборки, но в отличие от него, результат запроса сохраняется в новой таблице.

Итоговые запросы, назначение которых отдаленно напоминают готовые функции электронных таблиц (производят математические вычисления по заданному полю и выдают результат).

Специфические запросы SQL- запрос к серверу базы данных, написанные на языке запросов SQL.

Отчеты

Отчеты – это средства отображения данных и результатов при выводе на печать. Отчет является эффективным средством представления данных в печатном формате.

Большинство отчетов являются присоединенными к одной или нескольким таблицам и запросам из базы данных. Источником записей отчета являются поля в базовых таблицах и запросах. Отчет не должен включать все поля из каждой таблицы или запроса, на основе которых он создается.

Страницы доступа к данным - Web-страницы для удаленного доступа к БД.

Макрокоманды - инструкция, выполняющая определенное в СУБД действие (открыть документ, изменить размер шрифта и т.п.)

Макросы - набор из одной или более макрокоманд, выполняющих последовательность операций (таких, как открытие форм или печать отчетов). Они могут быть полезны для автоматизации часто выполняемых задач.

Модули - наборы описаний и подпрограмм на VisualBasic для автоматизированной работы с БД.

Сортировка данных

Простая сортировка - все записи поля сортируются по возрастанию или по убыванию (но не одновременно). Выполняется в режимах формы, таблицы или страницы.

Сложная сортировка - по некоторым полям допускается сортировка по возрастанию, а по другим полям сортировка по убыванию. Выполняется в режиме конструктора запроса или отчета, в окне расширенного фильтра, в режиме сводной диаграммы или сводной таблицы.

Фильтрация данных

Фильтр - это набор условий, применяемых для отбора подмножества записей. Существуют четыре типа:

Фильтр по выделенному фрагменту - способ быстрого отбора записей по выделенному образцу.

Обычный фильтр – от таблицы остается одна запись. Каждое поле становится списком, в котором можно выбрать выводимые значения для данного поля.

Расширенный фильтр – фильтр создается как SQL-запрос. Условие отбора можно построить, пользуясь построителем выражений.

8.**Структурированный язык запросов SQL.**

С целью поиска информации и модернизации данных в СУБД используется специальный непроцедурный язык структурированных запросов SQL (Structured Query Language). Он является индустриальным стандартом. Его непроцедурность означает, что на этом языке можно указать, что нужно сделать с базой данных, но невозможно описать алгоритм этого процесса. Язык SQL включает группы операторов, ответственных за ведение баз данных и поиск в них нужной информации.

Основные директивы (операторы) SQL:

CREATE-создание объектов без данных (например; таблиц)

ALER- изменение объектов БД

INSERT-добавление новых записей в уже существующие в БД таблицы

DELETE- удаление записей из таблицы

SELECT - выборка данных из существующих в БД таблиц по определённым критериям.

UPDATE - изменение данных в существующих записях созданных таблиц.

Директива CREATE.

CREATE TABLE – создать таблицу.

Общий синтаксис этой директивы выглядит след образом:

CREATE TABLE **Имя_таблицы** (**Имя_столбца1 Тип_данных** (размер_данных), **Имя_столбца2 Тип_данных** (размер_данных)), . . .)

Имя_таблицы – это имя создаваемой таблицы.

Т.к не все серверы баз данных воспринимают русскоязычные имена таблиц, то поэтому имена таблиц следует задавать на английском языке.

Имя_столбца – это имена необходимых столбцов в создаваемой таблице.

Тип_данных – задается для каждого столбца и определяется данными, которые будут храниться в столбце. Для некоторых типов данных в обязательном порядке необходимо задать **Размер_данных. Размер_данных** определяет, какое количество символов будет храниться в каждой записи этого столбца.

Имена таблиц должны быть уникальными, т.е. в одной таблице не может быть двух таблиц с одинаковыми именами. В каждой таблице имена столбцов должны быть уникальными. В качестве имен таблиц, столбцов и др. объектов нельзя использовать зарезервированные слова SQL-языка. Например, TABLE, INSERT, DELETE и др.

Пример.

CREATE TABLE Tabl1(NUM Integer, FIO Char(50), FOTO Image);

В результате выполнения этого запроса будет создана таблица с названием Tabl1, в которой 3 столбца NUM, FIO, FOTO.

Этот SQL-запрос можно запускать на выполнение всего 1 раз.

Директива ALTER- позволяет модифицировать созданный объект. С помощью этого оператора можно добавлять и удалять столбцы таблицы.

Синтаксис оператора ALTER TABLE для добавления новых столбцов выглядит следующим образом:

ALTER TABLE Имя_таблицы

ADD COLUMN

Имя_столбца Тип_данных (размер_данных),

ADD COLUMN

Имя_столбца Тип_данных (размер_данных),

…….

ADD COLUMN

Имя_столбца Тип_данных (размер_данных);

Синтаксис оператора ALTER TABLE для удаления существующих столбцов из таблицы выглядит так:

ALTER TABLE Имя_таблицы

DROP COLUMN

Имя_столбца;

В этих директивах:

ALTER TABLE –зарезервированные слова SQL-запроса,

Имя_таблицы – имя той таблицы, которая будет изменяться,

ADD COLUMN, DROP COLUMN - зарезервированные слова SQL-запроса,

ADD –создание столбца,

DROP – удаление столбца,

Имя_столбца – имена столбцов, с которыми будут выполняться действия (создание, удаление),

Тип_данных – для каждого вновь создаваемого столбца указать тип данных. Для некоторых типов данных указать размер.

Удаляя столбцы из таблицы, следует иметь в виду, что все данные в удаляемом столбце тоже удаляются. Нельзя удалить из таблицы все столбцы, необходимо оставить хотя бы один столбец.

Пример.

Запрос 1.

CREATE TABLE Tabl1 Tovar Char (32);

Запрос 2.

ALTER TABLE Tabl1

ADD COLUMN Zena Integer,

ADD COLUMN Kolichestvo Integer,

ADD COLUMN Summa Integer;

Запрос 3.

ALTER TABLE Tabl1

DROP COLUMN Summa;

Директива DROP TABLE-

позволяет удалять таблицу. Синтаксис следующий:

DROP TABLE Имя_таблицы,

где DROP TABLE – зарезервированные слова,

Имя_таблицы – имя удаляемой таблицы.

После запуска такого SQL-запроса таблицы удаляются безвозвратно вместе со всеми данными, накопленными в них.

Пример.

DROP TABLE Tabl1.

Директива INSERT-

позволяет вставлять в таблицу новые записи. Синтаксис следующий:

INSERT INTO Имя_таблицы (Имя_столбца1, Имя_столбца2, …)

VALUES (значение1, значение2, …);

INSERT INTO - зарезервированные слова,

Имя_таблицы – имя той таблицы, в которую будет добавлена новая запись,

Имя_столбца1,…- в круглых скобках перечисляется список столбцов, имеющихся в таблице,

VALUES - зарезервированное слово,

Значение1, значение2,…- в круглых скобках перечисляются значения , которые присвоены перечисленным ранее столбцам.

Пример.

Запрос 1. CREATE TABLE Сотрудники

(Name Char (16), Famil Char (24), Vozrast Integer, Ind_Cod Integer);

Запрос 2.INSERT INTO Сотрудники

(Name, Famil, Vozrast, Ind_Cod);

Запрос 3.INSERT INTO Сотрудники

(Name, Famil) Values (Гаджи, Юнусов);

В каком порядке перечисляются столбцы, в таком порядке и должны перечисляться присваиваемые значения.

Т.е. после выполнения этих команд получим

Name	Famil	Vozrast	Ind_Cod
Ахмед Гаджи	Магомедов Юсупов	20	100

Возможен и следующий синтаксис:

INSERT INTO Имя_таблицы

VALUES (Значение1, значение2, …);

В нем пропускается список столбцов и сразу следует список значений. При такой записи сначала анализируется таблица, в которую добавляется запись, а затем по порядку (начиная с 1-го столбца) подставляются значения. Количество значений должно быть равно количеству столбцов в таблице.

Директива DELETE (удалить)-

имеет следующий синтаксис:

DELETE FROM Имя_таблицы

[WHERE Условия_поиска],

где DELETE FROM – зарезервированные слова SQL-языка,

Имя_таблицы – имя той таблицы, из которой удаляются записи,

WHERE– зарезервированное слово SQL-языка, которое означает, что следующее за ним выражение определяет условия поиска записей в таблице,

Условия_поиска – это выражение, определяющее для директивы DELETE критерии записей, которые необходимо удалить.

Директива DELETE удаляет записи из таблицы безвозвратно, без права восстановления. Если в этой директиве отсутствует слово WHERE и условия поиска, то из таблицы удаляются все записи.

Пример.

Создадим таблицу со столбцами ID, Famil, Name, tel_rab, tel_dom, gruppa. Для этого

CREATE TABLE Svedeniya

(ID integer, Famil Char(16), Name Char(24), tel_rab Char(8), tel_dom Char(8), gruppa integer);

1)Удалим из таблицы всех студентов из группы 5:

DELETE FROM Svedeniya

WHERE gruppa=5.

2)Удалить из таблицы всех студентов, чья фамилия Магомедов:

DELETE FROM Svedeniya

WHERE Famil =’Магомедов’;

3)Чтобы полностью очистить таблицу от всех записей, надо:

DELETE FROM Svedeniya.

Директива SELECT (выборка данных)

имеет следующий синтаксис:

SELECT Столбец1 [Столбец2, Столбец3, …]

[INTO Новая таблица]

FROM Таблица1 [Таблица2, Таблица3, …]

[WHERE Условия_поиска]

[GROUP BY выражение (условие) группировки]

[ORDER BY выражение упорядочивания],

где *SELECT* - зарезервированное слово SQL-языка,

Столбец1, Столбец2,… -это те столбцы, которые будут отображаться в результирующей таблице запроса,

INTO - зарезервированное слово SQL-языка, которое предписывает директиве SELECT сохранить результирующий набор данных в Новую таблицу,

FROM - зарезервированное слово SQL-языка, после которого следует определение объединения таблиц, из которых выбираются запрашиваемые данные,

WHERE - зарезервированное слово SQL-языка, которое указывает выбирать только те данные, которые удовлетворяют условиям поиска,

GROUP BY - зарезервированное слово SQL-языка, выполняющее группировку данных в результирующем наборе записей, в соответствии с условиями группировки,

ORDER BY - зарезервированное слово SQL-языка, выполняющее упорядочивание данных в результирующем наборе данных в соответствии с Выражением_упорядочивания.

Директива SELECT – служит для отбора данных из одной или нескольких таблиц и предоставление их в виде результирующего набора записей.

Пример.

Создадим таблицу, содержащую *Номер, Фамилию, Имя, Дату_рождения, Город, Группу.*

```
CREATE TABLE Student
   (Номер Integer,
   Фамилия Char (30),
   Имя Char (15),
   Дата_рождения Date,
   Город Char (30),
   Группа Integer);
```

Если в запросе необходимо выбрать значения всех столбцов, то можно вместо перечисления всех столбцов использовать символ «*».

Например, SELECT* FROM Student;

Если из таблицы надо выбрать только столбцы Фамилия и Имя, то

SELECT Фамилия, Имя FROM Student;

Чтобы выбрать студентов махачкалинцев из группы 6, надо создать запрос

SELECT* FROM Student

WHERE Город =’Махачкала’ and Группа=6;

WHERE определяет, какие строки из указанной таблицы должны быть выбраны. В условиях могут использоваться операции сравнения (= равно, > больше,< меньше, >= больше или равно, <= меньше или равно, <> не равно, а также логические операторы AND, OR, NOT).

Выбрать не махачкалинцев

SELECT Город FROM Student;

получим таблицу, в которой встречаются одинаковые строки.

Для исключения из результирующей таблицы повторяющихся значений используют ключевое слово DISTINCT (отличный).

Поэтому предыдущий запрос правильно писать в виде:

SELECT DISTINCT Город FROM Student.

Противоположное действие оказывает ключевое слово ALL (все), т.е. при его использовании повторяющиеся строки включаются в состав выходных данных. Этот режим действует по умолчанию, поэтому в запросах для этих целей ALL практически не используется.

ORDER BY – позволяет упорядочивать данные в результирующей таблице. При этом можно задать возрастающую (ASC) и убывающую последовательность (DESC) для каждого столбца.

Запрос, сортирующий по убыванию Фамилий студентов:

SELECT* FROM Student

ORDER BY Фамилия DESC;

ORDER BY может использоваться с GROUP BY для упорядочения групп записей. При этом ORDER BY в запросе всегда должен быть последним.

SELECT Фамилия, Группа FROM Student

GROUP BY Группа

ORDER BY Фамилия.

Агрегирование и групповые операции.

Агрегирующие функции позволяют получать из таблицы сводную (агрегированную) информацию, выполняя операции над группой строк таблицы. Для задания в SELECT-запросе агрегирующих операций используются следующие ключевые слова:

COUNT – определяет количество строк или значений поля, выбранных с помощью запроса;

SUM – вычисляет сумму выбранных значений;

AVG – вычисляет среднее из выбранных значений;

MAX, MIN – вычисляет наибольшее и наименьшее значение.

В SELECT-запросе эти функции используются аналогично именам полей, при этом имена полей указываются как аргументы в скобках ().

Из предыдущего примера найти количество человек в таблице, т.е. подсчитать количество строк в таблице.

SELECT COUNT (*) FROM Student;

Найти количество студентов в каждой группе. Для этого надо сгруппировать все записи в столбце Группа и к каждой группе записей применить агрегирующую функцию:

SELECT COUNT (Фамилия) FROM Student

GROUP BY Группа;

Найти самого старшего человека в таблице:

SELECT Фамилия, MIN (Дата_рождения) FROM Student;

При необходимости часть сформированных с помощью GROUP BY групп может быть исключена с помощью предложения HAVING.

HAVING определяет критерий, по которому группы следует включать в выходные данные (аналогично WHERE, которое осуществляет это для отдельных строк).

Пример. Пусть имеем таблицу с ФИО_заказчика, Сумма_заказа, Дата_заказа. Выбрать максимальные суммы заказов каждого заказчика, которые больше 1000р.

```
SELECT ФИО_заказчика, Дата_заказа, MAX (Сумма_заказа)
FROM Таблица1
GROUP BY ФИО_заказчика, Дата_заказа
HAVING MAX (Сумма_заказа)>1000;
```

Директива UPDATE

изменяет данные в уже существующих записях существующих таблиц.

Синтаксис следующий:

```
UPDATE Имя_таблицы
SET Столбец1=Значение1 [Столбец2=Значение2, …]
WHERE Условия_поиска;
```

где Имя_таблицы – имя той таблицы, в которой должны быть изменены данные,

Столбец1, … , столбец n – имена существующих в таблице столбцов, которым присваиваются новые значения,

Значение 1,…, Значение n – SQL-выражения, возвращающие значения, присваиваемые соответствующим столбцам,

WHERE Условия_поиска – означает , выражение определяет выбор записей в таблице, в которых будут изменены данные.

Контрольные вопросы.

1. Перечислите основные способы упорядочения лексики в словарях.
2. Какие из перечисленных связей можно охарактеризовать как парадигматические? (цвет есть свойство; лошади кушают овес; чашка – это емкость; вчера он опоздал на работу; арестован еще один торговец наркотиками; медь - металл)
3. Представьте синтаксическую структуру данного простого предложения сокращенной скобочной записью (а там, где возможно, сохраняйте порядок следования слов в тексте.)

4. Письменный лексикон как простейшая составляющая лингвистических ресурсов.
5. Письменные текстовые массивы.
6. Фонетические лингвистические ресурсы.
7. Опишите возможности СУБД MS Access.
8. Какие объекты входят в состав файла базы данных MS Access?
9. Какие ограничения на имена полей, элементов управления и объектов ействуют в MS Access?
10. Чем отличаются режимы работы с объектами базы данных в MS Access: оперативный режим, режим конструктора?
11. Опишите, какие типы данных могут иметь поля в MS Access. Каков их предельный размер?
12. Каково назначение справочной системы MS Access? Чем отличается поиск подсказки на вкладках: Содержание, Мастер ответов и Указатель?
13. Что такое выражения в MS Access? Какие бывают выражения и для чего они используются?
14. Какие особенности в записи различных операндов выражений: имя поля, число, текст?
15. Каково назначение построителя выражений?
16. С какой целью выполняется проектирование базы данных и в чем оно заключается?
17. Какие операции с данными в таблице базы данных вы знаете?
18. Каково назначение сортировки данных в таблице? Какие бывают виды сортировки?
19. Что такое фильтр? Каковы особенности расширенного фильтра?
20. Зачем в базах данных используются формы? Какие разделы имеются в форме и зачем они предназначены? Какими способами можно создать форму?

21. Какие элементы управления могут иметь объекты базы данных: форма, отчет, страница доступа к данным?
22. Что такое запрос? Каково отличие запроса-выборки и запроса с параметром? Какими способами можно создать запрос?
23. Опишите назначение языка SQL.
24. Для чего нужен отчет? Какие сведения отображаются в отчете? Какова структура отчета? Какими способами можно создать отчет?
25. Для чего предназначены страницы доступа к данным? Какие компоненты имеет страница доступа к данным?
26. Какие средства используются в СУБД MicrosoftAccess для целей автоматизации операций с объектами баз данных? Чем они отличаются?
27. Как можно автоматически выполнить макрокоманду или набор макрокоманд при открытии базы данных?
28. Зачем устанавливается связь между таблицами? Какие типы связей между таблицами возможны?
29. Зачем для связанных таблиц используется механизм поддержки целостности данных? В чем заключается его действие?
30. Какие возможности предоставляются пользователю для изменения настроек и параметров СУБД Access?

Тема 4. Лингвистические информационные ресурсы.

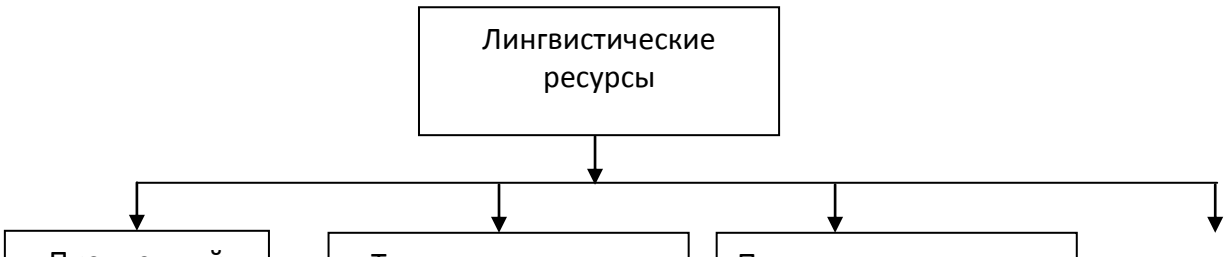
1. Лингвистические информационные ресурсы. Основные понятия
2. Письменный лексикон как простейшая составляющая лингвистических ресурсов.
3. Терминологические словари и банки данных.
4. Письменные текстовые массивы.
5. Фонетические лингвистические ресурсы.

1. Основные понятия

Лингвистические информационные ресурсы — это одна из составляющих частей информационных ресурсов. Под ***информационным ресурсом*** понимают некоторый интеллектуальный ресурс, результат коллективного творчества. К пассивным формам информационных ресурсов относят книги, журналы, газеты, словари, энциклопедии, патенты, базы и банки данных и т. п. Активные формы включают алгоритмы, модели, программы, базы знаний.

Лингвистические информационные ресурсы — это множество определенным образом организованных речевых и языковых данных, находящихся на машинных носителях информации и используемых в различных сферах практической деятельности (образовании, промышленности, экономике, культуре, искусстве, издательстве и т.п.). Термин «лингвистические ресурсы» впервые был использован итальянским ученым Антонио Замполли в 1992 году. Он употребил его в докладе «Структура Европейского агентства языковых технологий», который сделал на конференции, посвященной проблемам создания основы для развития индустрии европейских языков. Выбор этого термина был связан с необходимостью выразить идею о том, что большие массивы лингвистических данных и описаний, используемые для создания и развития эффективных систем обработки текста и речи, играют такую же существенную, фундаментальную роль, как и железные дороги, автомобильные шоссе, электросети (электроэнергия), средства коммуникации для промышленности и экономического развития страны.

С современной точки зрения пассивные лингвистические информационные ресурсы включают следующее множество лингвистических данных.



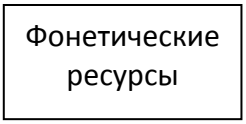


Рис. 5. Лингвистические ресурсы.

В самом общем виде лингвистические ресурсы — это своеобразные лингвистические базы данных, которые можно обновлять (добавлять новые данные, исключать или изменять старые) и в которых можно искать ту или иную информацию. Лингвистические ресурсы необходимы как пользователям ПК, так и различным компьютерным системам, связанным с обработкой текста и речи. В частности, они используются для распознавания речи и анонимных текстов, реферирования, аннотирования и перевода текстов, построения диалоговых систем, автоматического анализа текста, синтеза речи и текста и т.д..

Проблемам создания лингвистических ресурсов ежегодно посвящается большое число международных и национальных научных конференций во всем мире. Создан ряд крупных организаций, объединяющих исследователей десятков стран, занимающихся разработкой лингвистических ресурсов. Наиболее известными из них являются LDC (Linguistic Data Consortium) (США), ELRA (European Language Resources Association) и TELRI (Trans European Language Resources Infrastucture) (Европа). Однако лингвистические традиции стран — участниц этих объединений, недостаточность финансирования, разобщенность участников приводят к тому, что лингвистические ресурсы, созданные отдельными коллективами одной страны, не всегда могут быть использованы в других

странах. Все это ставит перед разработчиками лингвистических ресурсов ряд проблем, важнейшими из которых являются следующие:

1. Разработка единых стандартов создания ресурсов.
2. Разработка способов защиты лингвистических ресурсов от несанкционированного доступа.
3. Создание единых экспертных требований.
4. Планирование единой стратегии разработки лингвистических ресурсов.
5. Создание многофункциональных лингвистических ресурсов большого объема для использования в разных странах.

2. Письменный лексикон как простейшая составляющая лингвистических ресурсов

Как видно рис.5, письменный лексикон представлен лексическими ресурсами двух типов: 1) одноязычными и 2) многоязычными лексиконами (словарями).

В самом общем смысле **словарь** — это справочная книга, которая содержит слова (морфемы, словосочетания, идиомы и т.п.), расположенные в определенном порядке (различном в разных типах словарей). В нем может содержаться толкование значения описываемых единиц, а также различная информация о них. Как видно из этого определения, существуют различные типы словарей. Среди них выделяют:

а) по лексикографической форме основной единицы словаря:

1) для единицы, меньшей, чем слово:

- словари корней;
- словари морфем (приставок, суффиксов, окончаний);
- словари «-буквенных сочетаний;

2) для единицы, эквивалентной слову:

- словари словоформ;
- словари слов;

3) для единицы, большей, чем слово:

- словари словосочетаний;
- фразеологические словари;
- словари цитат;

б) по специфике отбираемой лексики:

1) словари, отражающие некоторые тематические и стилевые пласты лексики:

- терминологические словари;
- диалектные словари;
- словари просторечий;
- словари арго (табуированной лексики);
- словари языков писателей;

2) словари, содержащие различные разновидности слов:

- словари неологизмов;
- словари архаизмов;
- словари редких слов;
- словари сокращений;
- словари иностранных слов;
- словари имен собственных;

в) по способу описания основных единиц словаря выделяют словари, раскрывающие отдельные аспекты таких единиц и отношений между ними:

- этимологические словари;
- словообразовательные словари;
- грамматические словари;
- орфографические словари;
- орфоэпические словари (правила произношения);
- синонимические словари;
- антонимические словари;
- паронимические словари (содержат слова с частичным звуковым

сходством при их семантическом различии, например: *главный — заглавный, выплата — оплата* и т. д.);

- частотные словари;

- словари рифм;

- словоуказатели;

- словари созвучий;

г) по расположению материала в словарях:

- идеографические словари, содержащие не слова, а рисунки со смыслом (например, глаз с капающей слезой обозначает «горе»);

- аналогические словари (в них слова располагаются не по алфавиту, а по смысловым ассоциациям — *мебель: стол, стул, кровать, диван, кресло* и т.п.);

- обратные словари (в них слова располагаются по алфавиту конечных букв);

д) по эпохе функционирования слова:

- исторические словари;

е) по назначению (адресату):

- словари трудностей какого-либо языка;

- словари ошибок;

- учебные словари.

Любой словарь может быть представлен в виде двухмерной реляционной базы данных, где по вертикали расположены единицы лексикона — части слов, слова, словосочетания — экземпляры объекта «СЛОВАРЬ», а по горизонтали — их некоторые характеристики (данные), упомянутые в приведенном выше определении понятия «словарь».

Единица лексикона	Характеристика, X				
	XI	X2	X3	X4	…..
……	……	…….	…….	…….	…….

Простейшей лингвистической базой данных может служить *частотно-алфавитный словарь словоформ* какого-либо текста. Каждый экземпляр объекта (слово) описывается в нем одним данным — абсолютной частотой употребления слова в некотором тексте.

Более сложную организацию имеет *словоуказатель.* В нем кроме абсолютной частоты употребления словоформы в тексте указываются номера страниц и строк на странице, где встретилась данная словоформа. Словоформы располагаются, как правило, по алфавиту. Например, фрагмент словоуказателя к рассказу М. Шолохова «Судьба человека» выглядит так, как показано в таблице 5.

Таблица 5

№ п/п	Частота, *F*	Словоформа	Страни-ца/строка	Страни-ца/строка	Страни-ца/строка	
56	1	*батареей*	51/29	51/33		
57	2	*батарей*	35/13	35/9		
58	3	*батарея*	34/31	55/11		
59	1	*батареями*	57/9	52/38		
60	2	*бегает*	41/4			
…	…..	……..		…….		

В других типах словарей в качестве характеристик XI, Х2, ХЗ, ... выступают толкования слов, их грамматические или стилистические особенности и т.д.

Еще более сложным типом словарей являются ***конкордансы.*** **В** них каждая словоформа текста характеризуется не только численными показателями (частотой, номером страницы, номером строки и т.д.), но и некоторым контекстом, в котором она употреблена. Как правило, этот контекст состоит из трех предложений: предложения, в котором встретилась словоформа; предложения, стоящего перед основным предложением, и предложения, стоящего после него. Предполагается, что контекст такого объема является достаточно полным и содержит внутри себя отрезок, законченный в смысловом отношении. Но возможны

контексты и на уровне отдельных слов и словосочетаний. Конкордансы — ценный материал для научных исследований. Они помогают различать значения слов, широко используются в качестве примеров в других словарях (толковых, синонимических, антонимических, словарях писателей и поэтов и т.д.).

На основе конкордансов составляются ***контекстологические словари.*** Их теоретической основой является теория детерминант. В соответствии с ней каждое значение (перевод) многозначного слова определяется в контексте другими словами (детерминантами), с которыми сочетается исходное слово. Детальные сведения об устройстве и функционировании таких словарей приведены в работах.

Если говорить об ***одноязычных словарях*** как части лингвистических ресурсов, находящихся на машинных носителях информации, то наибольшее их число разрабатывается в настоящее время в рамках организации ELRA. Так, по ее URL-адресам http://www.icp.grenet.fr/ELRA/home.html и http://www.elda.fr можно заказать Европейский португальский лексикон (ELRA-L0033 LusoEX European Portuguese Lexicon), содержащий 61000 основ (лемм) и 1600 соответствующих формообразующих парадигм. Там же можно найти японский компьютерный словарь (ELRA-L0036), в который включено 260 000 японских слов, английский компьютерный словарь (ELRA-L0037), состоящий из 190000 английских слов, и ряд других словарей. Известны созданные на машинных носителях французские словари общеупотребительной и специализированной лексики.

В *многоязычных словарях* дается перевод значения слова исходного языка на один или несколько иностранных языков. В качестве лингвистических информационных ресурсов чаще всего используются двуязычные словари. Наиболее известными из них являются серии электронных словарей МУЛЬТИЛЕКС, LINGVO и КОНТЕКСТ. Все они построены на основе известных бумажных словарей и служат для быстрого

поиска переводных эквивалентов слов, уточнения перевода, поиска синонимов и целого ряда других операций. По указанным выше URL-адресам организация ELRA предлагает целую серию двуязычных словарей. Например, японско-английский словарь (ELRA-M0023), включающий 230000 японских слов с их переводами на английский язык, англо-японский словарь, содержащий 160000 английских слов и соответствующих японских переводных эквивалентов.

Энциклопедии — это словари, содержащие характеристики не слова как такового, а обозначенного им предмета, факта или явления. Термин «энциклопедия» впервые был употреблен в 1541 году и обозначал набор систематизированных отрывков, описывающих различные разделы человеческого знания и искусства. Описание некоторого предмета или понятия в энциклопедии осуществляется в рамках словарной статьи. Например, статья, описывающая такой предмет, как *буксир,* в однотомном Советском энциклопедическом словаре выглядит так:

«Буксир (от гол. *boegseren* — 'тянуть') — самоходное судно для вождения (буксировки) несамоходных судов, плотов и т.п. Подразделяются на буксировщики (для вождения судов на тросе), кантовщики (для швартовки судов к причалу), толкачи (для буксировки судов толканием), спасатели (для оказания помощи судам в открытом море и их буксировки в порт-убежище)».

Чаще всего в энциклопедиях объясняются понятия, выражаемые именем существительным или именем собственным. Научные понятия часто выражаются именными словосочетаниями. В настоящее время энциклопедии являются важным ресурсом для создания баз знаний различных интеллектуальных задач. Энциклопедии могут использоваться в различных компьютерных моделях понимания текста, когда контекст не дает возможности распознать неоднозначную ситуацию.

Существует достаточно большое число различных энциклопедий на машинных носителях информации. Наиболее известна среди них энциклопедия «Britannica». Она включает 82000 статей и 700 дополнительных материалов, опубликованных с 1768 года. Не менее известны французские энциклопедии «Tons les savoire du Monde», «Le monde sur CD-ROM», «Versailles» и др.. На русском языке изданы «Большая Энциклопедия Кирилла и Мефодия» и «Иллюстрированный Энциклопедический Словарь». Основная особенность электронных энциклопедий заключается в том, что их статьи содержат большое число иллюстраций, видео и звук.

Тезаурус — принципиально иной тип словарей. В нем в явном виде указаны семантические связи между определенной частью лексических единиц. Как правило, такие словари строятся для текстов достаточно узкой проблемной области: вычислительной техники, музыки, кораблестроения, сельского хозяйства и т.д. При этом в простейшем случае между лексическими единицами тезауруса могут указываться следующие типы семантических связей*:* 1) синонимические; 2) антонимические; 5) родовые отношения; 4) видовые отношения, 5) ассоциативные отношения; 6) вхождение в состав более крупной семантической единицы; 7) состав (из более простых семантических единиц) и т.п. Например, в «Тезаурусе по теоретической и прикладной лингвистике» слово *предложение* представлено так *:*

ПРЕДЛОЖЕНИЕ

Синонимы: предикативная синтагма, фраза

Родовое понятие: высказывание, коммуникативная единица *Видовое понятие:*

- главное предложение, придаточное предложение, ввод ное предложение;

- простое предложение, сложное предложение;

- повествовательное предложение, восклицательное предложение,

вопросительное предложение;

- полное предложение, неполное предложение;

- распространенное предложение, нераспространенное предложение.

Вхождение в состав более крупной единицы: сверхфразовое единство, текст. *Состав* (из более простых семантических единиц):

- главные члены предложения, второстепенные члены предложения;

- синтаксическая структура, синтаксическая группа, синтаксическая конструкция;

- знаки препинания;

- пропозиция, модальность.

Главное слово тезауруса называют дескриптором (слово *предложение),* а все нижестоящие слова и словосочетания — ключевыми словами (или словосочетаниями).

Первый тезаурус (тезаурус Роже) как средство «для облегчения выражения мыслей и помощи при написании сочинений» был создан в 1852 году в Англии. Особенно широко тезаурусы использовались в 70-х годах XX века, когда с целью поиска фактических данных об оборудовании (станках, автомобилях и т. п.) или технической литературы активно разрабатывались информационно-поисковые системы. Особой популярностью в те годы пользовался «Тезаурус научно-технических терминов». Находят применение тезаурусы и в наши дни. Так, в Москве в 1996 году был создан тезаурус по маркетингу и общеязыковой тезаурус русского языка. Он содержит 1600 дескрипторов и около 7000 распределенных по этим дескрипторам русских слов.

3. Терминологические словари и банки данных

Терминологическим словарем (ТС) называется словарь, основной единицей которого является термин. Нет единого подхода к определению этого понятия. В качестве рабочего определения примем следующее*:* **«Термин** — это слово или подчинительное словосочетание, имеющее

специальное значение, выражающее и формирующее профессиональное понятие и применяемое в процессе познания и освоения научных и профессионально-технических объектов и отношений между ними».

В число единиц терминологических словарей в настоящее время входят и номены (номенклатурные единицы, идентификаторы). К ним, как правило, относят географические названия (океанов, морей, озер, гор, долин, городов, сел, полей), названия станков, приборов, аппаратов, фирм, систем, лекарств, медицинских препаратов и т. п. В качестве номенов выступают отдельные слова *(Кавказ),* словосочетания *(Северный Ледовитый океан),* буквенные символы *(IBM),* сочетания слов и символов *(Pentium MMX),* цифры, символы и цифры *(IBM 486)* и т.п..

Существуют различные подходы к классификации терминологических словарей. Так, по тематике, охватываемой лексикой терминологического словаря, различают:

1) многоотраслевые словари, включающие термины, относящиеся к различным научно-техническим отраслям (подъязыкам):машиностроению, судостроению, автомобилестроению и т.п.;

2) отраслевые (тематические) словари, содержащие термины, относящиеся только к одной теме, например вычислительной технике, оптике, органической химии и т.п.;

3) узкоотраслевые словари, содержащие термины по отдельным разделам некоторой более широкой темы. Например, существуют узкоотраслевые словари по принтерам (тема «Вычислительная техника») и т.п.

По цели и назначению различают:

1) ТС, ориентированные на специалистов;

2) учебные ТС;

3) классификаторы;

4) словари (сборники) рекомендуемых терминов;

5) терминологические стандарты и т.д.

В приведенной классификации два первых наименования не требуют особых объяснений.

Классификатор — это словарь терминов с указанием различных типов иерархических отношений между ними. Подобные словари по принципу своей организации схожи со словарями-тезаурусами. Например, в классификаторе «Республика Беларусь» можно выделить следующие термины, которые сами являются классификационными рубриками: «Область», «Район», «Город», «Район в городе», «Улица» и т. п. При компьютерном использовании классификаторов их единицы получают определенные коды.

Терминологические стандарты содержат термины вместе с их вариантами и рекомендациями о возможности употребления в тех или иных видах текстов.

По числу используемых естественных языков выделяют следующие терминологические словари:

1) одноязычные;
2) двуязычные;
3) многоязычные.

По способу упорядочения терминов различают следующие терминологические словари:

1) алфавитные;
2) неалфавитные (тематические);
3) частотные.

В неалфавитных (тематических) терминологических словарях термины расположены гнездами, т.е. сначала указывается заглавное слово, а затем перечень относящихся к нему ключевых слов, расположенных по алфавиту, например:

компьютер	***системный***
дисплей	блок

клавиатура	винчестер
монитор	ОЗУ
мышь	плата
принтер	процессор
системный	
сканер	

Многообразие терминологических словарей объясняет тот факт, что объем информации, закладываемый в словарные статьи ТС, сильно отличается от словаря к словарю. В общем виде каждая статья включает термин и некоторые его характеристики (пометы): в частотных словарях — частоты употребления терминов, в классификаторах — коды понятий, в многоязычных словарях — адреса переводных эквивалентов и т.д. Есть и более сложная организация словарной статьи, типичная для многоязычных терминологических словарей.

Основными источниками для составления словарей служат научные монографии и статьи по соответствующей проблемной области («вычислительная техника», «сельское хозяйство» и т.п.), соответствующие учебники и учебные пособия для вузов, рефераты и аннотации на научные издания, описания изобретений, патенты на изобретения, а также термины, зафиксированные в различных энциклопедиях. Упомянутая выше международная организация ELRA занимается созданием и терминологических словарей. Так, в ее рамках разработан технический терминологический словарь (ELRA-L0041), содержащий 80000 английских терминов и 120000 их японских переводных эквивалентов.

Терминологический банк данных — это некоторое множество терминологических словарей, относящихся к различным темам, определенным образом характеризующим какую-либо науку, производство или вид деятельности (медицину, образование, связь, транспорт и т.п.). Такие банки данных используются сегодня в основном для следующих видов работ:

1) создания новых терминологических словарей (в лексикографических

целях);

2) перевода текстов с одного языка на другой (как человеком, так и с помощью компьютера);

3) создания учебников, учебных пособий, написания диссертаций и т. п. (в исследовательских целях).

4. Письменные текстовые массивы

Письменный текстовый массив или корпус текстов является в настоящее время основным понятием нового направления в лингвистике, называемого *корпусной лингвистикой (corpus linguistics).* Суть этого направления сводится к тому, что достоверные данные о фонетической, морфологической, синтаксической и семантической структуре языка и речи могут быть получены только из достаточно большого массива текстов. Впервые эти идеи были высказаны российским ученым профессором Р. Г. Пиотровским в докладе «Статистическое исследование лексики и грамматики текста с помощью электронно-вычислительной машины», прочитанном в Московском государственном педагогическом институте иностранных языков в 1965 году . Дальнейшее развитие этой идеи, ее теоретическое обоснование и успешное практическое применение показаны в целом ряде крупных работ Р.Г.Пиотровского и его учеников .

В сегодняшнем понимании **корпус текста** — это совокупность текстов, являющаяся достаточной для обеспечения надежных научных выводов о некотором языке, диалекте или ином другом подмножестве языка. Такие письменные совокупности текстов могут быть использованы для решения большого числа лингвистических задач:

1) в лексикографии и лексикологии (для составления различных словарей, определения значений многозначных слов, выявления ассоциативных связей слов в тексте, выделения терминов и терминологических словосочетаний и т.п.);

2) в грамматике (для определения частоты употребления грамматических

морфем в текстах различного типа, выявления наиболее употребляемых типов словосочетаний и предложений, определения значений синонимичных морфологических единиц, частоты употребления классов слов и т.д.);

3) в лингвистике текста (для дифференциации типов текста, создания конкордансов, выявления связи между предложениями в абзацах и между абзацами и т.д.);

4) при автоматическом переводе текстов (для поиска контекстов слов, имеющих несколько переводных эквивалентов, поиска переводных эквивалентов терминологических и фразеологических словосочетаний в параллельных текстах и т.д.);

5) в учебных целях (для выбора цитат, отдельных фрагментов произведений, примеров, используемых в процессе создания учебников и учебных пособий, и т.д.).

Существуют различные подходы к классификации корпусов текстов в зависимости от типа текстов, способов их организации, языка и т. д. С точки зрения их использования лингвистами наиболее значимы следующие виды корпусов текстов.

1. Исследовательские — создаются с целью изучения различных аспектов функционирования языка. Как правило, такие корпусы текстов содержат несколько десятков миллионов словоупотреблений.

2. Иллюстративные — служат для выделения из них лингвистических примеров, подтверждающих те или иные языковые (речевые, текстовые) факты, обнаруженные ранее иными лингвистическими приемами.

3. Статические — содержат тексты какого-то небольшого временного промежутка.

4. В динамические корпусы текстов включают письменные источники большого временного периода. Они предназначены для проведения различных диахронических исследований.

5. Корпусы параллельных текстов состоят из множества текстов-оригиналов, написанных на каком-либо исходном языке, и текстов-переводов

этих исходных текстов на один или несколько других языков. Это неоценимый материал для проведения сравнительно-сопоставительных исследований и для обучения переводу человека и компьютера.

Входящие в корпус тексты могут быть представлены в нем следующим образом:

1) в виде текстового архива в исходной форме, переносимой с письменных источников;

2) в виде банка данных, при этом тексты и их единицы предварительно определенным образом размечаются;

3) в виде базы данных информационно-поисковой системы, в этом случае тексты записываются в специальном формате, предназначенном для поиска текстов, их единиц и последующей статистической обработки.

Важной особенностью корпуса текстов является то, что это не просто множество случайным образом объединенных текстов того или иного языка. При его создании возникает целый ряд проблем. Основными из них являются следующие:

1. Что должно являться основной единицей корпуса текстов?
2. Каков должен быть объем корпуса текстов (сколько единиц он должен содержать)?
3. Какие письменные текстовые источники должны быть представлены в корпусе текстов и в каком количестве?
4. Из какой исходной языковой области должны быть выбраны тексты, включаемые в состав корпуса?

Первые ответы на эти вопросы были даны в многочисленных исследованиях учеников профессора Р.Г.Пиотровского в 1965-1980 годах. Именно тогда были впервые использованы различные статистические приемы для оценки генеральной совокупности выборки, объема выборки, порции выборки (элементарной выборки) и т.п.

С сегодняшней точки зрения основной единицей корпуса текстов могут быть ***словоупотребления*** (обычно их называют словами), ***основы***

(корни, леммы) и ***предложения.*** Объем создаваемого корпуса текстов в принятых единицах зависит от целей создания. Он может быть небольшим при изучении частоты употребления букв, буквосочетаний, звуков, звукосочетаний. Гораздо большим он должен быть при изучении лексики, морфологических явлений. Неизмеримо большее количество единиц требуется при изучении синтаксических или стилистических особенностей текстов. Создаваемые сегодня корпусы текстов являются многофункциональными (т. е. могут быть использованы для изучения различных языковых явлений) и содержат миллионы слов, десятки тысяч лемм, тысячи предложений. Так, в апреле 2000 года по URL-адресам http://www.icp.grenet.fr/ELRA/home.html и http://www.elda.fr организация ELRA предлагала корпус французских текстов (ELRA-WOO20 PAROLE French Corpus), содержащий 20093099 слов. Она же предлагала корпус португальских текстов (ELRA-WOO24 PAROLE Portuguese Corpus), включающий 3000000 слов. По тому же URL-адресу можно найти португальский лексикон (ELRA-LOO33 LUSOLEX European Portuguese Lexicon), содержащий 61000 лемм и 1600 соответствующих морфологических парадигм.

Более трудной и менее определенной является задача конкретизации качественного состава корпуса текстов. При ее решении необходимо определить следующее.

1. Тексты каких функциональных жанров включать в корпус текстов (художественную прозу, драму, стихи, научные тексты, газеты, журналы, технические описания и т.д.)?

2. Тексты каких временных промежутков включать в корпус текстов (современные, 10-летней давности, 50-летней давности, древние и т.п.)?

3. Включать тексты только литературного языка или диалектические источники?

При решении этой задачи разработчики корпуса текстов обычно используют консультации специалистов по языкознанию и

лингвостатистике либо метод анкет. Исходя из своего опыта исследований, специалисты определяют общий объем корпуса текстов, время издания текстов, число текстов и размер элементарной выборки, жанры отбираемых текстов и их количество, число элементарных выборок из каждого жанра. Так, при создании одного из первых корпусов текстов, корпуса Брауна (The Brown Standard Corpus of American English), группа консультантов-ученых определила его объем в 1000000 словоупотреблений. Было решено, что он должен состоять из 500 текстов по 2000 словоупотреблений каждый. Тексты должны быть взяты из произведений американских авторов, изданных в США в 1961 году. При этом было рекомендовано отобрать 15 письменных жанров: 9 - информативная проза и 6 — художественная проза. Из каждого жанра было сделано от 6 до 80 элементарных выборок.

Метод анкет в сочетании с опытом специалистов был использован при создании корпуса текстов The American Heritage Intermediate Corpus. Специалисты, ориентируясь на заданное время создания корпуса, определили его объем в 5000000 слов (словоупотреблений) и рекомендовали включить в него лексику из 22 разделов (жанров) детской и юношеской литературы на английском языке. Для конкретизации текстов в 221 школу США были разосланы анкеты с просьбой указать, какие тексты желательно включить в корпус. После изучения анкет был составлен список из 19000 названий книг. Из этого множества было отобрано 1045 текстов. На их основе было составлено 10000 элементарных выборок по 500 словоупотреблений каждая.

В упомянутый выше корпус португальских текстов, предлагаемый ELRA, вошли:

1) тексты газет (за 1996—1997 гг. 3 названия) — 65 %;

2) художественные книги (12 названий) — 20%;

3) периодические журналы (7 еженедельных изданий одного названия) — 5 %;

4) периодические журналы (8 названий) — 10%.

Последняя из четырех основных проблем, возникающих при создании корпуса текстов, связана с определением исходного объема языковой области, из которой должны быть сделаны выборки.

Она решается эмпирически, путем проведения предварительных экспериментов.

Как показали результаты использования корпусов текстов для практических исследований, многие лингвистические задачи с их помощью не могут быть решены. Так, во многих языках нельзя установить принадлежность слова предложения к тому или иному грамматическому классу (имени существительному, глаголу и т. п.), что, в свою очередь, не позволяет определить частоту употребления грамматических классов слов, структуры предложений на уровне классов слов (частей речи), а значит, и употребительность таких структур. Без специальной подготовки текста не разрешимы омография морфем и слов, полисемия и целый ряд других важных задач. Поэтому в последние годы стали создаваться ***таггированные (размеченные) корпусы текстов*** (от англ. *tag*— 'индекс, помета'). Все слова такого корпуса получают некоторые буквенные или цифровые индексы, которые обозначают их грамматические, лексические, семантические или структурные признаки. Таких индексов может быть несколько. Принципы разработки подобных индексов (кодов) и их практического использования впервые также были предложены Р. Г. Пиотровским и его учениками .

Например, слова предложения *Этой весной опять расцвела акация* получили следующий набор индексов *:*

Этой — МЖЕТ21 *весной* — СЖЕТ22 *опять* — Н22 *расцвела* — ГЖЕПЗЗ *акация* - СЖЕИЧ42.

Первый индекс у каждого слова указывает на класс слова (М — местоимение, С — имя существительное, Н — наречие, Г — глагол), второй индекс служит для обозначения рода, третий — числа, четвертый — падежа

(или времени для глагола). Первая цифра в конце кода указывает на число слогов в слове, а вторая — на место ударного слога.

Естественно, «приписывание» таких индексов словам текстов — задача достаточно сложная и дорогостоящая. Для некоторых языков с определенной долей вероятности такое приписывание может быть сделано автоматически или полуавтоматически .

Существует огромное число подходов к разметке текстов и его единиц в корпусе текстов. Для этих целей можно выделить следующие группы признаков:

1) признаки для кодирования отдельного текста;

признаки для кодирования предложения;

2) признаки для кодирования словосочетаний;

3) признаки для кодирования слова;

4) признаки для кодирования морфемы.

К первой группе признаков относятся следующие:

1) название текста (полное и краткое);

2) фамилия, имя и отчество автора;

3) псевдоним автора (если есть);

4) дата рождения автора (и смерти, если автор умер);

5) время создания текста (или его первой публикации);

6) место первой публикации;

7) тип текста (художественный, научный и т.п.);

8) графика текста (кириллица, латиница, книжно-славянская и т.п.);

9) наличие в тексте таблиц, графиков, рисунков;

10) указание на рецензии и критические материалы к тексту;

11) адрес места хранения текста;

12) другая информация, важная для проведения конкретных исследований.

Для кодирования отдельных предложений текста используются следующие лингвистические признаки:

1) тип предложения (простое или сложное);
2) тип сложного предложения;
3) вид придаточного предложения;
4) число сегментов (формальных групп: группа подлежащего, группа сказуемого и т.п.);
5) структурная формула предложения на уровне классов слов и/или членов предложения;
6) другая информация.

Словосочетание может получить в корпусе текстов следующие признаки:

1) число слов в словосочетании;
2) тип (свободное, идиоматическое, фразеологическое);
3) тип по главному слову (именное, нагольное, предложно-именное и т.п.);
4) тип связи между главным словом и другими словами словосочетания (согласование, управление, примыкание и др.);
5) другая информация.

Слово (словоупотребление) максимально может иметь следующую информацию:

1) признак класса слова;
2) тип слова (простое, сложное, составное);
3) набор грамматических признаков;
4) семантический признак;
5) синтаксический признак;
6) стилистический признак;
7) наличие приставки;
8) наличие суффикса;
9) наличие окончания;

10)структурная формула слова;

11)другая информация.

Для отдельной морфемы слова возможна такая информация:

1) число букв в морфеме;

2) тип морфемы (приставка, основа, суффикс, окончание);

3) семантический тип морфемы («отрицание», «уменьшительность» и т.п.);

4) другая информация.

Так как чаще всего основной единицей корпуса текстов является словоупотребление, то весь возможный объем информации, который может быть получен из таггированных текстов, зависит от того, насколько удачно проведено кодирование каждого отдельного словоупотребления. Пока, к сожалению, не разработаны единые обозначения для кодирования грамматических, семантических и стилистических признаков словоупотребления в создаваемых корпусах текстов.

С опорой на правильно проведенное кодирование всех словоупотреблений предложения во многих случаях может быть проведено автоматическое таггирование самого предложения.

Особая проблема возникает при подготовке корпусов параллельных текстов. Она заключается в установлении соответствий между текстом оригинала и его переводами. Для решения такой задачи используется так называемый метод автоматического выравнивания текстов. Его суть заключается в параллельной сегментации оригинального текста и его перевода по предложениям. При этом могут использоваться шесть возможных соответствий между предложениями обоих текстов.

1.Одно исходное предложение переводится одним предложением.

2.Два исходных предложения переводятся одним предложением.

3.Одно исходное предложение переводится двумя предложениями.

4. Два исходных предложения переводятся двумя предложениями, но внутренние границы этих предложений в тексте оригинала и тексте

перевода не совпадают.

5.Предложение исходного текста не переводится.

6. Предложение в тексте перевода не имеет эквивалента в тексте оригинала.

Одна из проблем, возникающая при создании и эффективном использовании корпусов текстов, связана с созданием удобного для пользователя языка, позволяющего извлекать из корпуса максимум информации. По существу задача сводится к созданию специальной поисковой системы, способной функционировать в сети Интернет. Такая система должна иметь возможность найти все те тексты, предложения, словосочетания, слова и морфемы, которые обладают перечисленными выше признаками.

Еще больший объем информации позволяют получить параллельные корпусы текстов. С их помощью, дополнительно к вышесказанному, можно:

1) автоматически строить двуязычные и многоязычные переводные словари;

2) автоматически создавать и пополнять словари для систем машинного перевода;

3) автоматически устранять полисемию лексических единиц. Это становится возможным в связи с тем, что компьютер может использовать контекстное окружение многозначного слова, по длине превышающее предложение;

4) автоматически переводить терминологические и фразеологические единицы текста;

5) осуществлять полностью автоматический перевод в рамках новых систем машинного перевода, называемых системами с переводческой памятью. Суть данного подхода заключается в том что в памяти компьютера накапливаются корпусы исходных текстов и их переводов, выровненных между собой на различных уровнях.

В процессе перевода такая система пытается отыскать переводимое предложение в массиве исходных параллельных текстов. Если оно найдено в исходном массиве текстов—оригиналов, то система выбирает перевод такого предложения или его части в массиве переведенных текстов.

5. Фонетические лингвистические ресурсы

Как видно из общей структуры лингвистических ресурсов, их составной частью являются также фонетические лингвистические ресурсы. При их создании возникают те же проблемы, что и при создании письменных текстовых массивов. Однако главная трудность создания фонетических лингвистических ресурсов связана с необходимостью транскрибирования устной речи. При этом возникают следующие проблемы:

1. Какой алгоритм использовать для транскрибирования?
2. Учитывать ли индивидуальные особенности произношения?
3. Учитывать ли весь устный текст или его фрагменты?
4. Учитывать ли диалектные варианты произношения слов?
5. Учитывать ли ударения в словах?
6. Учитывать ли просодические признаки произносимых фраз?
7. Отмечать ли слова, которые при прослушивании не распознавались?
8. Отмечать ли в записи для фонетического корпуса паралингвистические явления, сопутствующие речи (паузы, смех, бормотание, кашель и т. п.)?

В настоящее время общепринято, что для создания машиночитаемых фонетических корпусов используется ***транскрипция на основе орфографического представления звуков*** речи с дополнительными знаками, передающими (при необходимости) просодические, паралингвистические и другие особенности произношения.

Несмотря на трудности создания, в мире уже существует много достаточно представительных фонетических корпусов. Так, **в** 70-х годах

XX века в США Х. Далем и его коллегами был создан «Корпус устной речи американского варианта английского языка». Он включал 1000000 словоупотреблений, взятых из записей психоаналитических сеансов. С каждой из 15 кассет, имевшихся **в** распоряжении составителей корпуса, было случайным образом отобрано 225 записей сеансов. Они содержали речь 8 женщин **и** 21 мужчины из 9 городов США. Отобранные записи были транскрибированы на основе стандартной английской орфографии. Диалектные варианты произношения не учитывались. Нераспознанные слова при записи обозначались буквой Z Ударения и другие просодические характеристики речи также не учитывались. В то же время при орфографической записи устной речи **в** качестве специальных комментариев отмечались паузы, смех, вздох, кашель и другие паралингвистические явления. Известен Международный машиночитаемый архив современного английского языка (The International Computer Archive of Modern English — ICAME). В последние годы предлагается коммерческий «Корпус разговорного профессионального американского варианта английского языка» (Corpus of Spoken Professional American English). Он включает 2000000 слов с индексами части речи при каждом слове (его можно найти по URL-адресам www.athelstan.com и www.athel.com').

Существует несколько фонетических корпусов немецкой устной речи. Одним из первых является Фрейбургский корпус. Он создан на базе 820 магнитофонных записей устной немецкой речи тонца 60-х — начала 70-х годов XX века во Фрейбургском отделении Института немецкого языка. Корпус включает записи радиопередач, а также различных конференций, заседаний и других общественных мероприятий. Для транскрибирования было отобрано *122* текста различного объема (от 175 до 16390 словоупотреблений) общей длиной в 600000 словоупотреблений. Для приведения записей в машиночитаемый вид была разработана специальная система транскрипции, опирающаяся на стандартную немецкую орфографию.

По указанным выше URL-адресам организация ELRA предлагает и фонетические корпусы текстов. Так, норвежский фонетический корпус (ELRA-SOO81 Norwegian Speech Dat (II)

FDB-1000) содержит записи телефонных разговоров 1016 носителей норвежского языка (517 мужчин и 499 женщин). Известен датский корпус разговорной речи, содержащий 10000000 слов . Фонетические корпусы текстов широко используются для решения следующих задач:

1) сопоставительного изучения устной и письменной форм языка;

2) изучения грамматических и лексических особенностей устной речи;

3) исследования фонетических особенностей диалектов;

4) построения частотных списков фонем и их сочетаний;

5) изучения акустических свойств речевых единиц и их использования в психолингвистических и лингвистических экспериментах;

6) создания компьютерных систем, распознавания и синтеза устной речи.

Тема 4 . Обработка текстов на естественном языке.

1. Основные задачи обработки текстов.
2. Анализ отдельных слов.
3. Анализ отдельных предложений.
4. Семантический анализ.

1. Основные задачи обработки текстов.

Задачи обработки текстов возникли практически сразу вслед за появлением вычислительной техники. Но, несмотря на полувековую историю исследований в области искусственного интеллекта, огромный скачок в развитии ИТ и смежных дисциплин, удовлетворительного решения большинства практических задач обработки текста пока нет.

Компьютерная лингвистика - это раздел науки, изучающий применение математических моделей для описания лингвистических закономерностей. Эту область можно разделить на две большие части. Первая изучает способы применения вычислительной техники в лингвистических исследованиях - применение известных математических подходов (например, статистической обработки) для выявления закономерностей. Данные закономерности используются второй частью компьютерной лингвистики, изучающей вопросы осмысления текстов, написанных на естественном языке - создание математических моделей для решения лингвистических задач и разработка программного обеспечения, функционирующего на основе этих моделей. Вторая часть компьютерной лингвистики тесно соприкасается с разделом искусственного интеллекта, занимающимся разработкой систем обработки текстов, написанных на естественном языке (NLP, Natural Language Processing).

Наиболее общая схема обработки текстов инвариантна по отношению к выбору естественного языка - независимо от того, на каком языке написан исходный текст, его анализ будет проходить все указанные стадии.

Первые две стадии (разбиение текста на отдельные предложения и слова) практически одинаковы для большинства естественных языков. Единственное, где могут проявляться черты, специфичные для выбранного языка - это обработка сокращений слов и обработка знаков препинания (точнее, определение того, какие из знаков препинания являются концом предложения, а какие нет).

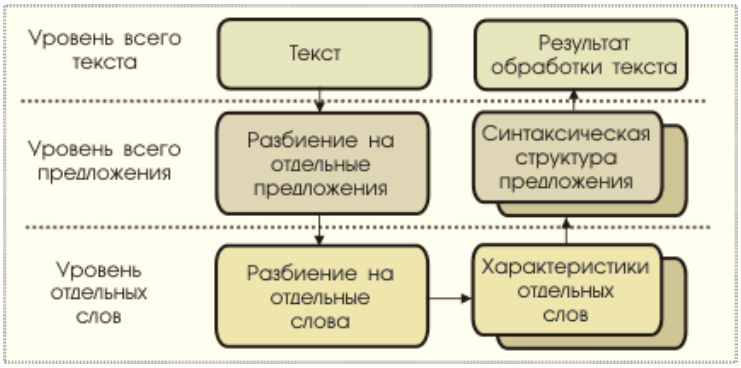


Рис. 6. Общая схема обработки текста

Последующие две стадии (определение характеристик отдельных слов и синтаксический анализ), наоборот, очень сильно зависят от выбранного естественного языка. Последняя стадия (семантический анализ), как и первые стадии, мало зависит от выбранного языка, но, это проявляется только в общих подходах к проведению анализа.

Семантический анализ полностью основывается на результатах работы предыдущих фаз обработки текста, которые всегда специфичны для конкретного языка. Следовательно, способы представления их результатов тоже могут сильно варьироваться, оказывая большое влияние не реализацию методов семантического анализа. Результаты анализа, произведенного на ранних стадиях, могут быть многозначны - для выходных параметров указывается не одно, а сразу несколько возможных значений (например, может существовать несколько способов трактовки одного и того же слова). В таком случае последующие стадии должны определять наиболее вероятные значения результатов ранних стадий анализа и уже на их основе проводить дальнейший анализ текста.

Рассмотрим более подробно каждую из стадий анализа текста после разделения текста на отдельные слова и предложения. К первой стадии (анализ отдельных слов) относится морфологический анализ (определение морфологических характеристик каждого слова: часть речи, падеж, склонение, спряжение и т.д.) и морфемный анализ: приставка, корень, суффикс и окончание. Ко второй стадии относится синтаксический анализ текста, к третьей - различные задачи семантического анализа: поиск фрагментов текста, формализация, реферирование и т.д.

2. Анализ отдельных слов.

В эту стадию обработки текста входит морфологический и морфемный анализ слов. Входным параметром этих методов является текстовое представление исходного слова. Целью и результатом

морфологического анализа является определение морфологических характеристик слова и его основная словоформа. Перечень всех морфологических характеристик слов и допустимых значений каждой из них зависят от естественного языка. Тем не менее, ряд характеристик (например, название части речи) присутствуют во многих языках. Результаты морфологического анализа слова неоднозначны, что можно проследить на множестве примеров.

Существует три основных подхода к проведению морфологического анализа. Первый подход часто называется "четкой" морфологией. Для русского языка он основан на словаре А.А. Зализняка. Второй подход основывается на некоторой системе правил, которые по заданному слову определяют его морфологические характеристики. В противоположность первому подходу, его называют "нечеткой" морфологией . Третий, вероятностный подход, основан на сочетаемости слов с конкретными морфологическими характеристиками . Он широко применяется при обработке аналитичных языков со строго фиксированным порядком слов в предложении и практически неприменим при обработке текстов на русском языке. Рассмотрим каждый из указанных способов морфологического анализа более подробно.

Словарь Зализняка содержит основные словоформы слов русского языка, для каждой из которых указан некоторый код. Известна система правил, с помощью которой можно построить все формы данного слова, отталкиваясь от начальной словоформы и соответствующего ей кода. Помимо построения каждой словоформы, система правил автоматически ставит ей в соответствие морфологические характеристики. При проведении четкого морфологического анализа необходимо иметь словарь всех слов и всех словоформ языка. Этот словарь на входе принимает форму слова, а на выходе выдает его морфологические характеристики.

Данный словарь можно построить на основе словаря Зализняка по очевидному алгоритму: необходимо перебрать все слова из словаря, для

каждого из них определить все возможные их словоформы и занести их в формирующийся словарь. Эта задача решена.

При таком подходе получается, что для проведения морфологического анализа заданного слова необходимо просто найти его в словаре, где уже хранятся точные, "окончательно известные" значения всех морфологических характеристик заданного слова. Возможно, что для одного и того же входного слова будут храниться сразу несколько вариантов значений его морфологических характеристик.

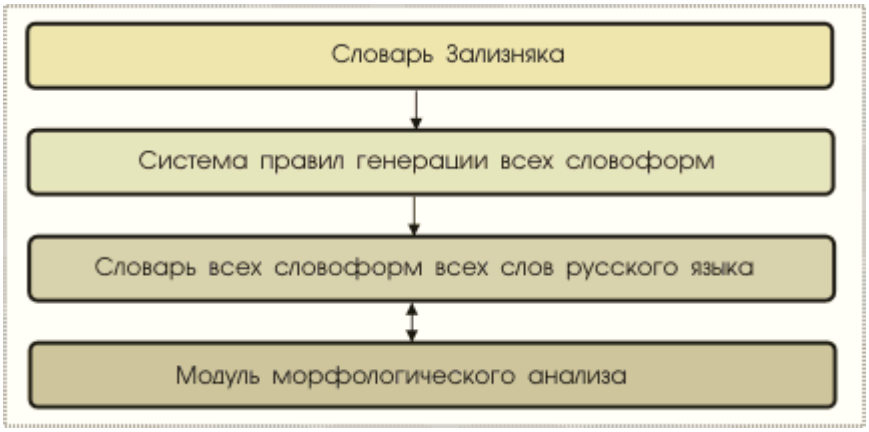


Рис. 7. Морфологический анализ на основе словаря Зализняка

К сожалению, этот способ применим не всегда, поскольку слова, поступающие на вход, могут не входить в словарь всех словоформ. Такая ситуация может возникнуть из-за ошибок ввода исходного текста, наличия в тексте имен собственных и так далее. В случае, когда метод точной морфологии не дает нужного результата, применяется неточная морфология.

Целью морфемного анализа слова является разделение слова на отдельные лексемы: приставки, корни, суффиксы и окончания. В словаре морфем русского языка указанно разделение каждого слова на отдельные части, но не указаны типы каждой из получившихся частей (какая из них является приставкой, какая корнем и т.д.). Известно, что множество всех корней слов русского языка открыто, но множество всех возможных приставок, суффиксов и окончаний - ограничено. Кроме того, известно, что в любом слове сначала идут приставки, затем корни, далее суффиксы и

окончание. Поэтому на основе словаря морфем русского языка можно построить другой словарь, который будет содержать не только разбиение каждого слова на части, но и тип каждой из них. В таком случае, для проведения морфемного анализа слова необходимо обратится к этому словарю. Подобная задача также решена.

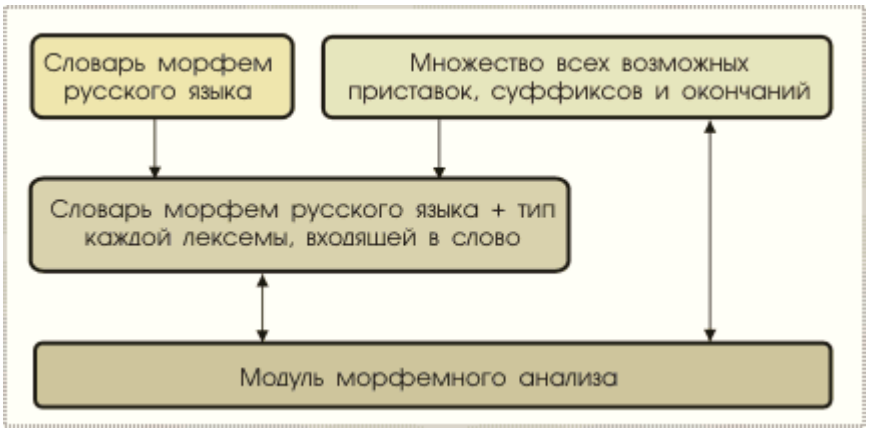


Рис.8. Морфемный анализ

Морфемный анализ не ограничивается обращениями к словарю. В ситуации, когда слово отсутствует в словаре, возможно непосредственное проведение анализа, на основе стандартного строения слов русского языка (приставка - корень - суффикс - окончание) и множества всех приставок, суффиксов и окончаний.

Вернемся к морфологическому анализу слова в той ситуации, когда не удалось определить характеристики слова с помощью методов точной морфологии, но удалось расчленить слово на отдельные части. Наличие тех или иных лексем может определять морфологические характеристики этого слова можно построить систему правил, которая будет опираться на наличие или отсутствие каких-либо частей и выдавать одно или несколько предположений о морфологических параметрах. Данный набор правил можно построить двумя способами. Первый основан на морфемном анализе слов, содержащихся в словаре всех словоформ, и их морфологических характеристик. Рассмотрим эту задачу более формально: известны пары

значений, состоящие из морфемного строения слова и его морфологических характеристик. Это есть не что иное, как "вход" и "выход" системы правил, которая по морфемному строению слова будет определять его морфологические характеристики. Задача построения такой системы правил может быть решена с помощью самообучающейся системы некоторого типа. В данном случае могут быть использованы деревья решений, ILP (Inductive Logic Programming) и прочие алгоритмы.

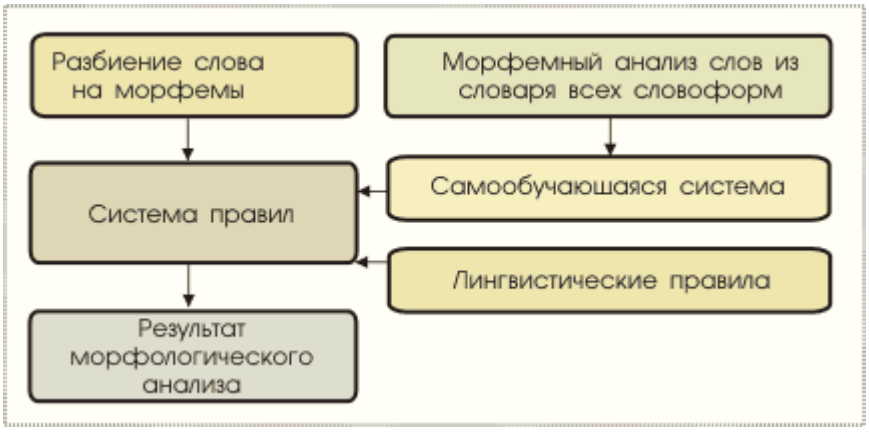


Рис.9. Неточный морфологический анализ

Второй подход заключается в формировании набора правил вручную. По большому счету, реализация такого подхода ни что иное, как написание экспертной системы диагностирующего типа.

Вероятностный способ проведения морфологического анализа слов заключается в следующем. Одна и та же словоформа может принадлежать сразу к нескольким грамматическим классам. Для каждой словоформы определяются все ее грамматические классы, а также вероятность ее отношения к каждому из этих классов. Это происходит на основе некоторого набора документов, где каждому слову предварительно поставлен в соответствие грамматический класс. После этого вычисляются вероятности сочетаний определенных грамматических классов для слов, стоящих рядом (для двоек, троек, четверок и так далее). На основе этих чисел может

проводиться анализ слов, но для него необходимо уже не только само слово, но и стоящие рядом с ним слова.

К методу вероятностного анализа необходимо сделать два важных замечания. Во-первых, он применим только для аналитичных языков, у которых четко фиксирован порядок слов в предложении. Если же порядок слов можно изменять, то данный метод неприменим, поскольку все возможные сочетания грамматических классов будут практически равновероятны. Во-вторых, если первые два способа анализа (точная и неточная морфология) на входе принимают отдельные слова, то вероятностный способ, наоборот, на входе принимает или все предложение или, по крайней мере, несколько слов, стоящих рядом.

3. Анализ отдельных предложений.

После того, как произведен анализ каждого слова, начинается анализ отдельных предложений (синтаксический анализ) для определения взаимосвязей между отдельными словами и частями предложения. Результатом такого анализа является граф, узлами которого выступают слова предложения, при этом, если два слова связаны каким-либо образом, то соответствующие им вершины графа связаны дугой с определенной окраской. Возможные окраски дуг зависят, во-первых, от языка, на котором написано предложение, во-вторых, от выбранного способа представления синтаксической структуры предложения.

При синтаксическом анализе предложений русского языка в качестве возможных окрасок дуг можно использовать вопросы, задаваемые от одного слова к другому. В вершинах графа пишутся слова не в том виде, в котором они встречаются в предложении, а в их основной словоформе. Некоторым словам (например, предлогам) вообще не соответствует ни одна из вершин графа, но эти слова влияют на вопросы, задаваемые от одного слова к другому.

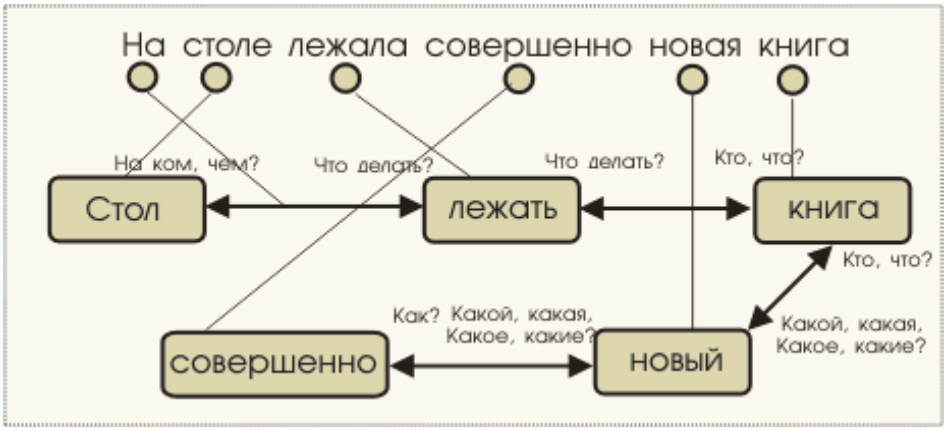


Рис.10. Пример синтаксического анализа предложения

Дуги графа обладают следующими свойствами. Во-первых, они являются двунаправленными, при этом, вопросы, написанные в начале и в конце дуги, различаются между собой. Во-вторых, вопросы соответствуют морфологическим характеристикам основной словоформы слова и не соответствуют той форме слова, которая употреблена в предложении. В-третьих, они не отражают смысловой нагрузки слов, и это свойство проявляется, прежде всего, в том, что не делается различия между одушевленными и неодушевленными предметами. В-четвертых, образующийся граф является деревом. В-пятых, если вершины графа расположить слева направо в порядке, соответствующем словам в исходном предложении, то дуги графа не будут пересекаться.

Возможны и другие способы представления зависимостей между словами в одном предложении (например, разбор предложения по частям и выделение подлежащего, сказуемого и так далее). На основе системы этих зависимостей могут быть разработаны несколько иные способы представления синтаксической структуры предложения.

Перейдем теперь к методам синтаксического анализа предложений. Их можно разделить на две группы: методы с фиксированным, заранее заданным набором правил и самообучающиеся методы. Заметим, что правила представляются не в виде классических продукций ("если …, то …"), а в более удобном виде - в виде грамматик, задающих синтаксис языка.

Исторически, первым способом описания синтаксиса языка были формальные грамматики. Формальные грамматики хорошо изучены и широко применяются при описании формальных языков (например, языков программирования), но они непригодны для описания синтаксиса естественных языков.

Трансформационные грамматики разрабатывались уже специально для задания синтаксических правил построения предложений, написанных на естественном языке. Эти грамматики задаются в виде ориентированного графа состояний, всем дугам которого поставлены в соответствие определенные части речи. В начале работы алгоритм синтаксического анализа находится в некотором начальном состоянии, которому соответствует некоторая вершина графа. Алгоритм просматривает предложение слева направо, анализирует встречающиеся слова и делает переходы из одного состояния в другое в соответствии с выходящими из текущей вершины дугами и очередным словом предложения. Работа заканчивается либо когда предложение просмотрено полностью, либо когда невозможно сделать переход из текущего состояния (нет выходящей дуги с необходимой пометкой).

Основными недостатками трансформационных грамматик является то, что они, во-первых, не способны задавать рекурсивные синтаксические правила. Во-вторых, построение трансформационной грамматики даже для небольшого подмножества языка требует больших усилий. Два описанных способа заключают в себе четко заданную систему правил, согласно которым производится синтаксический анализ предложения. Помимо уже указанных недостатков, они имели еще один большой минус, который заключается в неспособности этих методов анализировать неправильно построенные предложения. Это привело к созданию новых методов синтаксического анализа, основанных на вероятностном подходе, к которым нужно отнести вероятностные грамматики и вероятностный разбор.

Вероятностные грамматики являются расширением формальных и отличаются от них в следующем: каждому правилу построения указана некоторая вероятность применения этого правила при построении предложения. После того, как произведен синтаксический анализ предложения известно, на основе каких правил было построено это предложение и на основе сопоставленных с ними вероятностей может быть посчитана "суммарная" вероятность. Очевидно, что одно и то же предложение может быть разобрано несколькими способами. Для каждого из них считается его "суммарная" вероятность и выбирается наиболее вероятный способ построения предложения. Этот метод способен анализировать неправильно построенные предложения, но он, как и два предыдущих способа анализа, включает в себя систему заранее задаваемых правил.

Синтаксический анализ на основе обучающихся систем - это пока еще малоизученный подход. Он заключается в следующем: разрабатывается некоторое множество примеров, каждый из которых содержит пару: исходное предложение и результат его синтаксического анализа. Этот результат вводится человеком, занимающимся обучением системы, в ответ на каждое подаваемое на вход предложение. Затем, при подаче на вход предложения, не входящего в список примеров, система сама генерирует результат. В качестве подхода к реализации такой обучающейся системы используются ряд способов: нейронные сети, деревья вывода, ILP и методы поиска ближайшего соседа.

Это далеко не весь спектр методов синтаксического анализа. Удовлетворительных решений данного вопроса пока еще не найдено, хотя есть методы, дающие неплохие результаты, но работающие только на подмножестве языка. Решение задачи синтаксического анализа текста должно послужить основой, во-первых, для построения более совершенных синтаксических корректировщиков (программных средств, проверяющих

правильность построения предложения) и, во-вторых, для построения алгоритмов более качественного семантического анализа текстов.

4. Семантический анализ.

Семантический анализ текста базируется на результатах работы синтаксического анализа, получая на входе уже не набор слов, разбитых на предложения, а набор деревьев, отражающих синтаксическую структуру каждого предложения. В силу того, что методы синтаксического анализа пока мало изучены, решения целого ряда задач семантической обработки текста базируются на результатах анализа отдельных слов, и вместо синтаксической структуры предложения, анализируются наборы стоящих рядом слов.

Большинство методов семантического анализа должны, так или иначе, работать со смыслом слов, а, следовательно, должна быть какая-то общая для всех методов анализа база, позволяющая выявлять семантические отношения между словами. Такой основой является тезаурус языка. На математическом уровне он представляет собой ориентированный граф, узлами которого являются слова (в их основной словоформе). Дуги задают отношения между словами и могут иметь следующие окраски.

1. Синонимия. Слова, связанные дугой с такой окраской являются синонимами.
2. Антонимия. Слова, связанные дугой с такой окраской являются антонимами.
3. Гипонимия. Дуги с такой окраской отражают ситуацию, когда одно слово является частным случаем другого (Например, слова "мебель" и "стол"). Дуги направлены от общего слова к более частному.
4. Гиперонимия. Это отношение является обратным к гипонимии.
5. Экванимия. Дугами с такой окраской связаны слова, являющиеся гипонимами одного и того же слова.
6. Амонимия. Слова, связанные таким отношением, имеют одинаковое написание и произношение, но имеют различный смысл.

7. Паронимия. Данный тип дуги связывает слова, которые часто путают.

8. Конверсивы. Слова, связанные такой окраской, имеют "обратный смысл" (Например, "купил" и "продал").

Таким образом, тезаурус задает набор бинарных отношений на множестве слов некоторого естественного языка. В настоящий момент имеется тезаурус английского языка , но для русского языка работа по созданию тезауруса еще не проделана, хотя имеются коммерческие продукты, включающие в себя тезаурус подмножества русского языка , а также отдельные словари синонимов и антонимов для подмножества русского языка. К сожалению, эти словари в электронном виде пока не доступны.

Семантический анализ текста включает в себя ряд практически важных задач и, поэтому, будем рассматривать не методы анализа, а актуальные задачи и уже существующие их решения. Одной из наиболее изученных задач является контекстно-свободный поиск текстовой информации. Ее смысл заключается в следующем: имеется большой набор файлов, содержащих тексты на некотором естественном языке, и задана некоторая строка. Необходимо найти все файлы, в которых она или похожая текстовая информация встречается. В подавляющем большинстве случаев необходим именно "неточный" поиск, то есть по смыслу слова с учетом специфики естественного языка. Большинство существующих систем основываются исключительно на морфологическом анализе слов и не задействуют более сложных схем анализа.

Наиболее важной с практической точки зрения задачей является извлечение информации из текстовых данных и представление ее в виде формальной системы знаний, в частности, в виде семантической сети. На данный момент имеются экспериментальные разработки в данном направлении, ориентированные на конкретную предметную область знаний,

однако готовых коммерческих программных продуктов пока нет. Рассмотрим эту задачу более подробно.

Имеется семантическая сеть, состоящая из узлов и связей. Каждому из узлов соответствуют некоторые атомарные данные, а каждой дуге - некоторая окраска. Если семантическая сеть построена на основе анализа некоторого текста и является формализованным представлением содержащихся в нем знаний, то каждому ее элементу соответствует некоторый фрагмент исходного текста. Узлам, отражающим одинаковые по смыслу данные (например, вес, возраст, дата рождения), соответствуют аналогичные по синтаксическому строению фрагменты исходного текста. Значит, можно ввести некоторые шаблоны. Каждый из них описывает синтаксическую структуру части исходного текста и создаваемые элементы семантической сети. При описании синтаксической структуры указываются не только связи слов в предложении, но и условия, накладываемые на каждое из слов. Эти условия могут проверять как морфологические или семантические (на основе тезауруса) характеристики слова, так и смысловые пометки этого слова, поставленные при поиске других шаблонов. Если какая-то часть текста удовлетворяет всем условиям, указанным в шаблоне, то происходит ее формализация.

Извлечение информации из текстовых данных является основой для решения двух важных задач: во-первых, "раскопки" текста, во-вторых, создания систем загрузки текстов в хранилища данных. Такие системы существуют и предназначены для интеграции и очистки данных, помещаемых в хранилища, но они не предоставляют никаких средств ввода данных, содержащихся в текстовом виде.

Наряду с извлечением информации существует и обратная задача генерации правильно построенных текстов. Исходными данными для таких систем является четко формализованные знания. На первый взгляд эта задача может показаться странной, поскольку в большинстве случаев формализованные знания можно представлять в виде бланков, имеющих

четкую, заранее определенную систему полей. Но это не всегда так: в случае, когда предметная область имеет сложную и разветвленную структуру, то часто оказывается, что большинство полей бланка оказываются пустыми, что сильно затрудняет восприятие информации, и для конченого пользователя было бы гораздо проще и удобнее иметь дело не с таким бланками, а с неформализованным (но корректно построенным) текстовым описанием тех же самых данных.

С позиции двух указанных задач интересно рассмотреть методы обработки текстовой информации, разработанные Роджером Шенком . Они образуют психолингвистический подход к анализу текстовой информации и основываются на двух идеях. Во-первых, для анализа одного предложения не обязательно рассматривать все его слова- смысл предложения можно определить по "ключевым" словам и наличию связей между ними. Вторая идея заключается в представлении результатов анализа текста в виде концептуальной сети, способной формально описать смысл, содержащийся в исходном тексте и являющейся семантической сетью с предопределенным набором типов узлов и дуг.

Наконец, осталось рассмотреть две достаточно известные задачи обработки текстовой информации: автоматическое реферирование и автоматический машинный перевод. Основные требования, предъявляемые к реферату: он должен отражать основные идеи и моменты текста, а также, реферат должен являться корректно построенным текстом. Известны два основных направления в решении этой проблемы: удаление из исходного текста всех "ненужных" предложений и самостоятельное построение реферата исходного текста. Основные сложности, связанные с первым подходом, заключаются в определении ключевых предложений текста, и затем связи этих предложений в единый, удобочитаемый текст. Второй подход включает в себя три этапа: анализ текста и построение его формального описания, выбор из этого описания ключевых моментов, формирование реферата. На сегодняшний день имеются как научные, так и

коммерческие разработки систем реферирования, способные обрабатывать русскоязычные тексты.

Автоматический машинный перевод - это одна из старейших задач искусственного интеллекта и на текущий момент представлено множество коммерческих систем, способных переводить несложные тексты.

Тема 5. Информационные технологии в обработке текстов.

1. Автоматическое чтение текста.
2. Автоматическое реферирование и аннотирование текста.
3. Формулировка задачи автоматического реферирования и аннотирования текста.
4. Системы автоматического реферирования и аннотирования текстов.

1. Автоматическое чтение текста.

В наши дни для быстрого и качественного ввода текстовой информации в компьютер широко используются сканеры. Сканер работает по принципу фотоаппарата, позволяя ПК «увидеть» текст. Для того, чтобы «понять» его содержание, т.е. перевести графическое (точечное) изображение символов в пригодную для дальнейшей обработки (редактирования, реферирования, перевода и т.д.) текстовую форму, необходима система автоматического чтения текста или оптического распознавания символов (OCR-система — Optical Character Recognition).

В классическом понимании ***система автоматического чтения текста*** — это компьютерная программа, позволяющая преобразовать текст с бумажного носителя в электронный текстовый файл, который может быть прочитан средствами обработки текстов. Исходный текст должен быть вначале введен в ПК с помощью сканера или получен на факс-модем.

История появления современных программ в области распознавания начинается с конца 40-х годов XX века, когда ученые многих стран стали работать над идеей обучения компьютера умению решать разные интеллектуальные задачи. Автоматическое чтение текста, распознавание речи, решение шахматных задач и головоломок и даже сочинение музыки и стихотворений — вот далеко не полный перечень идей, которые выдвигались и разрабатывались в то время. К концу 50-х годов эти идеи оформились в отдельную область знания — ***искусственный интеллект.*** Одной из задач, которая вскоре выделилась в отдельное направление, была задача распознавания образов. Идеальная компьютерная система распознавания должна уметь формировать, анализировать и интерпретировать любое изображение, в том числе и символьное.

В настоящее время во всем мире широко известны две OCR-системы, созданные российскими разработчиками. Это FineReader компании «ABBYY Software House» и CuneiForm фирмы «Congitive Technologies». Несмотря на определенные трудности распознавания текстов (на одной странице может встретиться до трех и более шрифтов разного размера, стиля, начертания и гарнитуры; тексты часто бывают многоязычными, многоколоночными, включают в себя таблицы, графические изображения и т.д.), возможности систем автоматического чтения текста огромны. Рассмотрим основные из них.

1. В процессе сканирования и распознавания текста документа OCR-системы автоматически подбирают яркость сканирования, фрагментируют каждую страницу, выделяя в ней области графических иллюстраций и таблиц, распознают символы текста, проверяют орфографию распознанных слов и показывают окончательный результат в текстовом редакторе.

2. OCR-системы позволяют распознавать печатные символы почти двух сотен языков.

3. Хорошо распознаются рукопечатные символы, т. е. символы,

написанные от руки печатными буквами с небольшим интервалом между ними.

4.OCR-системы узнают все используемые в тексте документа шрифты без предварительного обучения, хорошо воспринимают полужирный, курсивный, слипшийся, подчеркнутый и многоколоночный текст. Изначально в мире преобладали системы автоматического чтения текста, требующие обучения каждому новому шрифту (новой гарнитуре, стилю, размеру и т.д.). Такие системы называли ***мультифонтовыми*** (от англ./оя/ — 'шрифт'). Противоположным классом ORC-систем являются так называемые интеллектуальные программы, именуемые еще ***омнифонтовыми.*** Их не нужно обучать, эти программы распознают разные стилевые начертания одной и той же буквы не потому, что их обучили различным гарнитурам шрифтов, а потому, что они знают топологию (правила начертания) этой буквы.

5.Системы способны самообучаться и распознавать плохо пропечатанные символы или символы незнакомых программе языков.

6.Наряду со сплошными текстами (без таблиц и иллюстраций) программы автоматического чтения текста хорошо распознают:

а) тексты с графикой, подписями, логотипами;

б) таблицы;

в) тексты, напечатанные на цветном (гербовом) фоне;

г) тексты разноформатных документов (например, чертежей).

7.OCR-системы поддерживают все модели сканеров и любые графические форматы.

8.Появились и широко используются сетевые версии программа автоматического чтения текста.

9.Программы автоматического чтения текста поддерживают публикацию бумажных документов в глобальной сети Интернет.В процессе распознавания и генерации HTML-страницы[1] ее оформление

производится по всем правилам Web-публикации.

10. Точность распознавания OCR-систем на текстах хорошего и среднего качества достигает 97—99 %.

Развитие программ автоматического чтения текстов в ближайшем будущем пойдет в направлении повышения точности распознавания текстов низкого качества, выделения текстовой информации на фоне шумов (например, распознавание номерных знаков автомобилей), а также интеграции OCR-систем с различными программами обработки информации (системами машинного перевода, системами автоматического аннотирования и реферирования текстов, электронными архивами, системами автоматизации делопроизводства и т.д.).

2. Автоматическое реферирование и аннотирование текста.

Рефератом будем называть связный текст, который кратко выражает не только центральную тему или предмет какого-либо документа, но и цель, применяемые методы, основные результаты описанного исследования или разработки. Рефераты обычно составляют к научно-техническим документам — научным книгам, статьям, патентам на изобретение и т.п. Поэтому в приведенном определении и говорится о «методах и основных результатах описанного исследования или разработки». Реферат акцентирует внимание читателя на новых сведениях и определяет целесообразность его обращения к исходному документу. Он помогает человеку ориентироваться в информационных потоках, оперативно отбирать для себя наиболее ценную и полезную информацию. Процесс составления реферата называется реферированием.

Аннотацией называют краткое изложение содержания документа, дающее общее представление о его теме. Таким образом, если реферат в краткой форме знакомит читателя с сутью излагаемого в документе содержания (фактами, методикой, экспериментами и т.п.), то аннотация выполняет лишь сигнальную функцию, сообщая о том, что опубликована

статья или книга на определенную тему. Процесс составления аннотации называется аннотированием.

Рефераты и аннотации представляют собой вторичные документы. Первичные, или исходные, документы — это, как мы уже отмечали, книги, статьи, патенты и т.п. В каждом вторичном документе можно выделить два компонента информации:

Первый компонент содержит информацию первоисточника (о чем книга, статья). Второй компонент — это сведения о самом первичном документе (тип документа: книга, статья и т.п.; вид: печатный, рукописный; год издания; место издания и т.д.). В дальнейшем речь пойдет только о первом компоненте вторичного документа.

Научно-технический прогресс привел к появлению большого числа публикаций (книг, статей и т.п.) по самым разным проблемам науки, техники, образования, и специалисты не успевают следить за новейшей литературой по своей области знания. Для этого, как установлено, человек должен был бы прочитывать ежедневно 1500 страниц текста на разных языках, что явно превышает его физические возможности. Поэтому для оперативного «поверхностного» знакомства с новейшими публикациями используются рефераты и аннотации книг и статей, которые составляются в специальных организациях и публикуются в реферативных журналах (РЖ) и реферативных сборниках (РС).

Реферирование и аннотирование текста являются довольно сложными и трудными видами интеллектуальной деятельности. Составление человеком рефератов или аннотаций занимает много времени. Это приводит к тому, что до ученых, педагогов, инженеров и других специалистов новейшая информация (особенно зарубежная) доходит очень медленно, что, в свою очередь, ведет к повторению в разных странах и в пределах одной страны одних и тех же исследований, более позднему применению новейших методик, технологий, процессов. Чтобы как-то избежать этого, для составления рефератов и аннотаций применяют

современные компьютеры. Составление реферата или аннотации текста с помощью компьютера называется автоматическим реферированием или аннотированием.

Формулировка задачи автоматического реферирования и аннотирования текста

Для того чтобы дать точную формулировку задачи ***автоматического реферирования*** и ***аннотирования*** текста, необходимо проанализировать, как выполняет работу по составлению реферата га аннотации человек (референт). Анализ литературы, содержащей принципы построения человеком рефератов или аннотаций, показывает, что при этом выделяют три этапа:

1) подготовительный — референт определяет тематическую направленность текста и пытается понять и осмыслить документ в целом;

2) аналитический — референт делит текст на некоторые фрагменты (абзацы, аспекты и т.п.). Каждый фрагмент внимательно изучается, в нем выделяют основные смысловые единицы (предложения, словосочетания, слова). Данный этап заканчивается составлением плана будущих реферата или аннотации;

3) этап непосредственного построения реферата или аннотации - выделенные ранее смысловые единицы (их комбинации или преобразования) располагаются в единый вторичный текст в соответствии с планом реферата или аннотации.

Рассмотрим несколько подробнее отдельные моменты 2-го и 3-го этапов. В качестве основных смысловых единиц, выделяемых из исходного текста на 2-м этапе, могут выступать: 1) целые ключевые предложения; 2) ключевые словосочетания и слова.

Ключевое (опорное) слово — это термин, относящийся к основному содержанию текста и повторяющийся в нем несколько раз (с учетом всех возможных синонимов).

Ключевое словосочетание — это сочетание слов, среди которых есть одно или несколько ключевых.

Ключевым предложением считается предложение, содержащее два и более ключевых слова или ключевых словосочетания.

Составление плана будущих реферата или аннотации заключается в выделении некоторых смысловых ориентиров, которые на 3-м этапе будут развернуты более подробно. В качестве таких ориентиров выступают:

1) основные темы и подтемы исходного текста;

2) основные аспекты исследования;

3) основные ключевые предложения, словосочетания и слова.

Выбор тех или иных ориентиров зависит от типа составляемого реферата или аннотации.

Создаваемый на 3-м этапе реферат или аннотация содержат выделенные ранее смысловые единицы. В качестве смысловых единиц реферата могут выступать:

1) полные (без изменения) ключевые предложения исходного текста;

2) перефразированные ключевые предложения исходного текста;

3) предложения, составленные из ключевых слов или словосочетаний исходного текста с помощью специальных связующих элементов;

4) предложения, обобщающие несколько предложений исходного текста (не обязательно ключевых).

При перефразировании применяются различные лексико-грамматические явления: использование синонимов, конверсивов, замен по принципу «вид — род», «часть — целое» и т.п.

При получении новых предложений из ключевых слов и словосочетаний исходного текста чаще всего используют различные логико-

смысловые скрепы, например, *потому что, в то время как, поэтому, вследствие* и т.п.

В обобщающих предложениях исходный текст передается совершенно другими словами. В них то же самое содержание излагается в более кратком виде.

Смысловыми единицами аннотации могут быть:

1) ключевые слова или словосочетания исходного текста с предшествую-щими им специальными фразами — реляторами типа: «В статье рассматриваются следующие вопросы: ...», «Книга посвящена следующим проблемам: ...» и т.п.;

2) специальные предложения, содержащие оценочные элементы: «Рассматривается важная проблема...», «Статья посвящена актуальной теме...» и т.д.;

3) специальные предложения, содержащие клише, т. е. специализированные словесные штампы, фиксирующие внимание читателя на определенных аспектах содержания: *«Недостаток...* заключается», *«Цель* публикации...», «Ставится *задача...»,* «Делается *попытка...»* и т.д.

Следующий важный вопрос, который необходимо рассмотреть, связан с тем, как человек выбирает из текста ключевые предложения, словосочетания и слова. Это делается, как уже отмечалось, на 2-м этапе общего процесса составления вторичного документа. Читая текст повторно (первый раз он читается на подготовительном этапе) или в третий раз, человек мысленно выделяет в нем три типа единиц (предложений, словосочетаний, слов):

1) единицы, которые обязательно должны быть включены в реферат или аннотацию. Такие единицы отражают новые идеи, гипотезы, новые методы, явления, процессы, новые результаты, т. е. все новое и оригинальное, что есть в исходном документе. Это, по существу, и есть основные смысловые единицы текста (ключевые предложения,

словосочетания и слова);

2)единицы, которые отражают фактические данные: параметры изделий, процессов, методов и т.д. Такие единицы не являются принципиально новыми;

3)единицы, которые аргументируют и иллюстрируют единицы первых двух типов.

Единицы первого уровня обязательно используются при составлении реферата. Из единиц второго уровня используются лишь некоторые (в зависимости от типа реферата или его потребителя). Третья группа единиц изредка переносится в реферат в обобщенном виде.

Если поручить составление реферата или аннотации компьютеру, то, очевидно, его надо научить выполнять те же действия, которые осуществляет человек. Компьютер должен уметь:

1) находить в тексте ключевые слова, словосочетания и предложения;

2) находить в тексте менее значимые единицы;

3) составлять из текстовых единиц двух первых типов смысловые единицы реферата или аннотации;

4) составлять из таких единиц текст реферата или аннотации. Говоря о двух последних «умениях» компьютера, необходимо помнить, что почти во всех существующих системах автоматического реферирования в качестве основных смысловых единиц реферата выступают ключевые предложения или ключевые словосочетания и слова исходного текста. Первые в их последовательной совокупности (в том порядке, в котором они идут в исходном тексте) образуют текст (квазитекст) реферата. Второй тип смысловых единиц (ключевые словосочетания и слова) используется компьютером для построения так называемых ***табличных рефератов***[1]**.**

При составлении с помощью компьютера аннотации также используются как ключевые предложения (в том виде, что и при составлении реферата), так и ключевые слова и словосочетания.

Последние перечисляются вслед за реляторами вида: «В статье рассматриваются следующие вопросы:...», «Книга посвящена следующим проблемам: ...», «Статья раскрывает следующие понятия: ...» и т.д.

По способам выделения из исходных текстов ключевых словосочетаний и предложений (первые два «умения» компьютера) различают несколько методов автоматического реферирования и аннотирования текстов. Наиболее известны следующие три группы методов:

1) статистические;
2) позиционные;
3) логико-семантические.

Суть статистической группы методов заключается в том, что:

1) ключевыми словами считаются такие знаменательные слова текста, которые с учетом всех синонимов встречаются в тексте наибольшее число раз;

2) ключевым предложением считается предложение текста, которое:

а) имеет несколько ключевых слов;

б) содержит ключевые слова на небольшом расстоянии друг от друга.

Принадлежность слова, словосочетания или предложения к числу ключевых определяется специальными статистическими коэффициентами.

В *позиционных методах* автоматического реферирования и аннотирования ключевым предложением считается предложение, входящее в заголовок, подзаголовок, начало или конец какой-то части текста или всего текста. Такие предложения, как правило, содержат информацию о целях, методах, выводах и результатах исследования, описанного в первичном документе. Важность тех или иных предложений с указанной точки зрения определяется экспертами путем изучения семантической структуры первичных документов определенного типа.

Логико-семантические методы опираются на исследование структуры и семантики текстов. Существует несколько вариантов этих

методов, но цель их одна — выделить из конкретного текста предложения с наибольшим функциональным весом. Величина эта зависит от многих факторов: наличия в исследуемом предложении специальных семантически значимых слов, связи этого предложения с другими предложениями текста, синтаксического типа самого предложения и т.д.

Формулируя задачу построения системы автоматического аннотирования и реферирования текста, необходимо четко указать:

1) метод, который используется для выделения ключевых слов предложения;

2) способ определения ключевых словосочетаний предложения;

3) критерий выделения ключевых предложений текста;

4) тип подготавливаемой аннотации: текстовая, в виде релятора с последующими ключевыми словами и словосочетаниями, или табличная;

5) тип формируемого реферата: текстовый или табличный. Учитывая все сказанное, сформулируем задачу ***автоматического реферирования и аннотирования текста*** следующим образом:

«На устройстве внешней памяти (например, дискете или винчестере) находится английский научно-технический текст. Начало каждого абзаца в нем обозначено знаком *. Используя для выделения ключевых (опорных) слов текста один из вариантов статистического метода, а именно коэффициент важности слова:

$$Кважс = \frac{F * m}{N * n}$$, где

F-частота словоупотреблений в тексте;

m-число абзацев текста, в которых встретилось слово;

N-общее число словоупотреблений в тексте;

n-общее число абзацев в тексте.

составить алгоритм, позволяющий получить:

а) аннотацию текста в виде релятора со следующими за ним ключевыми словосочетаниями текста. Ключевым сочетанием считается

ключевое имя существительное со стоящим перед ним определением, выраженным именем прилагательным или причастием, не относящимся к числу общеупотребительных;

б) словесный реферат текста в виде последовательной цепочки ключевых предложений. Ключевым предложением текста будем считать предложение, содержащее три и более разных ключевых слова.

Аннотацию и реферат вывести на экран дисплея».

2. Системы автоматического реферирования и аннотирования текстов.

Наиболее известной в наши дни промышленной системой автоматического реферирования является система «Либретто», разработанная по технологии компании «МедиаЛингва». Она осуществляет автоматическое реферирование русских и английских текстов любого объема и любой степени сложности. Исходный текст может сжиматься с необходимым пользователю коэффициентом сжатия, а реферат выдается в виде цепочки ключевых предложений или ключевых слов (аннотация). На западном рынке к системам подобного типа относятся системы автоматического реферирования «Inxight Summerizer», Prosum. Несколько упрощенный вариант реферата в виде последовательности именных групп, выделенных с помощью синтаксических анализаторов, могут выдать системы «Extractor» и «TextAnalyst». Последняя система создана в Москве в инновационном центре «Микросистемы».

Тема 6. Информационные технологии в обучении языкам.

1. Общие принципы компьютерного обучения языкам.
2. Теоретические обоснования выбранного метода обучения.

3. Создание технологии компьютерного обучения языкам.

4. Использование персональных компьютеров в обучении иностранным языкам.

5. Способы использования компьютеров для обучения языкам.

6. Компьютерные программы индивидуализированного обучения языкам.

1. Общие принципы компьютерного обучения языкам.

Широкое использование компьютеров в различных аспекта деятельности человека не обошло стороной проблему обучения человека языкам. Уже сейчас создано большое число компьютерных программ обучения иностранным языкам. Эти проблемы рассматриваются в журналах, выпускаемых по всему миру. Ежегодно проблемам обучения языкам посвящается большое число конференций.

Сложность задачи обучения языкам объясняется тем, что любое обучение — задача комплексная, требующая учета данных психологии, педагогики, методики, особых свойств изучаемого предмета. По существу, каждая обучающая программа — это достаточно сложная система искусственного интеллекта.

За последние годы значительно изменился принцип применения компьютерных программ для обучения иностранному языку. Если ранее утверждалось, что наиболее эффективно они могут быть использованы в рамках автоматизированных обучающих систем, то с появлением компьютеров в доме обучаемых такие программы чаще используются индивидуально и подбираются в зависимости от цели обучения. Однако принципы построения таких обучающих программ в целом остались неизменными. Они реализуются путем выполнения следующих основных задач:

1) теоретического обоснования выбираемого метода обучения;

2)создания с опорой на выбранный метод технологии обучения;

3)построения обучающей программы, реализующей выбранную технологию обучения.

Рассмотрим процесс решения этих задач более подробно.

2. Теоретические обоснования выбранного метода обучения.

С точки зрения принципов восприятия информации при обучении с помощью компьютера выделяют, как правило, два теоретических подхода: бихевиористский и когнитивно-интеллектуальный .

Бихевиористский подход, связанный с постулатом «чем чаще употреблено слово, тем лучше оно запоминается», в последние годы усовершенствовался использованием целого ряда приемов: дедуктивного контроля ответов, созданием универсальных банков Данных об изучаемом объекте или явлении, построением справочной информации в виде гипертекстов и т.д. В рамках такой теории различают следующие методы автоматизированного обучения:

1) программирование учебной деятельности обучаемого;

2) тестирование;

3) информирование.

Первый из этих методов обучения характерен тем, что управляющие воздействия на обучаемого полностью определяются обучающей программой. В такой программе каждому обучаемому в зависимости от его уровня знаний полностью задается последовательность учебных или контрольных заданий.

При тестировании компьютер по специальным программам выявляет индивидуальные профессиональные и психологические характеристики обучаемых и достигнутые ими уровни знаний. При этом обучаемый лишь отвечает на вопросы, но оценку за знания не Получает. Этот метод достаточно часто используется при оценке различных аспектов знания иностранных языков (тестирование словарного запаса, способности к изучению иностранных языков и т.п.).

Суть метода информирования заключается в том, что в память компьютера помещаются некоторые справочно-информационные данные (грамматический справочник, орфографический словарь, Двуязычный словарь и т.п.), которые обучаемый может использовать при подготовке к занятиям или непосредственно в процессе занятий. Тем не менее бихевиористский подход не может преодолеть механистичность обучения и отсутствие развития когнитивных способностей обучаемых.

При ***когнитивно-интеллектуальном подходе*** у обучаемого активизируются познавательные функции. Для успешной реализации такого подхода в памяти компьютера создается универсальная учебная среда, включающая различные грамматические справочники, словари, спеллеры, другие вспомогательные материалы. При таком подходе в принципе возможны следующие методы автоматизированного обучения:

1) моделирование учебной среды;

2) свободное обучение.

Суть процесса моделирования учебной среды сводится к созданию компьютерных программ, которые моделируют структуру некоторого объекта или принципы его действия. При этом, как и при бихевиористском подходе, управляющие воздействия также полностью определяются обучающей программой. Обучаемый может выбирать учебные задания из некоторого конечного множества заданий, включенного в универсальную учебную среду. Различают два типа моделей учебной среды:

1) модели объективного (предметного) типа;

2) модели мыслительного типа.

Модели первого типа служат для познания некоторого объекта или приобретения навыков работы с таким объектом. Примером такой модели является компьютерная программа, обучающая студента работе на клавиатуре.

Модели мыслительного типа — это уже упоминавшиеся выше воспроизводящие инженерно-лингвистические модели. Их задача —

изучение некоторых процессов, явлений или же динамического поведения некоторого объекта. Чаще всего они используются в процессе преподавания радиотехники, электроники, физики, биологии и других дисциплин. Широко используются подобные модели и в лингвистике.

Метод свободного обучения характерен тем, что он дает возможность самому выбрать тематику обучения и способ работы с компьютером. Компьютер при этом не только предъявляет обучающие кадры по указанию обучаемого, но и уточняет его действия, предлагая обучаемому наиболее эффективный способ использования учебного материала. При этом выделяют три вида обучения:

1) структурно-управляемое обучение;
2) обучение принятию решения;
3) генеративное обучение.

Учебный материал, используемый при структурно-управляемом обучении, представляется в виде некоторой иерархической структуры данных. Для каждого структурного уровня в компьютере заранее определены цели обучения и условия для успешного достижения такой цели. Элементарным примером такого вида обучения может служить задача обучения формообразованию русских глаголов. Здесь можно выделить четыре структурных уровня, вводящих по принципу «от простого к сложному» основные понятия, связанные с формообразованием русского глагола.

При структурно-управляемом свободном обучении обучаемый сам может выбирать начальный уровень обучения (в нашем примере это может быть и 2-й, и 3-й структурные уровни). Однако локальная цель высшего уровня должна совпадать с конечной целью обучения по данной программе.

Таблица 6. Структурные уровни и локальные цели обучения.

Номер структурного уровня	Локальная цель обучения	Условия для достижения локальной

1 -й структурный уровень	Усвоение понятий о: 1) категории числа глагола; 2) категории лица глагола; 3) категории вида	Правильное выполнение конкретного числа заданий на определение по форме глагола его числа, лица, вида
2-й структурный уровень	Усвоение понятий о: 1) вспомогательных глаголах; 2) модальных глаголах	Правильное выполнение определенных заданий на опознавание в рамках предложения вспомогательных и мо-
3-й структурный уровень	Усвоение понятий о категории времени глагола	Правильное выполнение определенного числа заданий на определение по форме
4-й структурный уровень	Усвоение понятий о правилах формообразования русского глагола	Правильное выполнение определенного числа заданий на оформление основы глагола его формообразующи-

При обучении принятию решения учебная информация, заложенная в компьютер, представляется в виде некоторого набора специальным образом записанных ситуаций (например, в виде фреймов*).* При таком виде свободного обучения обучаемый, исходя из задаваемой компьютером исходной ситуации и ориентируясь на его ответы, должен прийти к некоторой конечной ситуации. Наиболее ярко это проявляется, например, при построении компьютерной обучающей системы, моделирующей действия студента-медика при постановке диагноза болезни. Компьютер при этом выступает в качестве больного, которому студент задает определенные вопросы.

При обучении иностранному языку этот метод также может широко использоваться. Например, можно дать задание компьютеру выступить в роли продавца в магазине, где обучаемый, ведя диалог с продавцом, хочет купить некоторую вещь. При этом обучающая программа может корректировать действия обучаемого или после каждого его шага принятия

решения, либо она может комментировать его действия после окончательного завершения всей программы.

Генеративное обучение как вид свободного обучения пока возможно лишь теоретически. Здесь предполагается активное взаимодействие двух моделей: постоянно изменяющейся модели обучаемого (как следствие последовательного усвоения в процессе обучения некоторых знаний) и модели учебного материала. Учебный материал в данном случае представляет собой совокупность некоторых понятий и связей между ними, которые должен усвоить обучаемый. Обучающие воздействия компьютера при таком обучении не заданы заранее, а вырабатываются обучающей программой в зависимости от степени усвоения исходного материала обучаемым. Таким образом, решая задачу теоретического обоснования выбираемого метода обучения, следует выбирать одну из описанных выше моделей обучения.

2. Создание технологии компьютерного обучения языкам.

Допустим, что для создания обучающей программы выбран один из описанных выше методов. Каждая компьютерная программа, обучающая какому-либо аспекту языка, представляет собой некоторую совокупность более простых программ, связанных с обучением простым лингвистическим понятиям или явлениям. Обучающую программу в ее полном виде назовем ***автоматизированным учебным курсом*** (АУК) и определим ее следующим образом: АУК — это комплекс компьютерных программ, направленных на достижение одной или нескольких учебных целей (формирование, систематизация, закрепление, применение или контроль знаний и умений). Нет единой технологии создания АУК. Разработчики выделяют при этом от 4 до 14 конкретных задач. Рассмотрим детально те из них, которые связаны непосредственно с разработкой технологии некоторого курса компьютерного обучения.

Наиболее приемлемым представляется выделение в процедуре разработки компьютерной технологии обучения некоторому курсу четырех следующих основных задач:

1) проектирование состава курса и его содержания;

2) методическая проработка учебного материала каждой отдельной задачи, входящей в состав курса, и создание ее обучающего сценария;

3) создание обучающей программы по данной задаче и ее экспериментальная проверка (тестирование);

4) объединение обучающих программ всех задач курса в единой АУК.

3. Способы использования компьютеров для обучения языкам.

В наши дни уже не вызывает сомнения необходимость использования компьютеров в обучении иностранным языкам. С информационной точки зрения искусство обучения может рассматриваться как наиболее эффективная организация общения обучаемых со знанием, а основная задача современного образования- научить эти знания добывать, классифицировать и использовать в ей повседневной жизни.

Если рассматривать современные персональные компьютеры просто как средство технической поддержки учебного проса, а как устройство, способное выполнять педагогические функции, несущее в себе конкретные знания и передающее эти знания в процессе диалога с обучаемым, то можно выделить три способа использования компьютеров в обучении.

1.Компьютер — помощник преподавателя. В этом случае процесс обучения строится в соответствии с традиционным содержанием образования и методами передачи знаний от преподавателя к обучаемым. Используемые при этом обучающие программы лишь моделируют некоторые задачи, темы, разделы изучаемого курса и общаются с обучаемым по достаточно жесткому сценарию. Здесь преобладает групповой метод

обучения в традиционных группах, классах и т.п.

2.Компьютер — преподаватель. При таком способе также моделируется традиционная методика обучения и строится жесткий сценарий обучения. Однако соответствующие обучающие программы направлены на обучение целому курсу (информатике, английскому языку и т.д.). Как правило, они предназначены для индивидуализированного обучения (чаще всего в домашних условиях).

3.Компьютер — источник знаний и «оцениватель» знаний обучаемого. Здесь используется так называемая альтернативная педагогика, когда обучаемый, исходя из целей обучения и своих возможностей, опираясь на собственный опыт и знания, обращается к компьютеру как к носителю необходимых для него знаний или «оценивателю» полученных обучаемым знаний. Такой подход возможен как при групповом, так и индивидуализированном обучении в рамках дистанционного обучения.

Методика создания компьютерных обучающих программ, применяемых в качестве помощника преподавателя, подробно рассмотрена выше. Издано много литературы, посвященной проблемам использования таких программ в аудиториях, для самостоятельной работы в классах, их эффективности, преимуществ таких программ по сравнению с традиционным обучением и их недостатков.

Особенности организации обучающих программ, используемых двумя другими способами, требует более детального рассмотрения.

Компьютерные программы индивидуализированного обучения языкам

Компьютерные программы индивидуализированного обучения языкам обычно представляют собой некоторые законченные курсы, предназначенные для начального обучения, совершенствования языка и т.д. От программ, предназначенных для поддержки обучения, они

отличаются тем, что полностью должны заменить преподавателя в процессе обучения языку. Данный факт заставляет создателей таких программ формализовать иррациональную интуитивно-творческую деятельность преподавателя, приблизить процесс обучения с помощью подобной обучающей программы к реальному процессу обучения с преподавателем. Среди большого числа требований, которым должны удовлетворять компьютерные программы индивидуализированного обучения, наиболее важными с педагогической точки зрения являются следующие . Подобные программы должны:

1) совмещать в себе обучающую, контрольную и поисковую функции;

2) опираться на сценарии, приближенные к обычному традиционному обучению;

3) максимально использовать принцип наглядности и доступности, т. е. выводить на экран компьютера не только текст, но и звук, иллюстрации, видео и т.п.;

4) иметь средства быстрой и объективной оценки знаний обучаемых даже в тех случаях, когда ответ обучаемого далек от наиболее ожидаемого;

5) содержать возможность настройки на конкретного обучаемого (выбор способа подачи нового материала, типа упражнений, скорости ответа и т.п.).

Как правило, обучающие программы, используемые для индивидуализированного обучения, реализуются в виде так называемых ***мультимедийных обучающих программ.*** Слово *мультимедийный* появилось вне связи с компьютерами в англо-русском словаре 1969 года издания. В то время урок, проводимый преподавателем, назывался мультимедийным, если в нем присутствовали и рассказ учителя, и магнитофонная запись, и кино, и слайды, и любые средства технического обучения. Сегодня под мультимедийной обучающей программой понимается компьютерная программа, использующая текст, звук, цвет, графику и движение. В

понятие *звук* входят речь, музыка, их комбинации (музыка — речь — пение и др.), а также различные звуковые эффекты. Графика в таких программах может быть представлена различными рисунками, геометрическими фигурами (круг, ромб и т.д.), символами, фотографиями и сканированными изображениями. Движение в мультимедийных программах представляется в виде последовательности статических элементов (кадров) и может быть трех видов: видео *(life video),* квазивидео и анимация. Видео — это последовательность черно-белых или цветных фотографий, пропускаемая на экране компьютера со скоростью около 24 фотографий в секунду. Квазивидео — тоже последовательность кадров, движущаяся по экрану со скоростью 6—12 фото в секунду. Анимация — это последовательность рисованных изображений. Такие программы дают возможность активизировать различные каналы восприятия информации и повышают степень запоминания и усвоение учебного материала.

В целом процесс создания мультимедийных обучающих программ включает следующие этапы*:*

1)разделение всего курса, который будет предложен для обучения, на определенное число тем и подтем;

2)отбор для каждой темы или подтемы определенного лексического и грамматического материала;

3)создание для каждой темы или подтемы набора сценариев, в рамках которых будут закрепляться лексический материал и грамматические правила. Такие сценарии должны включать звук, графику и движение;

4)подбор в соответствии со сценариями необходимых текстов, аудио- и видеоматериалов; программирование сценариев. Такая работа может выполняться высококвалифицированными программистами на различных языках программирования. Такое программирование можно осуществить и с помощью специальных инструментальных систем для разработки и редактирования учебных программ (например, LINK WAY, TEACHCAD, MULTIMEDIA TOOLBOOK, МАГИСТР, СЦЕНАРИЙ, STRATUM,

AUTHORWARE PROFESSIONAL и др..

В целом задачи создания эффективных мультимедийных программ для изучения иностранных языков могут успешно решаться только крупными коллективами, содержащими в своем составе опытных преподавателей иностранных языков, методистов, психологов, программистов, электронщиков. К числу наиболее известных фирм такого рода относятся американские фирмы: «Seracuse Language System», «Compulink», «Foreign Lauguage Software Company», «Hyper Glot», «Macmillan Heinemann» и т.д., российские фирмы: «Мультимедиа Технологии», «Istrasoft», «New Media Generation», «Новый диск» и т. п.

Мультимедийную обучающую программу по иностранному языку можно охарактеризовать следующими дидактическими параметрами *:*

1) тип пользователя;
2) назначение программы;
3) количество разделов, тем и подтем в программе;
4) язык комментариев и подсказок;
5) количество единиц словаря;
6) способы представления единиц словаря;
7) логический объем курса в часах (продолжительность последовательного прослушивания всего звукового материала программы без повторов);
8) число иллюстраций.

Помимо этого, каждую программу можно охарактеризовать рядом технических параметров

1) объем учебного текста в Кб;
2) объем памяти в Мб, занятый звуковыми фрагментами;
3) объем видеофрагментов в Мб;
4) формат видео;
5) формат аудио;
6) формат иллюстраций.

По типу пользователей различают следующие мультимедийные программы обучения иностранным языкам: 1) для детей; 2) для молодежи и взрослых; 3) для бизнес-применений; 4) специализированные программы.

По назначению такие программы делят на следующие виды: 1) для игр; 2) для начального обучения языку; 3) для совершенствования знаний языка; 4) для сдачи различных экзаменов; 5) для работы с деловыми текстами.

Так, для начального обучения английскому языку молодежи и взрослых можно использовать такие мультимедийные программы, как Bridge to English, Репетитор English, Профессор Хиггинс, Learno speak English, Everyday English in Communication, Talk to Me, Triple Play Plus.

Для совершенствования знания английского языка молодыми людьми и взрослыми полезны такие программы, как: Complete English, English for Communication, English Cold, English Platinum.

Совершенствование знаний английского языка взрослыми в области бизнеса возможно путем использования мультимедийных программ Business English и EBC (English Business Contracts). Их структура и возможности хорошо описаны в работах [3; 171]. Наконец, примером специализированных мультимедийных обучающих программ, ориентированных на молодежь и взрослых и предназначенных для сдачи международного теста на владение английским языком как иностранным (TOEFL), является программа The Heinemann TOEFL.

Для того чтобы представить возможности мультимедийной обучающей программы, рассмотрим детальнее программу English Gold. Уже отмечалось, что она предназначена для совершенствования маний английского языка молодежью и взрослыми. Программа включает пять разделов: «Фонетика», «Грамматика», «Словарь», Диалоги», «Фильм» — и 144 урока. Комментарии и подсказки оформлены на русском языке. Словарь включает 12000 слов. Каждая словарная единица представлена в письменном и звуковом видах, а

также в виде изображения предмета. Логический объем звука программы составляет более 100 часов. Программа содержит 2096 иллюстраций.

Тема 7. Порождение текстов на естественном языке.

План лекции:

1. Исторический обзор проблемы.
2. Характер процесса порождения.
3.Общие подходы к проблеме порождения текстов на ЕЯ.
4.Лексический выбор. Фразовые словари. Обработка грамматики.

1. Исторический обзор проблемы.

Порождение текстов на естественном языке - процесс преднамеренного построения текста на естественном языке с целью решать определенные коммуникативные задачи. Термин "текст" рассматривается как общий, рекурсивный термин, который может относится к письменному или устному высказыванию, или к отдельным частям высказывания. При порождении текстов, в устной или письменной форме, человеку важно обдумать и отредактировать производимое высказывание. Едва ли можно сказать, что большинство программ может “говорить” сегодня, в основном все они лишь выводят слова на экран. Так как для программы порождения текстов на сегодняшний день не стоит вопрос конструирования фразы, эти детали принимаются во внимание только тогда, когда они задействованы в создании программы.

Цели исходят из другой программы, возможно экспертной рассуждающей системы или обучающей программы, которая общается с пользователем на естественном языке. Произведенные тексты могут быть различной длины: от одиночной фразы, данной в ответ на вопрос, до диалогов с большим количеством предложений или толкований на целую страницу. Порождение текстов на естественном языке отличается от программ, просто использующих естественный язык. Программы,

печатающие сообщения на естественном языке, существуют со времен появления компьютеров, но сейчас, например, никто не хочет разбираться, каким образом построены сообщения об ошибках при компиляции на ФОРТРАНе, как бы правильно они не были написаны. Сообщение об ошибках ничего не "означает" для программы, которая печатает их: связь между цепочкой слов и работой программы создается программистом. Даже использование утверждений с параметром, где зафиксированная цепочка слов может быть увеличена именами или простыми описаниями, заменяющими переменные, не является собственно порождением текстов на естественном языке. Успех таких приемов как “заполнить пробелы” или “шаблон” зависит от количества и сложности ситуаций, в которых программа должна использовать их. То, что они были адекватны до сих пор для работы программы, объясняется, по большей части, относительной простотой сегодняшних программ, чем возможностями порождения с использованием метода “шаблона”.

В отличие от таких "инженерных разработок", исследование порождения текстов на естественном языке, подобно другим областям вычислительной лингвистики, имеет своей целью компьютерное моделирование человеческой способности к порождению высказываний. Основное внимание при этом сосредотачивается на объяснении двух ключевых вопросов: многосторонность и творческий потенциал. Что люди знают относительно их языка, какие процессы они при этом используют, что дает возможность им быть универсальным, изменяя тексты в форме и акцентировании, чтобы покрыть огромный диапазон языковых ситуаций?

2. Характер процесса порождения.

В отличие от организации процесса понимания, который, на первый взгляд, может следовать традиционным стадиям лингвистического анализа: морфология, синтаксис, семантика, прагматика, процесс порождения имеет существенно отличный характер. Этот факт следует непосредственно из

присущих различий в информационном потоке в двух процессах. Понимание осуществляется от формы к содержанию; порождение есть совершенно противоположный процесс. При понимании, формулировка текста (и, возможно, интонация) - "известны". Из формулировки процесс создает и выводит примерное содержание, переданное текстом и, вероятно, усилиями диктора в создании текста. Первым делом следует просмотреть слова текста последовательно, в течение чего форма текста постепенно разворачивается. Главные проблемы вызваны неоднозначностью одна форма может содержать диапазон альтернативных значений, и аудитория получает большее количество информации из ситуационных заключений, чем это может быть фактически передано текстом. Кроме того, несоответствия у диктора и аудитории модели ситуации ведут к непредсказуемым заключениям.

Порождение имеет противоположный информационный поток. Оно переходит от содержания к форме, от целей и перспектив к линейно упорядоченным словам и синтаксическим маркерам. Модель ситуации и дискурс обеспечивают основу для создания выбора среди альтернативных формулировок и конструкций, которые производит язык: первое в построении заранее обдуманного текста. Большинство систем порождения производит поверхностные тексты последовательно слева направо, но только приняв решение сверху- вниз по содержанию и форме текста в целом. Проблема генератора состоит в том, чтобы выбрать из поставленных источников, как правильно сообщить о желаемых умозаключениях аудитории и какую информацию опустить из явного упоминания в тексте.

Можно вообразить, что процесс порождение также организован, как и процесс понимания, только в противоположном порядке. К некотором смысле это верно: идентификация намерения (цели) в значительной степени предшествует любой детализации информация, которая предназначается для аудитории: планирование риторической структуры, например, в значительной степени, предшествует любой синтаксической структуре, а синтаксический контекст слова должен быть зафиксирован, прежде чем

будут известны морфологическая и суперсегментная формы, которые примет слово.

Синтаксис и словарь языка становится как ресурсами, так и ограничениями, определяя элементы, доступные для создания текста, а также зависимости между ними, которые определяют возможные правильные комбинации. Эти зависимости, и тот факт, что они по умолчанию управляют, когда информация, от которой зависит каждое решение, становится доступной, - основная причина, почему программы порождения в значительной степени следуют стандартным стадиям, определенными лингвистами. Идентификация цели предшествует выбору содержания и риторическому планированию, которое предшествует синтаксической конструкции, только потому, что это - естественный порядок принятия решения; проще следовать потоку зависимостей, чем перепрыгивать и принимать случайное решение, которое может оказаться преждевременным и несостоятельным. Сегодняшнее исследование сосредоточено как на понимании, как лучше представить решения, которые являются возможными, и зависимости среди них, так и на том, как представить ограничения и возможности раньше решений, которые встанут на место последних во время процесса порождения.

Стандартные Компоненты и Терминология. Компоненты порождения естественного языка не существуют сами по себе. Они расположены внутри человеко-машинного интерфейса, который также используют и компоненты понимания естественного языка, - ВВОД в систему. В хорошем человеко-машинном интерфейсе сегодня также хотелось бы видеть координированную графическую поддержку ввода и вывода, дополняя систему ВВОДа-ВЫВОДа естественного языка. Интерфейс может закончиться здесь, а может также включать в себя другие общедоступные компоненты, типа контроллера дискурса, который указывает генератору, какие действия нужно предпринять, а также координирует интерпретации, сделанные компонентом понимания. За интерфейсом следует -

нелингвистическое рассуждение или программа базы данных, которую пользователи используют в качестве речевого интерфейса, ею может оказаться любая система ИИ: совместная база данных, экспертная диагностическая система, ICAI обучающая программа, комментатор, программа-консультант, машинный переводчик. Тип основной программы теперь не имеет никакого значения для самой порождающей системы (генератора естественного языка).

Сегодня большинство исследователей в этой области работает, в основном, с экспертными системами, где процесс общения контролируется программой, а не пользователем. Кроме того, ЭС и интеллектуальные машинные обучающие программы, вероятно, способны понимать довольно сложные тексты, что делает их привлекательными для специалистов, готовых работать с уже разработанными системами.

Процесс порождения начинается внутри основной программы, в случае, когда, например, необходимо ответить на вопрос пользователя; или во время беседы может возникнуть потребность прервать действия пользователя, чтобы указать надвигающуюся проблему. Как только процесс инициализирован, три вида действий должны быть выполнены:

1. Идентификация целей высказывания,
2. Планирование, как эти цели могут быть достигнуты, включая оценку ситуации и доступных коммуникативных ресурсов,
3. Реализация планов в текст.

Цели должны обычно передавать некоторую информацию аудитории или побуждать их к действиям или рассуждениям. Социальные и психологические, а также практические мотивы, побуждающие человека к общению, естественно, неприменимы для сегодняшних компьютерных программ. Планирование включает в себя отбор (преднамеренное вычеркивание) информационных модулей, которые появляются в тексте (например, концепции, отношения, индивидуальность).

Реализация зависит от знания грамматики языка и правил связности дискурса, и дает синтаксическое описание текста как промежуточное представление. При этом выделяется не только лингвистическая форма, но также знание относительно критериев, которые показывают, как используются эти формы. Во многих исследованиях процесс, который проводит грамматическую реализацию, называется лингвистическим компонентом, а иногда планирование и вместе с процессом идентификации цели называется стратегическим компонентом. Обычно это - только лингвистический компонент, который имеет любое прямое знание относительно грамматики производимого языка. Какую форму эта грамматика принимает - один из самых больших различий среди проектов порождения.

Традиционно для лингвиста, грамматика - костяк в отрезке утверждения/ высказывания. Содержание утверждений - специфические факты данного естественного языка - не представляет такого интереса для лингвиста.

Аналогичная ситуация с порождением текстов, за исключением того, что запись - процедурная и декларативная - разработана, чтобы обеспечивать очень специфическую функцию, с которой традиционный лингвист не сталкивается, а именно: вести и сдерживать процесс порождения текста со специфическим содержанием и целями в присутствии специфической аудитории. Грамматика теперь ответственна за наличие выбора, который язык предоставляет для формы и словаря. Исследователи порождения должны сделать верный выбор, чтобы, используя функции различных конструкций для достижения конкретной цели. Другая функция грамматики - следить за грамотностью текста, т. е. определение зависимостей и ограничивая решения.

Технический уровень. Разноплановое развитие и творческий потенциал в порождении текстов является возможным при следующих условиях:

1. Генератор включает в себя весь объем основной грамматики;

2. Основная программа имеет сложное, разностороннее, концептуальное представление(вид);

3. Текстовый планировщик может использовать модели аудитории и дискурса.

К сожалению, такие генераторы - все еще только предмет исследования сегодня, т. к. техническая сторона остается на уровне программы, которая порождала предложения в процессе ответа на вопросы, система “непосредственной замены”, порождающая простые грамматические глагольные корректировки в целях достижения удобочитаемого текста.

When did you pick up [the green pyramid]?

While I was stacking up the red cube, a large red block, and a large green cube.

К концу 1970-ых такие системы стали достаточно популярны в работе ЭС: для перевода многочисленных правил в этих системах. Необходимость программ порождения текстов в системах с составной структурой и коммуникативным контекстом была очевидной.

Исследователи заинтересованы в более сложных текстах, нежели в контекстно-свободных представлениях, которые требуются правилами системы. В качестве примера приводится простое описание из программы Сигурда, чья цель была выяснить, как с помощью интонации выявляется группировка:

The submarine is to the south of the port. It is approaching the port, but is not close to it. The destroyer is approaching the port too.

Использование слов-ссылок “but” “too” является большим прогрессом в структурировании системы. Предложение, которое является источником в базе данных ЭС , рассуждающее о субмаринах и эсминцах, не будет обрамлено концептуальными эквивалентами таких функциональных слов, и может быть прочтено простым шаблоном, потому что ссылки специфичны и могут быть употреблены только в отдельном конкретном случае.

Еще одна техническая, пока не разрешенная, проблема - “последующая ссылка”. Какими должны быть слова-заменители, если предмет появляется больше, чем один раз в тексте? Постоянное употребление местоимений может привести к неоднозначности. В качестве примера приводится отрывок из исследований Гранвилле, который классифицирует отношения между референтом и предметом и разрабатывает правила, по которым бы могли строиться последующие ссылки.

Pogo cares for Hepzibah. Churchy likes her, too. Pogo gives a rose to her, which pleases her. She does not want Churchy’s rose. He is jealous. He punches Pogo. He gives a rose to Hebzibah. The petals drop off. This upsets her. She cries.

Неудивительно, что у исследователей, разрабатывающих основную программу, генераторы обладают наибольшей эффективностью, что дает уверенность в том, что имеется концептуальная основа для группирования отдельных предложений/ утверждений в тексте. Важным моментом на этом этапе является программа PROTEUS, разработанная Дэйви в 1974. Программа дает описание игры крестики-нолики и считается одной из программ, наиболее свободно владеющей естественным языком. PROTEUS имеет модель толкования конкретных шагов: нападение, встречное нападение, включает в себя риторический принцип, что в текст нужно помещать только наиболее существенную информацию в ситуации.

Грамматика и средства реализации выбирают описанные и сгруппированные шаги, исправляют формы, так чтобы они были грамматичны в английских предложениях, и порождают собственно текст.

Следует упомянуть и программу ERMA Клиппенгера (1974)- единственная программа на тот момент, работающая со спонтанной речью. Как люди размышляют о том, что они говорят, как они динамически планируют или меняют свои намерения относительно того, что они хотят сказать в разговоре? В целях моделирования этого процесса, Клиппенгер анализировал стенограмму речи пациента по психоанализу с тем, чтобы понять рассуждения пациента, дающие объяснение одному из параграфов

стенограммы, который ERMA могла подробно воспроизвести. Клиппенгер разработал структуру из пяти основных взаимосвязанных компонентов, участвующих в порождении спонтанного текста. Но для компьютерного программирования в 1974 реализовать этот план было не под силу, вследствие чего проект был оставлен.

По сути дела, программы PROTEUS Дэйви и ERMA Клиппенгера являются самыми старшими в этой области. Во-первых, потому что до начала 80-ых сравнительно мало людей работало над проблемой порождения , во-вторых, сама проблема достаточно сложна, намного сложнее проблемы понимания речи. На самом деле, проблемой серьезно занимались в начале 1970-ых. Но справедливо отметить, что на важной конференции по данной проблеме в 1975г представленные отчеты о проделанной работе не нашли должного отклика, после чего исследования по порождению естественного языка были почти приостановлены до начала 1980-ых.

До 80-ых специалисты в области ИИ склонны были считать проблему порождения достаточно легкой. В самом деле, разве трудно взять утверждение из некоторого речевого фрагмента, связать его с определениями, хранящимися отдельно, и произвести, например, следующее “The big black block supports a green one”. Это было под силу SHRDLU Винограда уже в 1970г. Если бы можно было ограничиться этими знаниями, то, на самом деле, не возникало бы проблем. Но вариативность языка не давала такой возможности. Каким образом человек представляет грамматические знания, которые позволяют генератору использовать синтаксическую структуру предложения в целях создания соответствующего относительного предложения (“the green block that’s supported by the big red one”, “a green one”, а не “a green block”), а также вообще иметь представление о возможности таких относительных предложений и подобных замен.

3.Общие подходы к проблеме порождения текстов на ЕЯ.

Трудно идентифицировать общие элементы в различных проектах исследования по порождению естественного языка. Напротив, в

исследованиях по пониманию речи можно выделить несколько основных подходов к проблеме: использование расширенных сетей переходов, семантические грамматики, рабочие системы, основанные на представлении концептуальной зависимости, процедурная семантика и многое другое. Исследование порождения не может дать подобной классификации, поскольку очень мало специалистов ставили эту проблему во главу угла. Большие исследовательские группы, полностью сконцентрировавшиеся на вопросе порождения естественного языка, начали создаваться в последние два года. Основная проблема состоит в отсутствии общего отправного пункта, конкретной основы для сравнения, что осложняет работу, не дает возможности для взаимопомощи между исследователями: практически невозможно проверить свои эксперименты на системе другого разработчика. Однако имеются общие нити, связывающие различные проекты: похожие подходы, похожие представления, похожие грамматики.

Существует два вопроса, представляющих общий интерес. Первый вопрос: как сопоставить многообразие форм в естественных языках, чтобы разработать их функциональное использование, ответить на вопрос, почему человек использует одну форму, а не другую, а далее формализовать этот процесс.

Второй вопрос - это контроль над процессом порождения. Что определяет выбор говорящего в данной языковой ситуации? Как человек организовывает и представляет промежуточные результаты? Какими знаниями о зависимостях между вариантами выбора должна обладать система? Как представлены эти зависимости и как они могут влиять на алгоритмы управления?

4. Лексический выбор, фразовые словари. Обработка грамматики.

Лексический Выбор. Некоторые подходы к машинному пониманию основываются на небольшом наборе базисных элементов и, формулируют знания программы в виде набора выражений к базисным элементам, что

упрощает работу программы: становится легче выводить умозаключения, потому что при помощи базисных элементов они распределяются в естественные группы. Однако, сведение диапазона человеческих действий к определенному набору, например, лишь к 13 концептуальным базисным элементам, означает, что специфика значений распределяется в выражениях и извлекается оттуда каждый раз, если во время порождения необходимо использовать глаголы со специфическим значением. Голдман первый провел исследования по использованию сетей распознавания. Он показал, как производится выбор слова, в отрыве от основных базисных элементов. Например, из базисного элемента действия «глотать» можно получить глаголы «пить», «есть», «вдыхать», «дышать», «курить», или «проглотить», как бы проверяя при этом, был ли проглоченный объект жидкостью или дымом.

Проект сети распознавания заставляет исследователя порождения выходить за рамки основных различий типов объектов и включать контекстные факторы, напр., эмоциональные рассуждения говорящего.

Фразовые словари. Какое слово ассоциируется с простыми понятиями, типа «парикмахер» или «брить», является очевидным; однако, для объектов в комплексных основных программах, лексический выбор может оказаться более проблематичным. Помощь в этой ситуации может оказать использование фразового словаря. Это понятие было введено в 1975 Бекером и с тех пор стало важным инструментом систем порождения. С лингвистической точки зрения, «фразовый» словарь – концептуальное расширение стандартного словаря, включающее все непроанализированные фразы, - на той же самой семантической основе, что и словарь отдельных слов. Это обеспечивает фиксацию незаконсервированных идиом и различных речевых способов, которые люди используют каждый день. Так как люди используют эти « фиксированные фразы « как нерасчленимое целое, программы должны научиться делать то же самое.

Обработка Грамматики. В изучении порождения выбор формализации представления грамматики языка всегда связывался с выбором протокола контроля. Известны три основных подхода к решению этого вопроса:

1. грамматика как независимый корпус предложений и фильтр к ним (например, объединенная функциональная грамматика);

2. использование грамматики с целью выявления всех возможных поверхностных структур, доступных для языка; затем проведение выбора и реализации среди данных поверхностных структур (смысловые подходы);

3. грамматика как структура пересеченного графа, который контролирует весь процесс, как только создается план текста (план выражения) (грамматика расширенных сетей переходов, а также систематическая грамматика).

Контрольные вопросы.

1. Назовите основные задачи обработки текстов.
2. Основные этапы обработки текстов.
3. Каков требуемый объем словарей, необходимый для поддержки лингвистических информационных технологий (укажите порядок величины: 100, 1000, 10 000 и т.д.)?
4. Что означают выражения example-based analysis, case-based analysis?
5. В чём заключается семантический анализ текста?
6. Виды программного обеспечения по обработке текста.
7. Мировой рынок коммерческого ПО для автоматического чтения, реферирования и аннотирования текста.
8. Назовите самую популярную технологию распознавания письменных текстов.
9. В каких лингвистических технологиях необходимо использовать лингвистические фильтры?
10. Расшифруйте аббревиатуру OCR.

11. Назовите 2 самые известные англоязычные OCR-системы.

12. Что представляет собой лингвистическая технология ALICE?

13. Какие из перечисленных процедур пока не могут быть реализованы в полном объеме в коммерческих системах?

14. Что такое "машинно-ориентированный словарь" как объект в информационных технологиях? (текстовый файл, база данных, исполняемый файл, другое)

15. Какие два подхода к решению задач автоматизированной обработки текста вы знаете?

16. В чем заключается сложность задачи обучения языкам?

17. Выполнением каких задач реализуются принципы построения обучающих программ?

18. В чём заключается бихевиористский подход к выбору метода обучения?

19. В чём заключается когнитивно-интеллектуальный подход к выбору метода обучения?

20. Назовите и охарактеризуйте два типа моделей учебной среды.

21. Перечислите уровни структурно-управляемого обучения.

22. Что такое АУК?

23. Назовите три способа использования компьютеров в обу-чении.

24. Что такое TOEFL?

Тема 8: Концептно-ориентированная модель памяти переводов.

1. Перевод текстов. Общие понятия. Восемь типов памяти перевода.
2. Выделение терминов и анализ терминологии.
3. Аспекты использования памяти переводов.
4. Основные принципы работы.
5. Пути расширения возможностей.

1. Перевод текстов. Общие понятия

Существуют различные определения понятия **перевод текстов.** В качестве рабочего определения примем следующее: «Перевод есть вид человеческой языковой деятельности, в результате которой некоторый текст на одном языке ставится в соответствие тексту на другом языке, при этом обеспечивается их смысловая эквивалентность» . Слово «перевод» понимают двояко: как сам процесс перехода от текста на одном языке к этому же тексту на другом языке, так и результат этого перехода, т.е. тот текст, который получается в результате перевода.

Переводом текстов человек начал заниматься еще в античном мире — более 20 веков назад. Одним из первых основные принципы перевода сформулировал Марк Тулий Цицерон (106—43 гг. до н.э.), древнеримский политический деятель, оратор и писатель. Он переводил произведения древних греков на латинский язык и считал, что переводить следует не слова, а мысли, не букву, а смысл, в соответствии с условиями и духом своего языка.

Однако такой правильный взгляд на теорию перевода не являлся общепринятым как в древние, так и в последующие средние века (конец V—середина XVII в.). Вред научной теории перевода принесли переводы древнееврейских религиозных текстов на другие языки. При этом любое отступление от оригинала рассматривалось как ересь. Такой подход привел к возникновению и развитию ***пословного перевода,*** буквализму, искажавшему смысл и стиль текста исходного языка.

В эпоху Возрождения (XIV—XVI вв.) появились шедевры мировой литературы: произведения Ф. Рабле, У. Шекспира, С. Сервантеса, Ф. Петрарки, Дж. Боккаччо и целого ряда других писателей и поэтов, что привело к резкому возрастанию количества переводов. В это время особенно отчетливо стали осознаваться трудности перевода художественных текстов. Как протест против пословного, буквального перевода начинает возникать ***вольный перевод,*** при котором на другом языке передается лишь общая идея текста исходного языка. Такой перевод

не учитывает стилистические особенности автора исходного текста, реалии места и времени описываемых в нем событий и многие другие особенности переводимого произведения. В процессе вольного перевода тексты исходного языка порой искажались до неузнаваемости.

В течение многих веков речь шла лишь о переводе художественных текстов. И многие переводчики, литераторы, критики сомневались, что при этом можно передать художественную ценность оригинала. Известный лингвист Вильгельм фон Гумбольдт (1767— 1835) писал об этом так: «В сущности всякий перевод представляется мне попыткой решить неразрешимую задачу, ибо переводчик оказывается между Сциллой и Харибдой: либо он в ущерб традиционным пристрастиям и языку своей нации держится слишком близко к оригиналу, либо — напротив — приносит оригинал в жертву особенностям своего языка и национального вкуса. Благополучно миновать обе эти особенности не только трудно, но и невозможно». Как видно, В. фон Гумбольдт стоял на позициях «непереводимости»: возможен или буквальный, или вольный перевод. Третьего не дано.

Идеи перевода в значительной степени интересовали И. В. Гёте, Н. В. Гоголя, А. С. Пушкина, В. Г. Белинского и других писателей, поэтов, критиков, переводчиков. Их труды, а также работы их последователей постепенно помогли подойти к созданию истинно научных теорий перевода, утверждающих возможность хорошего перевода текста с любого языка на другой язык. Такие теории начали создаваться в 50—60-е годы XX века. Не останавливаясь подробно на этом вопросе, отметим их одну общую особенность: данные теории опираются на метод моделирования процесса перевода текстов человеком.

Существование нескольких современных теорий перевода объясняется, во-первых, тем, что объект моделирования — умственные действия человека в процессе перевода — сложен и недоступен прямому наблюдению. Во-вторых, теории перевода решают разные задачи,

связанные с языками, психологией человека, культурой и традициями страны исходного языка (ИЯ) и переводного языка *(ИЯ)*. Все эти теории (модели) гипотетичны. Степень их «пригодности» для той или иной пары языков проверяется практикой.

Насчитывается большое количество типов и видов переводов, существуют различные подходы к их классификации. Рассмотрим некоторые из них.

В зависимости от ***переводного материала*** различают:

1) перевод художественной литературы;

2) научно-технический перевод (перевод научно-технических, военных, юридических текстов и т.д.);

3) общественно-политический перевод (перевод газет, политических журналов и т.д.);

4) бытовой перевод (перевод текстов разговорно-бытового характера).

По ***форме презентации*** текста перевода и текста оригинала выделяют:

1) письменный перевод;

2) устный перевод.

По признаку ***основной прагматической функции*** перевода (с какой целью делается перевод) различают:

1) практический перевод (в целях получения новой технической, экономической, политической, эстетической и другой информации);

2) учебный перевод (для обучения основам перевода);

3) экспериментальный перевод (например, для оценки умения переводчика и качества работы);

4) эталонный перевод (как образцовый перевод, с которым сравниваются другие переводы).

По ***степени механизации*** процесса перевода выделяют:

1) традиционный («ручной») перевод, выполняемый человеком;

2) перевод, выполняемый человеком с помощью компьютера (например, когда в памяти электронного устройства находится

двуязычный словарь и компьютер по запросу пользователя ищет переводные эквиваленты иностранных слов);

3) перевод, выполняемый компьютером с помощью человека (например, когда компьютер по специальной программе делает перевод, а за справками — неизвестным переводным эквивалентом, неизвестной синтаксической структурой и т.д. — обращается к человеку-интерредактору);

4) машинный или автоматический перевод (выполняется компьютером без вмешательства человека).

Необходимость создания систем машинного перевода

Как было отмечено выше, машинный перевод осуществляется компьютером самостоятельно, без вмешательства человека. Более строго это понятие может быть определено следующим образом*:* **машинный,** или **автоматический перевод,** (МП или АП)| (англ.: *machine translation, automatic translation)* — это выполняемое компьютером действие по преобразованию текста на одном! естественном языке в текст на другом естественном языке при сохранении эквивалентности содержания, а также результат такого действия. Человек, как правило, в той или иной мере **учас**твует в подготовке машинного перевода или в его «доведении» | до удобочитаемого вида. Чаще всего до ввода в компьютер переводимый текст специальным образом готовится человеком-предредактором. Он упрощает структуру предложений, выделяет терминологические обороты, указывает класс слов для омографичных форм и т.д. Выданный компьютером перевод для удобства чтения подвергается стилистической правке человеком-постредактором.

В течение последних десятилетий большое внимание во всех странах уделяется научно-техническому переводу — основному источнику новых научных и технических знаний. Европейское; экономическое сообщество

имеет свою службу перевода, включающую около 2 тыс. переводчиков. Они переводят в год примерно 600 000 страниц текстов с пяти языков и также не справляются со все возрастающими потоками заказов на переводы. Это приводит к тому, что до специалистов различных стран зарубежная информация доходит с большим опозданием (порой через 5—10 лет). Единственным способом увеличения скорости перевода является использование в переводческой деятельности современных компьютеров, которые в миллиарды раз быстрее человека могут выполнять необходимые для перевода логические действия. Человек-переводчик тратит 20 % своего времени на перевод, 40 % — на поиск по словарю незнакомых слов и 40 % — на пере печатку и оформление перевода. Компьютер же в процессе перевода тратит 95 % времени непосредственно на перевод и 5 % на пополнение словаря. Если максимальная производительность труда переводчика составляет 4—5 авторских листов в месяц, то такая, например, система машинного перевода, как SYSTRAN, переводит в час до 1 млн словоупотреблений (около 120 авторских листов). Эти количественные характеристики свидетельствуют о преимуществе компьютерных систем. Однако качество такого перевода значительно уступает переводу, сделанному человеком. И тем не менее проблемами машинного перевода сейчас активно занимаются во всех развитых странах: США, Франции, Японии, Германии, Китае и т.д. Ежегодно по машинному переводу проводится несколько крупных международных конференций, в разных странах постоянно издаются журналы и книги по этой проблеме. Первый эксперимент по машинному переводу был проведен в США в Джорджтаунском университете 7 января 1954 года. ЭВМ перевела с русского языка на английский несколько достаточно простых предложений по физике. В России первый машинный англо-русский перевод был выполнен в 1955 году.

Основные понятия и проблемы машинного перевода

Чтобы понять, какие проблемы стоят перед специалистами по машинному переводу, рассмотрим, что нужно знать для того, чтобы перевести текст с одного языка на другой. Это прежде всего:

1) язык (лексика, грамматика), с которого осуществляется перевод;

2) язык (лексика, грамматика), на который переводится текст;

3) предметное содержание переводимого текста (реалии места и времени, область специальных знаний, индивидуальные особенности переводимого автора и т.п.).

Выше было отмечено (см. с. 77), что все теории перевода текстов построены на основе моделирования деятельности человека-переводчика. Это в полной мере относится и к теориям машинного перевода. Однако мало что известно о детальных процедурах перевода текста человеком. В общем виде процесс перевода текста состоит из следующих трех этапов*:*

1) постижение текста на ИЯ;

2) интерпретация текста на ИЯ;

3) перевыражение текста ИЯ и создание текста на ПЯ.

Суть ***постижения исходного текста*** заключается в понимании того, о чем говорится в исходном тексте. При этом выделяют три формы понимания:

а) филологическое, или дословное, понимание текста. При этом предполагается, что человек, читающий текст, понимает значение всех слов этого текста, смысл всех его предложений и содержание всего текста;

б) стилистическое понимание — это понимание настроения] героев, их иронии, трагических и комических эффектов и т.д.;

в) понимание идейного замысла автора. Оно выражается в осознании следующего: для чего автор писал свое произведение, что он

хотел сказать читателю, какие цели преследовались автором при создании текста.

Переходя к характеристике *интерпретации текста,* необходимо отметить, что какие бы пары языков ни использовались в качестве исходного и переводного, в таких языках нет полного семантического тождества: слову ИЯ, как правило, соответствует несколько слов ПЯ; один тип предложения ИЯ может быть передан несколькими синтаксическими структурами ПЯ; любая связная последовательность предложений ИЯ передается не сколькими допустимыми последовательностями ПЯ. Поэтому; ***лингвистически точный перевод текстов невозможен в принципе.*** Возможна лишь правильная интерпретация текста на исходном языке средствами переводного языка. При этом переводчик должен уметь передать объективный смысл всего произведения, сводя к минимуму свое субъективное отношение к описываемому в тексте.

Процесс ***перевыражения текста*** **ИЯ** в текст ПЯ носит творческий характер. Переводчику надо не только заменить слова и предложения ИЯ словами и предложениями ПЯ, но и сделать это стилистически верно, художественно. Текст перевода должен сохранить и предметное содержание исходного текста, и его идейный замысел. Результатом процесса перевыражения должен стать такой текст ПЯ, который считается явлением своей литературы, переводного языка и в то же время сохраняет оттенок чужого.

Перевод текстов — задача, которая до сих пор не имеет однозначного алгоритма решения. Многие действия, которые выполняет человек в процессе перевода текста, неясны, и это создает огромные трудности в ходе построения алгоритмов машинного перевода. Тем не менее большинство специалистов по переводу отмечают, что в процессе перевода текста выполняется следующая последовательность действий:

1) морфологический анализ каждого слова предложения ИЯ;

2) синтаксический анализ каждого предложения текста ИЯ;

3) синтаксический синтез каждого предложения ПЯ;

4) морфологический синтез каждого слова предложения ПЯ. Человек выполняет эти действия, опираясь на знания языка или словари. Очевидно, компьютер, осуществляющий перевод! текстов, должен уметь выполнять те же самые действия.

В процессе морфологического анализа слов предложения ИЯ каждое слово получает наборы лексико-грамматических признаков (часть речи, род, число, падеж, время, лицо, управление и т.д.). Компьютер может сформировать такие наборы либо по формальным признакам слов (суффиксам, окончаниям, приставкам), либо с опорой на специальный автоматический словарь. В нем каждой словоформе уже даны соответствующие лексико-грамматические признаки, и в процессе морфологического анализа слова компьютер берет их из словаря в готовом виде.

Синтаксический анализ предложения ПЯ сводится к поиску основных членов предложения (группы подлежащего, группы сказуемого и т.п.).

Синтаксический синтез предложения ПЯ заключается в создании предложения ПЯ определенной синтаксической структуры, определяемой правилами ПЯ и синтаксической структурой предложения ИЯ. Чтобы компьютер мог выполнить это задание, он должен иметь в памяти сведения о синтаксических структурах ИЯ, ПЯ и их соответствиях друг другу.

Еще одна задача этапа синтаксического синтеза связана с заменой слов ИЯ их переводными эквивалентами из словаря ПЯ.

Морфологический синтез каждого слова предложения ПЯ сводится к постановке слов ПЯ в нужном числе, роде, падеже, времени и т.д. Для этого компьютер должен владеть знаниями о лексико-грамматических признаках каждого слова ПЯ, которые берутся из упомянутого выше автоматического словаря.

Таким образом, система машинного перевода текстов, например с английского языка на русский, должна состоять из компонентов, показанных на схеме 6.

Назначение четырех представленных на этой схеме подсистем ясно из вышеизложенного. Рассмотрим более подробно две другие составляющие — англо-русский автоматический словарь и англо-русские синтаксические соответствия.

Восемь типов памяти перевода.

Основные определения

Концепт- не зависящее от конкретного языка понятие, соответствующее реальной или абстрактной сущности, свойству, действию, либо иному элементу, отражающему связь между другими понятиями.

Термин- слово или словосочетание на заданном языке, обозначающее в этом языке конкретный концепт.

Терминология- множество обозначающих один и тот же концепт терминов из различных языков.

Сегмент- непрерывный фрагмент текста, состоящего из терминов одного языка, обозначающих связанную по некоторому критерию группу концептов.

Вариант сегмента- сегмент, похожий на исходный по некоторому критерию.

Исходный язык- язык, с которого осуществляется перевод.

Целевой язык- язык, на который осуществляется перевод.

Языковая пара- упорядоченная пара сегментов, объявленных переводчиком эквивалентными по смыслу, первый из которых содержит термины на исходном языке, а второй- на целевом.

Восемь типов технологии перевода.

В современных профессиональных средах перевода возможности вычислительной техники используются на различных этапах и уровнях.

Всего можно выделить восемь способов применения компьютера при переводе (таблица 7).

Таблица 7

	Уровень терминов	Уровень сегментов
До перевода	• Выделение терминов • Анализ терминологии	• Сегментация текста
Во время перевода	• Автоматический поиск терминологии	• Поиск языковых пар в памяти переводов • Машинный перевод
После перевода	• Проверка соответствия терминологии	• Проверка целостности сегментов, формата и грамматики

2. Выделение терминов и анализ терминологии.

На этом этапе производится исследование текста с целью выяснения, какие слова или словосочетания могут быть взяты в качестве терминов. После того, как определен термин на исходном языке, осуществляется анализ терминологии на предмет того, какой термин на целевом языке следует выбрать для обозначения нужного концепта. Например, если в исходном тексте встретилось словосочетание "операционная система" то программа должна проанализировать его в качестве возможного термина, даже если в системе уже определены термины "операционный" и "система".

Автоматический поиск терминологии

Данный процесс может быть сравнен с машинным переводом на уровне отдельных терминов. Суть его заключается в том, что в процессе работы над текстом переводчик имеет возможность видеть варианты перевода для каждого термина, и быстро вставлять нужный перевод в текст на целевом языке, не рискуя допустить опечатку.

Проверка соответствия терминологии

После того, как перевод выполнен, компьютер осуществляет проверку того, что все вхождения каждого из терминов были переведены одинаково. Например, если термин "операционная система" был заменен при своем первом вхождении на "operatingsystem", а при втором вхождении на "operationalsystem", то должно быть выдано соответствующее предупреждении о нарушении единства терминологии.

Сегментация текста

Разбиение текста на сегменты является важным подготовительным этапом для полной или частичной автоматизации перевода. Сегменты должны по возможности содержать фрагменты текста, грамматически независимые друг от друга. Иными словами, должна быть обеспечена возможность корректного перевода каждого сегмента независимо от других. Обычно разбиение на сегменты выполняется по знакам пунктуации.

Поиск языковых пар в памяти переводов

Автоматическая память переводов, или просто память переводов (TranslationMemory), подразумевает, в первую очередь, просмотр ранее переведенных текстов. Она сравнивает переводимый в текущий момент текст с тем, что хранится в базе, "вспоминает" сегменты, которые изменились незначительно, и предлагает использовать их перевод повторно. Разумеется, критерии сходства сегментов могут быть различны, и они играют очень важную роль в расширении возможностей памяти переводов.

Машинный перевод

Данный способ перевода заключается в алгоритмической обработке исходного текста, в ходе которой происходит разбор сегментов, выделяются отдельные термины и отношения между ними, после чего осуществляется замена всех терминов на соответствующие термины целевого языка в нужной форме и взаиморасположении. Машинный перевод (MachineTranslation) применим только в очень узком контексте и требует значительного постредактирования переведенного текста.

Проверка целостности сегментов, формата и грамматики

Данные действия выполняются по окончании перевода и имеют своей целью проверить, все ли сегменты остались на своих местах, сохранилась ли форматирующая информация, и корректен ли результирующий текст с точки зрения грамматики целевого языка.

Среди перечисленных технологий наибольший интерес представляют терминологические словари и память переводов, поскольку именно от их эффективности зависит скорость и качество перевода. Технология построения терминологических словарей достаточно хорошо проработана и основана на принципах, аналогичных тем, что применяются в обычных двуязычных словарях. Разбиение текста на термины обычно осуществляется по пробелам с дополнительным привлечением некоторого морфологического анализа.

Сложнее обстоит дело с организацией памяти переводов. Наряду с тривиальной задачей поиска языковой пары, включающей сегмент, идентичный заданному, память переводов должна обеспечивать возможность поиска сегментов, похожих на данный по некоторому критерию. Таким образом, центральной проблемой классической памяти переводов является построение анализатора таких "нечетких совпадений" (fuzzymatches), характеристики которого и определяют преимущества и недостатки каждой конкретной системы профессионального перевода.

3. Аспекты использования памяти переводов.

Как следует из вышеизложенного, основой функционирования любой системы памяти переводов являются ранее переведенные тексты. Множество этих текстов постоянно пополняется новыми переводами, вследствие чего, процент автоматически переводимых сегментов, постепенно растет. Это означает, что для наиболее эффективного использования памяти переводов, все тексты должны содержать достаточное количество похожих фраз. Такое положение вещей имеет место в документации на различного рода продукты. Это обусловлено двумя факторами. Во-первых, документацию принято составлять максимально простым языком, лаконично и в строгих терминах.

Во-вторых, с появлением новых версий и модификаций поставляемого потребителям продукта содержание документации меняется лишь в незначительной степени. Память переводов, в подобных случаях, избавляет переводчика от необходимости по несколько раз переводить идентичные фрагменты текста, входящие в разные документы.

В то же время, использование памяти переводов требует от переводчика специальной подготовки, а также наличия соответствующего аппаратного и программного обеспечения. Другим негативным фактором является то, что для обеспечения ожидаемого эффекта все переводы должны быть сделаны в одной и той же среде, либо в средах, совместимых по формату представления данных. Наконец, полезный эффект памяти переводов проявляется с заметной отсрочкой во времени, требуя поначалу дополнительных капиталовложений.

Резюмируя вышесказанное, можно выделить три условия применимости рассматриваемой технологии:

1. большой объем перевода;
2. однотипность переводимых текстов;
3. готовность к отсроченному возврату капиталовложений.

4. Основные принципы работы

Память переводов представляет собой базу данных, хранящую языковые пары, и определенный механизм поиска. Несмотря на то, что различные профессиональные среды перевода, такие как "Translator'sWorkbench" фирмы Trados, "Transit" фирмы Star, "DejaVu" фирмы Atril, имеют, по-видимому, различную реализацию этого механизма ("по-видимому", поскольку алгоритмы не придаются огласке), общая идея становится ясной после изучения примеров. Поэтому с примеров и начнем.

Пусть в исходном тексте встречаются следующие фразы:

"Температура регулируется поворотом ручки."
"Температура регулируется поворотом ручки по часовой стрелке."
"Напор воды регулируется поворотом ручки по часовой стрелке."

Если сегментация выполняется по предложениям, то каждая из приведенных фраз попадет в отдельный сегмент. Пусть первый сегмент был переведен человеком следующим образом:

"The temperature can be adjusted by turning the knob."

Языковая пара, состоящая из исходного и переведенного сегментов, заносится в память переводов. Когда переводчик доходит до второй фразы примера, система определяет сходство и выводит на экран следующую информацию: таблица 8.

Таблица 8

Текущий сегмент	*Температура регулируется поворотом ручки по часовой стрелке*
Найденный сегмент	*Температура регулируется поворотом ручки*
Перевод	*The temperature can be adjusted by turning the knob*
Степень сходства	*~70%*

Теперь переводчик имеет возможность частично воспользоваться уже сделанным переводом, учтя различия: *"The temperature can be adjusted by turning the knob clockwise."*

После того, как сегмент, соответствующий второй фразе примера помечается как переведенный, в памяти переводов появляется еще одна языковая пара. Тем самым, когда дело доходит по третьей фразы, система уже имеет возможность показать переводчику два похожих варианта: таблица 9.

Таблица 9

Текущий сегмент	*Напор воды регулируется поворотом ручки по часовой стрелке*
Найденная языковая пара 1	*~~Температура~~ регулируется поворотом ручки по часовой стрелке*

	The temperature can be adjusted by turning the knob clockwise
Степень сходства	*~65%*
Текущий сегмент	*Напор воды регулируется поворотом ручки по часовой стрелке*
Найденная языковая пара 2	*~~Температура~~ регулируется поворотом ручки*
	The temperature can be adjusted by turning the knob
Степень сходства	*~40%*

Воспользовавшись, к примеру, первым из предложенных вариантов, переводчик быстро расправляется с оставшейся частью фразы:

"The water head can be adjusted by turning the knob clockwise."

Эффективность работы памяти переводов во многом определяется тем, насколько удачно решены следующие задачи:

1. сегментация;
2. обработка специальных символов и форматирующей информации.

Очевидно, что с увеличением размера сегментов будет уменьшаться число полных совпадений (и увеличиваться число частичных), что сильно повысит ресурсоемкость процедур поиска и потребует от переводчика значительных усилий в изучение предоставленных ему в качестве вариантов перевода языковых пар. С другой стороны, уменьшение размера сегментов сделает их малопригодными для повторного использования, поскольку сильно возрастет влияние контекста на перевод. Оптимальной единицей сегментации чаще всего оказывается фрагмент предложения, ограниченный знаками препинания. Во избежание ошибочной сегментации по точкам внутри аббревиатур и других подобных случаев используют регулярные выражения и списки исключений.

Вторая проблема обусловлена тем, что в тексте кроме букв зачастую присутствуют иные символы, как то: маркеры внедренных в документ

объектов, закладки, перекрестные ссылки, переключатели свойств шрифта. Все эти инородные элементы в ряде случаев могут повлиять на перевод. Например, выделенное курсивом слово может при переводе быть взято в кавычки и попасть в результирующий текст в неизменном виде. Для управления поведением анализатора в таких ситуациях во многих программных продуктах предусмотрены специальные настройки, в том числе, основанные на применении регулярных выражений.

5. Пути расширения возможностей

Поскольку функцией памяти переводов является поиск в базе данных переведенных фрагментов заданного сегмента, то пределом ее возможностей является, очевидно, выборка, максимально покрывающая исходный сегмент и не содержащая никакой избыточной (лишней) информации.

Попытаемся выделить возможные варианты повышения качества памяти переводов, воспользовавшись приведенным ранее примером. Выберем и рассмотрим две языковые пары: таблица 10.

Таблица 10

Языковая пара 1	*Температура регулируется поворотом ручки по часовой стрелке*
	The temperature can be adjusted by turning the knob clockwise
Языковая пара 2	*Напор воды регулируется поворотом ручки по часовой стрелке*
	The water head can be adjusted by turning the knob clockwise

Сходство сегментов на исходном языке позволяет сделать предположение, что их переводы, то есть сегменты на целевом языке также должны быть похожи. Коль скоро это так, что возникает резонное желание выделить из двух приведенных языковых пар общую часть и представить ее в виде новой языковой пары. Выполнив несложную операцию пересечения строк, получаем следующий результат: таблица 11.

Таблица 11

Языковая пара 3	*Регулируется поворотом ручки по часовой стрелке*
	can be adjusted by turning the knob clockwise

Теперь для любого сегмента, включающего фрагмент " *регулируется поворотом ручки по часовой стрелке"*, может быть выбрана языковая пара номер 3, содержащая только необходимый перевод для фрагмента.

Однако, не всегда все так хорошо. Чуть более внимательный взгляд на этот пример сразу же заставит нас признать, что создание таких "укороченных" языковых пар эквивалентно уменьшению размера сегмента, а мы помним, чем это грозит. Маленький фрагмент текста, в особенности, если он не ограничен никакими знаками препинания, едва ли может быть правильно переведен без учета контекста. Следовательно, при выделении общих частей в двух используемых уже языковых парах необходимо руководствоваться теми же принципами, что и при начальной сегментации исходного текста.

К тому же, не стоит забывать, что пересечение сегментов на исходном языке не обязательно изоморфно пересечению сегментов на целевом языке. Это связано с различиями правил грамматики в разных языках, порядка слов, соответствия слов понятиям. Поэтому осмысленное значение целевого сегмента языковой пары, образованной пересечением, можно ожидать только при:

1. значительном размере обоих сегментов вновь образованной языковой пары;
2. эвристически определенном изоморфизме пересечения сегментов на исходном и целевом языке (например, если пересечение осуществлено по знакам пунктуации);
3. морфологическом и синтаксическом анализе результата пересечения с привлечением технологии машинного перевода.

Еще одной немаловажной задачей при реализации механизма описанных манипуляций с языковыми парами является создание некоторого формализма, позволяющего однозначно определить, какие именно пары

должны подвергаться обработке, как именно должен формироваться результат, как должен осуществляться поиск сегмента и каковы критерии сравнения сегментов при поиске. Этим мы теперь и займемся.

Тема 9: Многоуровневая модель памяти переводов

1. Представление данных.
2. Поиск и добавление языковых пар.
3. Вычисление пересечения языковых пар.

1. Представление данных

Структура реально используемых ныне реализаций памяти перевода является одноуровневой и подразумевает наличие упорядоченного списка языковых пар. С введением механизма выделения в парах общих частей подобная организация данных окажется неудобной. Действительно, для любых двух пересекающихся языковых пар в базе будет создаваться дополнительный элемент, содержание которого будет полностью дублироваться в обеих языковых парах. От избыточности удастся избавиться, если удалить вынесенную общую часть из обеих пар, а на ее место поставить ссылку на вновь созданную языковую пару (рис. 11).

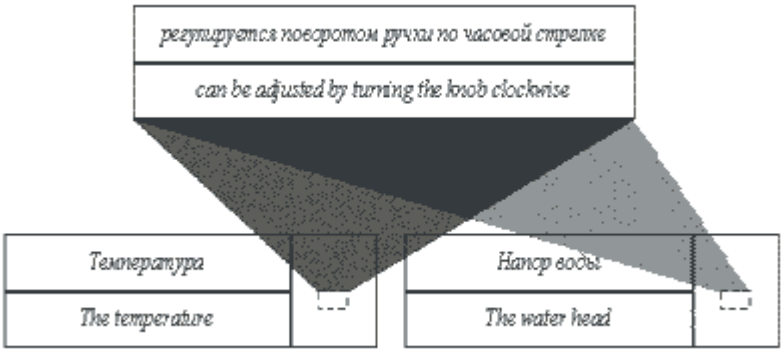


Рис. 11

Повторяя данную процедуру каждый раз, когда обнаруживается очередное пересечение, мы, в конечном счете, получим направленный граф, узлами которого являются языковые пары, а дугами- отношения включения.

Еще одной оптимизацией является разделение исходного и целевого сегментов каждой языковой пары. Это имеет смысл, поскольку не так уж

редки случаи, когда одна и та же исходная фраза переводится на целевой язык по-разному, что порождает две языковые пары с одинаковым первым элементом. Помимо этого, пересечения исходных и целевых сегментов не всегда изоморфны, и их результаты не обязательно образуют языковую пару.

Поэтому разделим языковые пары, и получим два ориентированных графа, "синхронизированных" друг относительно друга (рис. 12). При этом отношения, связывающие вершины разных графов, могут быть различной местности, но обязательно обладают свойством симметричности.

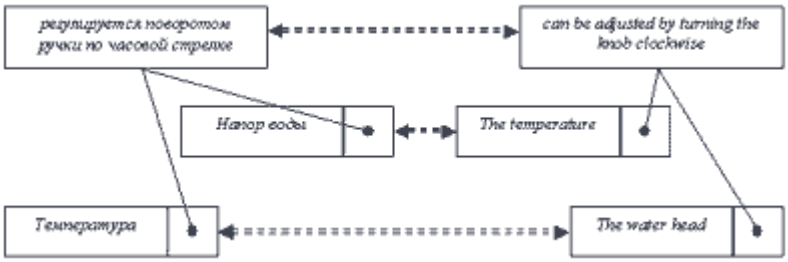


Рис.12

Развивая идею дальше, вспомним, что в нашем распоряжении уже имеется аппарат, позволяющий формализовать представленную схему. Это-объектно-ориентированный подход. В самом деле, каждому сегменту в базе может быть сопоставлен отдельный класс, отношение включения по своим свойствам идентично отношению наследования, а горизонтальные связи, формирующие языковые пары могут быть заданы механизмом ассоциирования, либо введением в класс метода Translate ("перевести").

Тем не менее, предложенная модель пока что не описывает внутреннюю структуру сегментов, которую необходимо анализировать при создании новой языковой пары на основе пересечения существующих. Для решения этой задачи разобьем все сегменты на отдельные слова (по пробелам). Теперь каждый сегмент может быть представлен классом, являющимся производным от всех слов, образующих текст сегмента. При этом совершенно необязательно иметь перевод для каждого слова, поскольку это будет отражено лишь отсутствием соответствующей связи между узлами графов, а метод Translate будет возвращать "нулевое" значение.

Памятуя о том, что в состав среды перевода помимо памяти переводов входит также терминологический словарь, деление сегментов можно осуществлять не по словам, а по терминам. Наличие терминологического словаря дает и еще одну возможность. Для каждого вхождения термина в сегмент можно определить его начальную форму, то есть ту, в которой он входит в базу словаря. Эта начальная форма послужит как бы абстрактным базовым классом для каждого вхождения термина, а конкретное вхождение будет содержать определение значений заданных в базовом классе атрибутов: род, число, падеж и т. п.

Последнее нововведение подталкивает нас к мысли о том, терминологический словарь можно слить воедино с памятью переводов, представив, тем самым, все ресурсы переводчика в виде универсальной модели (рис. 13).

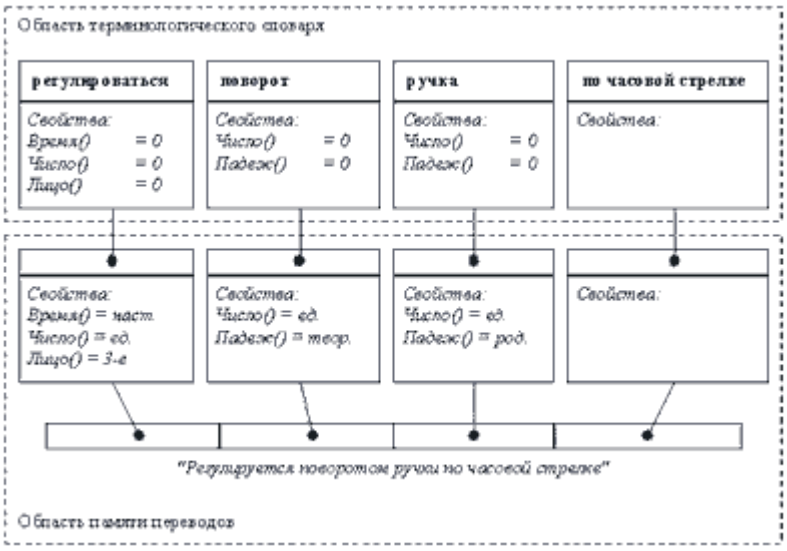


Рис. 13

2. Поиск и добавление

До тех пор, пока память переводов была линейной, сегменты неделимыми, а сравнение строгим, решение задачи поиска сводилось к введению отношения строгого лексикографического порядка над множеством сегментов на исходном языке. Иными словами, определялся оператор "меньше", на основе которого можно было осуществить обыкновенный двоичный поиск, и проверку на равенство. С введением

оператора "нечеткого совпадения", который позволял оценить степень сходства для любых двух сегментов, решение проблемы поиска резко усложнилось и, без дополнительных ухищрений с различного рода индексацией, стало эквивалентно задаче полного перебора. Предложенная многоуровневая модель памяти переводов, собственно, и предоставляет некоторый механизм неявной индексации: каждое входящее в сегмент слово, по сути, идентифицирует некоторое подмножество ориентированного графа памяти переводов, состоящее из узлов, которые можно достичь, начав обход от узла, соответствующего выбранному слову.

Используя особенности выбранной структуры памяти переводов, задачу поиска сегментов, похожих на заданный, можно решить путем выполнения следующих действий (рис. 14):

1. разбить заданный сегмент на слова;
2. найти в памяти переводов все узлы, соответствующие этим словам;
3. спускаясь по графу отношений наследования, помещать в список найденных сегментов все встречаемые узлы.

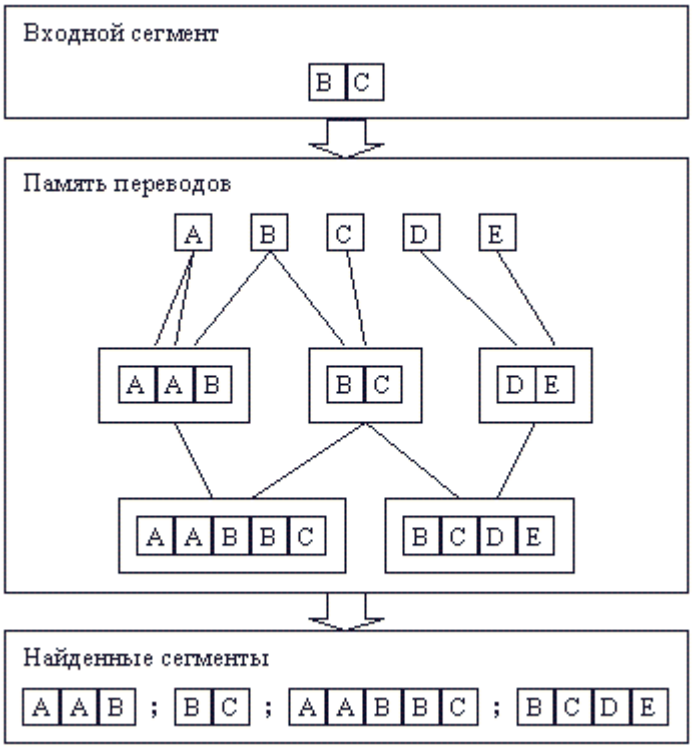


Рис. 14

Резонным представляется вопрос о том, в каком порядке следует предоставлять найденные сегменты переводчику: ведь приведенная процедура поиска выберет из памяти все сегменты, пересекающиеся с заданным по крайней мере по одному слову. Каковы правила фильтрации и сортировки найденных сегментов?

Ответ на этот вопрос лежит за пределами выбранного формализма, однако в этом нет ничего страшного. Дело в том, что результат поиска представляет собой классический вариант одноуровневой памяти переводов, анализ которого может быть произведена методами, формализованными в рамках существующих сред перевода. Для обеспечения эффективности поиска целесообразно осуществлять оценку "пригодности" сегментов по мере их нахождения. Например, если некоторый сегмент полностью совпадает с эталоном, то все его потомки в графе могут быть автоматически исключены из поиска.

Теперь поговорим о задаче добавления нового сегмента в память переводов. Очевидным условием корректности процедуры добавления является обеспечение успешного поиска. Стало быть, добавляемый сегмент должен иметь в числе своих предков (не обязательно прямых) все составляющие его слова. Следуя целям оптимальности, можно заключить, что среди предков должны присутствовать также узлы графа, содержащие фрагменты данного сегмента. Иными словами, если в памяти переводов присутствуют сегменты "AB" и "CD", то сегмент "ABCD" должен стать наследником этих двух сегментов. Аналогично, если в памяти присутствует сегмент "ABCD", то добавляемый сегмент "AB" должен стать его предком. В общем случае при добавлении сегмента в граф памяти переводов могут существовать альтернативные варианты наследования. В такой ситуации схема добавления заметно усложнится. В любом случае, проблема построения оптимальной иерархии классов решается в рамках объектно-

ориентированного подхода, поэтому мы не будем заострять здесь на ней внимание.

3. Вычисление пересечения языковых пар

Поскольку выделение общей части двух сегментов- важный этап предлагаемой технологии перевода, изучим этот вопрос более детально. При этом, помня о том, что сегмент в памяти перевода выступает, чаще всего, не отдельно, а как элемент языковой пары, будем рассматривать именно пересечение пар.

Для начала дадим определения пересечения сегментов. Итак, пересечение сегментов A и B- это множество сегментов C_i, таких что:

1. каждый из C_i содержится и в A, и в B;
2. никакие два C_i не содержат одинаковых фрагментов; не существует такого сегмента D, что и A, и B содержат D, и D содержит один из сегментов C_i.

Приведенное определение не подразумевает выделения из сегментов A и B всех общих фрагментов. Это сделано для того, чтобы можно было использовать алгоритмы различной сложности реализации пересечения.

Теперь перейдем к пересечению языковых пар. Как уже было упомянуто выше, очень важно определить, является ли пересечение изоморфным, иными словами, можно ли считать результаты пересечения исходных и целевых сегментов языковой парой. Два примера иллюстрируют это (рис. 15). В первом случае пару сегментов "достаточно высока" и "ishighenough" имеет смысл поместить в память переводов, поскольку она действительно представляет собой вариант перевода, который может быть повторно использован переводчиком. Во втором случае- это совершенно очевидно- сегменту "достаточно" не следует сопоставлять сегмент "ishighenough", поскольку данная языковая пара будет некорректной.

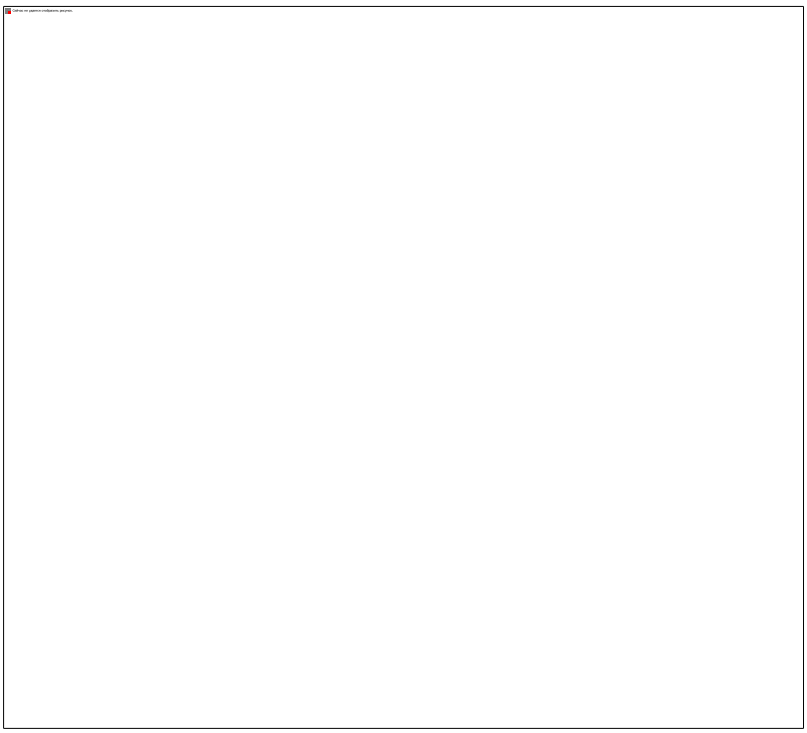


Рис. 15

Для проверки изоморфизма пересечений можно использовать подход, основанный на технологии машинного перевода. Его суть в сопоставлении терминов, образующих исходный и целевой сегменты. Для этого необходимо произвести грамматический разбор сегментов с целью выделения терминов и синтаксических связей между ними. После этого можно воспользоваться терминологическим словарем для определения того, какому термину в целевом сегменте соответствует заданный термин в исходном сегменте. Иными словами, изоморфизм можно определить по следующему критерию (рис. 16):

пересечение является изоморфным, если всем терминам его исходного сегмента, сопоставлены термины его целевого сегмента, и синтаксические связи между ними идентичны тем, которые присутствуют в сегментах, из которых было получено пересечение.

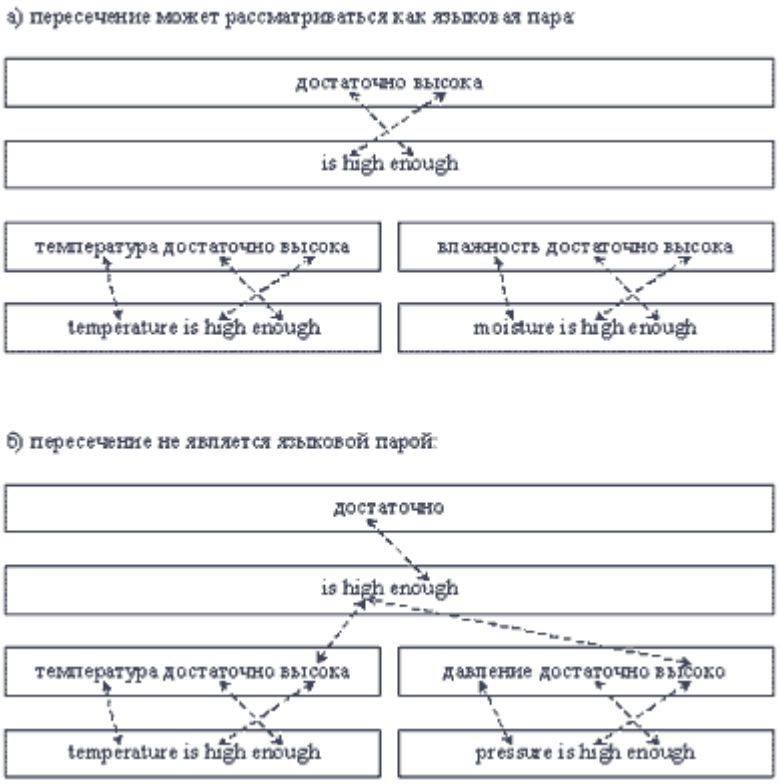


Рис. 16

В общем случае, для оценки изоморфизма можно проверять не только отдельные термины (суть корневые узлы графа памяти переводов), но и родительские сегменты всех уровней. Это повысит надежность оценки, снизив риск неправильного определения синтаксических связей.

Следует обратить внимание на тот факт, что в предлагаемой модели машинный перевод используется только для грамматического анализа текста, образующего сегмент. Слабым местом систем машинного перевода является выбор перевода для терминов сегмента, и именно эта задача решается более надежным способом- с помощью памяти переводов.

От языковых пар к языковым звездам

Нередкой является ситуация, когда перевод приходится осуществлять не только с языка А на язык В, но и, наоборот, с языка В на язык А. Одна и та же память переводов будет одинаково полезна в обоих случаях, поскольку содержит максимально синхронизированные графы сегментов на языке А и на языке В. Однако стоит нам усложнить задачу и предположить

необходимость перевода между несколькими языками, как полезность единой памяти переводов заметно падает. Действительно, если перевод осуществлялся с языка A на языки B и C, то в памяти не будет храниться соответствия между сегментами на языках B и C. Как же обеспечить подобную возможность?

Разумным решением могло бы явиться использование некоторого промежуточного языка X, на который осуществлялся бы перевод, а затем, вторым этапом, выполнялся бы перевод с языка X на целевой язык. В подобном случае все языковые пары в памяти переводов состояли бы из сегмента языка X и сегмента одного из целевых (либо исходных) языков. Тут имеются, однако, подводные камни. Во-первых, как мы уже убедились, пересечение языковых пар не всегда бывает изоморфным, следовательно, не все языковые пары в памяти переводов будут содержать перевод на язык X. Очевидно, такие пары будут бесполезны. Во-вторых, при переводе всегда имеется опасность потери смысла: двойной перевод значительно увеличивает эту опасность.

Каким же должен быть этот гипотетический промежуточный язык X, чтобы им было целесообразно воспользоваться? Его свойства вытекают из двух названных проблем. Во-первых, этот язык должен обеспечивать изоморфное пересечение для любого другого языка. Нарушение изоморфизма (по крайней мере, в родственных языках) обусловлено в значительной степени различием синтаксических правил, приводящим к разному порядку членов предложения, а также к различию форм одного и того же слова. Отсюда следует, что язык X должен быть инвариантен к порядку слов и как-то учитывать их формы в исходном языке. Во-вторых, он должен быть в состоянии передать смысл фразы на любом языке, следовательно- включать в себя специфические понятия всех существующих человеческих языков.

Если такой универсальный язык будет найден, то память переводов можно будет организовать не на основе языковых пар, а на основе языковых

звезд, где в центре находится сегмент на языке X, на лучах- варианты переводов его на другие языки. При значительном объеме перевода между большим количеством языков дополнительные затраты на удвоенную работу переводчика с лихвой окупятся гибким механизмом памяти переводов, значительно упрощающим многоязычный перевод.

Осталось только найти язык X. И такой язык существует! Это универсальный сетевой язык UNL (UniversalNetworkingLanguage), предложенный Институтом Развития Обучения (Institute of Advanced Studies-IAS) при Университете Объединенных Наций (United Nations University-UNU).

Тема 10. Понимание речи.

1.Теоретические предпосылки.

2.Акустико-фонетический анализ.

3.Фонологический анализ.

4.Структура системы понимания речи. Стратегии обработки.

1. Теоретические предпосылки.

Понимание речи обычно трактуют как преобразование акустического представления речи в смысловое. При создании практических систем смысл можно определить, как представление, из которого извлекаются действия, совершенные системой.

Понимание речи следует отличать от распознавания речи, где целью является сопоставить речевое высказывание с соответствующими словами в словаре. До начала 70-ых большинство исследований было направлено на распознавание речи. 5 лет потребовалось на создание системы, первоначальная исследовательская цель которой заключалась в распознавании речи, а конечные результаты в понимании. Казалось, что способность системы давать разумный ответ на речь была более значимым критерием для развития речевых систем. К тому же считалось, что речевой

сигнал является недостаточным источником информации, и знание контекста речевого высказывания важно только для успешного распознавания и интерпретации. Системы по распознаванию речи, основанные на динамическом программировании и соответствии с образцами, развивали для речевых высказываний, которые состояли почти полностью из изолированных слов, выбираемых из небольшого вокабуляра.

Однако такой подход, при котором ищется наиболее точное соответствие между определенными произнесенными словами и вокабуляром акустических образцов слов, меньше всего подходил к связанной речи, так как входной акустической сигнал в этом случае не может быть эффективно смоделирован, как простое сочетание произнесенных частей лексических единиц. В связанной речи изменчивость, выявляемая при соответствии с образцами, передает полезную информацию и для распознавания, и для интерпретации.

Однако, необходимо начинать с основных лингвистических единиц, таких как фонемы, и сохранять информацию о ритме и длительности речевого высказывания. Если следуют таким путем, то подход к обработке речи, основанный скорее на знании, чем на соответствиях с образцами, становится неизбежным, так как, чтобы извлекать преимущества из распознавания конкретных лингвистических единиц в сигнале, необходимо знать, как данная единица связана с остальной частью языка.

Системы понимания речи (СПР) имеют дело со связанными единицами речи, такими как, фразы, предложения и даже параграфы, так как "понимание" изолированных слов может означать только тривиальный процесс сопоставления некоторого значения к каждому слову словаря системы. Понимание связанной речи - очень сложная задача, и на проект СПР повлияли исследования в таких разных областях, как акустическая обработка сигнала, нейрофизиология, психолингвистика, психология. СПР была создана, чтобы понимать всего нескольких дикторов одного диалекта, производя грамматически ограниченное подмножество языка со словарем

около тысячи слов. Сейчас хотя и имеются много потенциальных прикладных программ для СПР их эффективность и надежность все еще недостаточна, чтобы широко использоваться.

Системы, зависимые от диктора, распознающие изолированные слова с небольшим словарем, использующие в качестве образцов-соответствий целые слова уже нашли свое применение, типа обработки багажа на авиалиниях. Тем не менее, признано, что усовершенствование такого типа систем (большие словари, независимость от диктора) требует подхода, основанного на более глубоких знаниях.

Посредником при преобразовании речи в ее значение должны служить определенные компоненты, которые используют разнообразные источники знания (ИЗ), т.к. речевой сигнал кодирует много различной информации, необходимой для восстановления значения. Например, вариативность в произношении слов в связанной речи больше не является помехой при подборе образца соответствия, но это довольно важный источник информации, например, относительно расположения границ слова или контекстуально важной (выделенной ударением) информации в произнесении. Единственной возможной организацией СПР и основных ИЗ является следующая: РЕЧЬ - ОБРАБОТКА АКУСТИЧЕСКОГО СИГНАЛА - ФОНЕТИЧЕСКИЙ АНАЛИЗ - ФОНОЛОГИЧЕСКИЙ АНАЛИЗ - МОРФОЛОГИЧЕСКИЙ АНАЛИЗ - ЛЕКСИЧЕСКИЙ ДОСТУП К СЛОВАРЮ - СИНТАКСИЧЕСКИЙ АНАЛИЗ - СЕМАНТИЧЕСКИЙ АНАЛИЗ - ЗНАЧЕНИЕ. При такой организации СПР информация течет вверх по мере того, как каждый элемент создает промежуточные представления, кодируя (частичные) гипотезы относительно ввода на основе ему доступного знания.

Акустическая обработка отцифровывает сигнал с входной частотой, которая сохраняет сигнал для понимания. Акустическая обработка также трансформирует оцифрованный сигнал различными способами, чтобы представить его в той форме, которая поддается фонетическому

декодированию. Например, спектральный анализ будет выполнен для каждого проанализированного фрейма, и дополнительные параметры, такие как частота основного тона, подсчитаны. Параметрический сигнал может затем быть помечен как дискретная последовательность фонем. Например, если сигнал с низкой амплитудой равномерно распространяется поперек спектра, то этот звук вероятно фрикативный, типа [f] или [v]. Кроме того, для каждой фонемы характерны такие особенности, как высота тона, длительность и амплитуда. Акустическо - фонетическое преобразование является решающим для эффективной работы СПР, но все еще одно из наиболее слабых сторон речевой обработки. И это являлось главным недостатком СПР.

Фонологический анализ выполняется на фонетическом представлении, которое определяет лингвистически важные различия, имеющиеся в фонетическом представлении произнесения, например, уровни и расположение ударения, интонационный контур, структуры слога, последовательности фонем, лежащих в основе произнесения. Фонологический анализ необходим для лексического доступа, т.е. процесса, который сопоставляет фонетическую форму произнесения с каноническими фонемными представлениями слов в словаре, чтобы восстановить информацию, хранящуюся там относительно их морфологических, синтаксических, и семантических свойств. Это отменяет такие эффекты быстрой речи, как ассимиляция или сокращения. Например, слова “did” и "you" могли бы иметь в словаре следующие последовательности фонем: /dld/ и /ju:/. Однако, акустическо - фонетическое преобразование могло бы восстанавливать фактические звуки или фонемы, типа [dIje]; связывать эту фонетическую последовательность с каноническими фонемными представлениями “did” и "you". Это необходимо, если нужно узнать, что палатализация произошла на границе слова, заменив [dj] на [j], и что неударный гласный "you" был редуцирован до нейтрального безударного.

Аналогично, фонологическое знание относительно допустимых последовательностей фонем в слогах может использоваться, чтобы распознать слог, и следовательно, границы слова. Например, в /houmhelp/ должна быть граница между /m/ и вторым /h/, потому что никакой слог в английском не может содержать /mh/. Как только фонологический анализ завершен, дальнейшая обработка ввода будет подобна пониманию текста. Дальнейшие морфологический, синтаксический, семантический и прагматический анализы способствуют распознаванию, эксплуатируя избыточность речи, в информационно - теоретическом смысле. В некоторых из проектов APRA задача синтаксического анализа заключалась в том, чтобы исключить гипотезы слова на основе синтаксически недопустимых последовательностей. Прежде, чем слова, выделенные в речевом сигнале будут сопоставлены с лексическими входам в словаре системы, необходимо провести морфологический анализ, который приведет слова к их основной форме, например, устранит окончание множественного числа /s/ или /z/, которые сильно бы расширили число входов в словарь.

После морфологического анализа возникшее морфофонологическое представление речевого ввода может быть найдено в словаре системы, чтобы получить синтаксическую и семантическую информацию относительно гипотезы последовательности слов. Синтаксический, семантический, и прагматический анализ - в основном тот же самый для речевого и текстового понимания. Однако, должно быть взаимодействие между этими и более низкими уровнями анализа не только, потому что они будут дополнять правильное распознавание произнесения, но также потому что некоторые аспекты фонологического анализа, особенно касающиеся ударения и интонации, будут способствовать интерпретации. Ударение, например, необходимо для определения контекстуально-новой информации и для нахождению зависимых слов для местоимений. ИЗ, использованные в понимании речи, являются прежде всего лингвистическими. Эффективность

СПР зависит во много как от эффективного использования этих ИЗ так и от разработки их содержания.

2. Акустическо - фонетический анализ.

Наиболее важная область в обработке речи, нуждающаяся в исследованиях - это акустическо - фонетический анализ. Если акустическо - фонетический анализ слабый, то ошибочные гипотезы выдадут в итоге неправильный анализ. Сегментация и идентификация акустического сигнала в последовательности лингвистических единиц чрезвычайно трудна. Сначала, речь - это код, а не шифр; то есть, акустическое сигналы, ассоциирующиеся с сегментами, непосредственно с ними не связанны; на эти сигналы сильно влияют соседние сегменты. Например, спектрограммы /d/ в /di/ и /du/ очень различны, т.к. на них влияют последующий гласный. Кроме того, не возможно разделить акустической сигнал на /d/ и следующий гласный. Эти наблюдения создали следующую теорию: конечное количество этих сегментов не всегда можно достичь из-за непрерывного движения вокального трактата. Такой синтезирующий анализ был бы, однако, очень в вычислительном отношении дорогой, так как он требовал бы, чтобы СПР умел генерировать всех возможные произнесения и сопоставлять их с акустическом вводом. Однако, во-первых, акустическое сигналы, в противоположность фонемам или алафонам, содержат инвариантные сигналы. Во-вторых, акустическое сигналы часто сильно редуцируются в безударном положении. Это часто вызывает много неправильных гипотез в системах, где акустическо - фонетический компонент будет принимать за гипотезу сегмент из фиксированного инвентаря. В-третьих, акустическое сигналы варьируют от диктора диктору из-за физиологических особенностей вокального тракта, различия в характеристиках речи и т.д. Люди способны компенсировать эти различия быстро и плавно, но все еще мало понятно, как сделать этот процесс автоматическим. Большинство коммерческих систем распознавания речи требует длинного обучения, повторяя за пользователем каждое слово в словаре системы несколько раз и - следовательно очень

зависимо диктора. Несколько из разработанных СПР достигли определенной степени независимости от диктора, пытаясь ввести параметр в акустическо - фонетический анализ для нового диктора на основе обучающегося предложения, которое знала система, пользователю же следовало его проговорить.

Проектируют СПР, где акустическо - фонетический анализ фактически не существовал и сегментный анализ не был точным. Конечное представление каждой системы было главным образом определено эффективностью более высоких уровней анализа при исправлении ошибок на фонетическом уровне. Более современные системы используют более сложный акустическо - фонетический анализ, интегрируя информацию из ряда преобразований акустического сигнала и создавая несколько типов фонетических представлений, но эффективность все еще ограничивается в среднем 70% успешным распознаванием фонем из речевого высказывания, произнесенных небольшим количеством дикторов.

3. Фонологический анализ.

Фонологический компонент необходим для любой, обрабатывающей речь, системы, основанной на знаниях, потому что система требует знания относительно фонологических процессов, активных в языке и в прикладных программах, чтобы восстанавливать канонические произношение слов, которые могут быть сопоставлены с соответствующими входами словаря, и получать дальнейшие сигналы к синтаксической и семантической/прагматической интерпретации речевого высказывания. Фонологические компоненты были разработаны для СПР. Однако, они были в значительной степени ограничены лексическими, сегментными процессами и обычно имели дело с фонологически управляемыми изменениями, генерируя альтернативное произношение для индивидуальных лексических единиц и сохраняя их в дополнительном словаре. Этот подход не может иметь дело адекватно с фонологическими процессами, которые соединяют границы слова, типа палатализации. Самая большая область прикладной

программы для фонологического правила - интонационная фраза; следовательно, фонологию нельзя рассматривать в терминах различного произношения для лексических единиц. Фонологический анализ обеспечивает много важной информации для СПР; например, различные виды фонологического правила блокированы различными лингвистическими границами между сегментами.

Полезно разложить на слоги и слова речь, сегментация может также обеспечить сведения для синтаксического анализа; палатализация соединяет границы слова, но блокирована на границах главных синтаксических составляющих, так что ее отсутствие может использоваться, чтобы решить неоднозначность относительно присутствия такой границы в данном месте речевого сигнала. Фонологические правила также изменяются среди диалектов. Следовательно, СПР, способные к пониманию дикторов с различными диалектами, требовали бы знания относительно этих различий и способности реконфигурировать себя для их речи. Палатализация, например, происходит чаще в американских диалектах, чем в британских или английских.

В конце семидесятых стали развиваться новые подходы к фонологии, такие как автосегментная, метрическая зависимости, фонология зависимости, для которых центральным является сверхсегментальный аспект. Некоторые из этих достижений были включены в СПР.

ИЗ бесполезны в СПР, если знание, которое они кодируют, не может быть представлено таким образом, который позволяет интерпретацию с помощью машины. Например, специалисты по фонетики обычно используют

Международный Фонетический Алфавит для фонетической записи. Однако, так как выбор представления воздействует на прикладную программу знания, системы представления ИЗ в СПР часто являлись компромиссом между описательной адекватностью и вычислительной эффективностью. Например, в ARPA проектируют каждый СПР, используя идею синтаксического представления, чтобы не выражать все

грамматические возможности английского языка. Формальный язык и теория автоматов предлагают эффективные алгоритмы для прикладной программы ИЗ, выраженные в наборах правил с соответствующими формальными свойствами. Например, минимально увеличенные контекстно - свободные записи для адекватного описания английского синтаксиса и фонологии. Однако, успехи этого вида не ведут автоматически в вычислительном отношении к ИЗ, так как наборы правил, требуемые, чтобы выразить знание в этой форме могут быть чрезвычайно большие. Кроме того, кажется маловероятно, что все ИЗ, используемые в СПР могут быть выражены внутри таких ограниченных записей. Тем не менее, более специализированные и мощные методы также были разработаны, типа интерпретаторов для промышленных систем или увеличенные сети переходов. Появляются некоторые экспертные оболочки системы, являющееся многообещающими прикладными программами для акустическо - фонетического преобразования. Чем лучше понимание специфической области, тем больше возможность представления знания адекватно и эффективно. Кроме того, вероятно, что различные схемы представления будут наиболее эффективны для различных ИЗ; следовательно, структура СПР, которая навязывает, одинаковую схему для всех ИЗ не идеальна. На выбор представления воздействуют факторы, другие чем доступность методики интерпретации для специфической схемы; например, несколько СПР не пытаются отображать непосредственно между акустическом сигналом и фонетическим алфавитом, но создавать промежуточные представления, отмечая акустическо яркие особенности типа назальности, помогать процессу распознавания фонем. На представления также воздействует порядок, в котором расположены различные ИЗ, относящиеся к речевому сигналу и полной структуре СПР. Недавно было предложено, чтобы начальный фонетический анализ отмечал согласные, гласные, а также ударные и безударные слоги и что это простое представление должно использоваться, чтобы получить набор слов-кандидатов из соответственно организованного словаря.

Детализированный фонетический анализ затем применялся бы к безударному слогу (слогам), чтобы распознать его между кандидатами.

4.Структура системы понимания речи. Стратегии обработки.

Проблема межкомпонентной связи во время обработки является основной, т.к. неоднозначности должны быть решены быстро, чтобы избежать ненужного вычисления, и также потому, что избыточность между ИЗ может использоваться, чтобы разложить на множители неправильные гипотезы, вызванные или ошибками системы или подлинной неоднозначностью в речевом сигнале. Например, акустическо - фонетический компонент мог бы предложить аспирированный /p/ или /b/, за которым следует гласные и /t/, результатом этого предположения могут стать такие слова-кандидаты, как “put” и "but". Однако, вероятно, одно из них будет отклонено на основе синтаксического анализа, так как глаголы и союзы не играют одинаковую роль в предложении. Аналогично, подлинная синтаксическая неоднозначность имеется в высказывании, типа " He gave her dog biscuits ", где сочетание "her” может функционировать и как прилагательное и как существительное. Но в этом случае неоднозначность может быть решена с помощью ударения и интонации, которые будут сопровождать обе интерпретации. Предложенные структуры - иерархические, с последовательным потоком информации через цепочку компонентов ИЗ, и неиерархические, без ограничения на поток информации между компонентами.

Преимущество иерархического подхода в том, что имеется естественный порядок для прикладной программы ИЗ, чтобы вводить речь; синтаксический анализ может осуществляться только на основе лексической информации и т.д. Кроме того, в целом управление системы просто. Однако, имеются много случаев, когда непоследовательные взаимодействия между цепочкой компонентов полезны; например, аспекты просодической, сверхсегментальной структуры высказывания будут релевантны по

отношению к фонологической, синтаксической, семантической, и прагматической интерпретации. Непоследовательное взаимодействие может быть достигнуто внутри иерархической модели, передавая все возможные анализы, совместимые с данным компонентом следующему, который затем выбирает подмножество анализов. Но это только тогда сработает, если промежуточные представления, переданные через СПР настолько обогащены, что можно было бы использовать всю проанализированную информацию в следующих компонентах. Таким образом, ввод синтаксического компонента в дополнение к синтаксической информации относительно слов должен включить всю доступную информацию для синтаксического анализа, типа просодической информации, и вся информация, относящаяся семантическому/прагматическому анализу должна быть также включена. Это усложняет схему представления, и дорого в вычислительном отношении, т.к. создает много неправильных гипотез. Неправильных гипотез можно избежать, т.к. информация, в которой отсутствует неоднозначность временно доступна, она закодирована в той части речевого сигнала, который уже проанализирован на более низких уровнях, но в иерархической модели этот способ не применяется, пока ввод не достигает соответствующего компонента в последовательной цепочке. Неиерархические системы избегают неэффективности, позволяя компонентам применять в наиболее эффективном порядке сложные межкомпонентные связи. Каждый компонент нужно обеспечить средствами, чтобы запрашивать и получить информацию из других компонентов или начинать определенную обработку в другом компоненте. Это требует специальных каналов связи между компонентами в системе. Разработка адекватной системы управления для такой модели невозможна, т.к. должна предусматривать все возможные потоки управления в стадии проекта. Практически, реальные неиерархические модели для СПР были ограничены однородными представлениями из ИЗ и одиночной глобальной структурой данных, как в (blackboard systems) рабочих системах.

Стратегии Обработки. Различные стратегии обработки использовались в разных структурах СПР, чтобы сократить вычисление, требуемое для успешного анализа. И иерархические и неиерархические системы могут работать со способами управления данными как снизу-вверх, так и сверху-вниз при использовании знания, чтобы создать гипотезы относительно ввода. Однако, самые современные СПР используют способ снизу-вверх из-за довольно слабого предсказания речи на основе ИЗ. Аналогично, СПР может исследовать пространство, определяя его глубину и ширину. Большинство систем оперирует с шириной пространства из-за сомнительного или ошибочного характера многих гипотез, но использует подсчитывающие методы, чтобы сохранить размер активного исследуемого пространства. Одна из таких методик, подсчитывающая неудачи, которая включает измерение совокупности множества индивидуальных слов-кандидатов в соотношении с теоретической верхней границей и обработку гипотезы, гарантирует, что СПР найдет наиболее полную подсчитывающую гипотезу для первого высказывания. Однако это не гарантирует, что наиболее привлекательная гипотеза является правильной; эффективность компонентов, которые способствуют порождению гипотез слова, все еще является определяющим фактором в полном представлении системы. Этим оценкам должны отвечать все компоненты, и они должны отражать различные добавления каждого ИЗ. Однако, значение, которое должно быть присоединено к любому ИЗ, должно измениться в соответствии с контекстом. Например, при распознавании безударного и фонетически редуцированного предлога, синтаксический анализ должен чаще обращаться к акустическому анализу, чем при распознавании ударного слога. Кроме того, исследования должны быть оценены с помощью времени. Хотя некоторые схемы оценки, которые использовались в готовых СПР, улучшают эффективность, это связано или по теоретическим причинам, с подсчитывающей методикой.

Анализ речевого сигнала может проходить слева направо через линейный сигнал или из середины островов большей акустической надежности в обоих направлениях. Подход, использующий острова надежности, имеет преимущество в принятии свободных от ошибок фонетических данных за начальную отметку за счет более сложной структуры управления и организации системы. По-видимому слушатели обращают большее внимание на ударные слоги, которые вообще более ясно произносятся, и следовательно более легко анализируются фонетически. Кроме того, фонологическая структура английского словаря вынуждена быть составленной таким способом, при котором каждое слово может быть получено даже при грубом фонетическом анализе структуры слога вместе с детальным анализом ударного слога. Следовательно, подход, использующий острова надежности по существу правилен, хотя и был бы более эффективен, если обработка началась в ударных слогах.

Тема 11. Распознавание речи.

План

1.Основные задачи технологии распознавания речи.

2. Акустическая структура сигнала.

3. Уровни артикулярного контроля.

1. Основные задачи технологии распознавания речи.

Существующие технологии распознавания речи не имеют пока достаточных возможностей для их широкого использования, но на данном этапе исследований проводится интенсивный поиск возможностей употребления коротких многозначных слов (процедур) для облегчения понимания. Распознавание речи в настоящее время нашло реальное применение в жизни, пожалуй, только в тех случаях, когда используемый словарь сокращен до 10 знаков, например при обработке номеров кредитных

карт и прочих кодов доступа в базирующихся на компьютерах системах, обрабатывающих передаваемые по телефону данные. Так что насущная задача - распознавание по крайней мере 20 тысяч слов естественного языка - остается пока недостижимой. Эти возможности пока недоступны для широкого коммерческого использования. Однако ряд компаний своими силами пытается использовать уже существующие в данной области науки знания.

Для успешного распознавания речи следует решить следующие задачи:

- обработку словаря (фонемный состав),
- обработку синтаксиса,
- сокращение речи (включая возможное использование жестких сценариев),
- выбор диктора (включая возраст, пол, родной язык и диалект),
- тренировку дикторов,
- выбор особенного вида микрофона (принимая во внимание направленность и местоположение микрофона),
- условия работы системы и получения результата с указанием ошибок.

Существующие сегодня системы распознавания речи основываются на сборе всей доступной (порой даже избыточной) информации, необходимой для распознавания слов. Исследователи считают, что таким образом задача распознавания образца речи, основанная на качестве сигнала, подверженного изменениям, будет достаточной для распознавания, но тем не менее в настоящее время даже при распознавании небольших сообщений нормальной речи, пока невозможно после получения разнообразных реальных сигналов осуществить прямую трансформацию в лингвистические символы, что является желаемым результатом.

2. Акустическая структура сигнала.

Вместо этого проводится процесс, первым шагом которого является первоначальное трансформирование вводимой информации для сокращения обрабатываемого объема так, чтобы ее можно было бы подвергнуть компьютерному анализу. Примером является «техника сопоставления отрезков», позволяющая сократить вводимую информацию с 50'000 до 800 битов в секунду. Следующим этапом является спектральное представление речи, получившееся путем преобразования Фурье. Результат преобразования Фурье позволяет не только сжать информацию, но и дает возможность сконцентрироваться на важных аспектах речи, которые интенсивно изучались в сфере экспериментальной фонетики. Спектральное представление достигнуто путем использования широко-частотного анализа записи. Хотя спектральное представление речи очень полезно, необходимо помнить, что изучаемый сигнал весьма разнообразен. Разнообразие возникает по многим причинам, включая:

- различия человеческих голосов;
- уровень речи говорящего;
- вариации в произношении;
- нормальное варьирование движения артикуляторов (языка, губ, челюсти, нёба).

Для устранения негативного эффекта влияния варьирования голосового тракта на процесс распознавания речи было использовано множество методов. Первым делом рассматривалась характеристика пространства траектории артикуляторных органов, включая гласные, используемые говорящим. Наиболее удачные формы трансформации, использованной для сокращения различий, были впервые представлены Сакоя & Чибо и назывались динамичными искажениями (dynamic time warping). Техника динамичного искажения используется для временного вытягивания и сокращения расстояния между искаженным спектральным представлением и шаблоном для говорящего. Использование данной техники дало улучшении точного распознавания (~20-30%). Метод динамичного искажения

используют практически все коммерчески доступные системы распознавания, показывающие высокую точность сообщения при использовании. Вначале сигнал преобразовывается в спектральное представление, где определяется немногочисленный, но высокоинформативный набор параметров. Затем определяются конечные выходные параметры для варьирования голоса(следует отметить, что данная задача не является тривиальной) и производится нормализация для составления шкалы параметров, а также для определения ситуационного уровня речи. Вышеописанные измененные параметры используются затем для создания шаблона. Шаблон включается в словарь, который характеризует произнесение звуков при передаче информации говорящим, использующим эту систему. Далее в процессе распознавания новых речевых образцов (уже подвергшихся нормализации и получивших свои параметры), эти образцы сравниваются с шаблонами, уже имеющимися в словаре, используя динамичное искажение и похожие метрические измерения. В настоящее время этот метод изучается и дополняется.

Очевидно, что спектральное представление речи позволяет характеризовать особенности голосового тракта человека и способ использования его говорящим. Самый обычный способ моделирования специфических эффектов "модель-источник" - использование фильтров. Речевой аппарат моделируется с использованием источников, вызывающих резонанс, ведущий к пиковым точкам интенсивности звука в соседстве с отдельными частотами, называемыми формантами. При произнесении звуков вибрация голосовых связок является источником возбуждения, и эти короткие импульсы вызывают резонанс между голосовыми связками и губами. Так как язык, челюсть, губы, зубы и альвеолярный аппарат двигаются, размер и место этих резонансов меняются, давая возможность воспроизведения особых параметров звуков.

Возможно построить очень точную модель, также прямо смоделировать движения артикуляторов физиологически реальным путем.

Использование этих моделей привели к пониманию пути, в котором происходит речевой сигнал. Но так как наблюдение над артикуляторами затруднено, остаются недостатки. Хотя природа вокального тракта очень сильно влияет на выходной сигнал речи, это не единственное ограничение, которое необходимо принимать во внимание, так как контроль над мускулами звукового тракта обусловлен сигналами моторного кортэкса мозга. Возможно все аспекты влияния акустической структуры контролируют сигналы и форму звукового выхода речи (хотя это не может быть доказано с систематической точки зрения).

Аспекты влияния акустической структуры включает в себя:

- природу сегментов индивидуального звука (гласные/согласные),
- структуру слога,
- структуру морфем (приставки, корни, суффиксы),
- лексикон,
- уровень синтаксиса фраз и предложений и
- долгосрочные ограничения речи (long-term discourse constraints) .

Необходимо также принять во внимание тот факт, что человеческий аппарат восприятия также должен быть смоделирован, он сам по себе накладывает на процесс восприятия дополнительные ограничения. Недавно процесс восприятия был изучен с помощью метода сигнального подавления барабанных перепонок через возбуждение нервных клеток, которые образовывают примерно 30 тысяч нервных окончаний слухового нерва. Но изучение нервных окончаний способно только прояснить формирование простых синтетических гласных. Перед исследователями встало новое главное направление в области изучения воспроизводства речи, связанное с интеграцией всей физиологии восприятия человека. В настоящий момент

появляются некоторые модели явлений, происходящих в ухе, и не без оснований можно ожидать дальнейшего улучшения понимания процесса распознавания речи из-за более полного понимания характеристик этого влияния.

3. Уровни артикулярного контроля.

Что касается уровня артикуляторного контроля, первым уровнем является индивидуальный фонетический сегмент, иначе говоря, - фонема. Во многих естественных языках их примерно 40. Но их набор существенно различается. Поэтому, например, английские гласные могут быть носовыми, даже ненамеренно, в то время как во французском носализация гласных является фонетическим контрастом, и поэтому влияют на значение произносимого. Во французском языке носовая коартикуляция доминирует в гласных и существенно влияет на восприятие фонем и следовательно на главный смысл значения. Хотя все говорящие имеют одинаковый голосовой аппарат, использование его разное. Так например, использование кончика языка или прищелкивание, как в некоторых африканских языках. Ясно, что природа артикуляционных движений имеет сильное влияние на метод воспроизведения речи. Эти ограничения всегда активно используются в практических системах.

На следующем уровне лингвистической структуры фонетические сегменты сгруппированы в согласные/гласные, а следовательно и в слоги. Далее, в зависимости от роли фонетического сегмента внутри этих слогов их реализация может быть сильно изменена. Так например, начальный согласный в слоге может быть реализован как абсолютно отличный от конечной позиции. Согласные очень крепко связываются между собой, что опять же влияет на последующие ограничения. Говорящие на родном языке избегают этих ограничений или могут активно их использовать во время процесса восприятия. Любое моделирование процесса восприятия может быть активным и может оказать большую помощь в понимании главного смысла.

Пример, показывающий необходимость применения сфокусированного поиска, может быть представлен в восприятии конечного согласного. Среди многих ключевых слов для распознавания конечного согласного существует спектральная природа шума, воспроизводимого при освобождении конечной перемычки и перехода резонанса второй форманты в гласный, следующий за этой перемычкой. Многие исследователи изучали эти влияния, и результаты их исследований показали, что ограничивающее влияние обоих вышеописанных характеристик на восприятие варьируется природой следующего гласного, и следовательно, мощная стратегия распознавания должна иметь некоторые знания о твердой позиции гласного перед конечным согласным перед тем, как будет сделано само распознавание конечного согласного. Конечные согласные дают яркий пример весьма интересного комплекса фонетики, используемого для лингвистической окраски. Например, при рассмотрении слов rapid и rabid обнаруживается 16 фонетический различий.

Кроме сегментного и слогового уровней существуют ограниченные влияния из-за структуры морфем, которые являются минимальными синтаксическими единицами языка. Они включают в себя приставки, корни, суффиксы. Можно себе представить, что это синтаксис на слоговом и на морфемном уровнях, также как и нормально распознанный синтаксис, характеризующийся способом, в котором английские слова объединяются во фразы и предложения. Возможно представить данные ограничения как последствия рассмотрения грамматики вне контекста. В этом виде ограничений много “шумных” вариаций сегментов речи, которые так же относятся и к иерархическим синтаксическим ограничениям.

Дополнительные ограничения на природе входа новой лексики в язык могут являться уровнем слова. Многие исследования обнаружили, что характеристика слов при введении разбиения на 5 жестких классов фонетических сегментов может быть сокращена до минимума, часто имея единственное в своем роде распознавание. Далее слишком усиливается

эффект порядка двух букв и фонетических сегментов с тех пор как в изучении английских и французских словарей было обнаружено, что более 90% слов имели единственное значение и только 0,5% имели 2 и больше альтернатив. На фонемном уровне было обнаружено, что все слова в английском словаре из 20 тысяч слов имели одно значение из-за беспорядочных фонемных пар

Кроме уровня слов синтаксис имеет дополнительное ограничительное влияние. Его влияние на последовательный порядок слов часто характеризуется в системах фактором, который в свою очередь характеризует количество возможных слов, которые могут следовать за предыдущим словом в процессе произнесения. Синтаксис также имеет ограничительные влияния на просодические элементы, такие как ударение. Возможно для того, чтобы охарактеризовать ударение в слове, нужно принять во внимание не только индивидуальное слово, но вышеприведенные дополнительные ограничения синтаксиса.

Несмотря на сложность описания характеристик источников различных ограничений, немаловажную роль играют современные системы влияния, которые представлены всеми возможными вариантами произнесения звуков. Например, система HARPI университета Carnegie-Mellon University является системой, в которой звуковоспроизведение описывается как путь через комплексную сеть. В этом способе ограничения структуры слога, слова и синтаксиса связаны одной структурой. Структура контроля, используемая для поиска, является адаптацией динамичной программной техники. Более сильный подход был предложен моделями использования цепей Маркова. Эти модели использовались как единая структура, где возможности могут быть точно изучены экспериментальным путем. Закодированные представления спектральной трансформации воспроизводства речи используются для нахождения самого правильного пути через сеть. Использование такого формально- структурного подхода, который

способствует автоматичному определению классов символов через структурирование и параметризацию.

При другом подходе базы данных и связанные с ними процессы обработки используются структурой контроля. В системах, которые изучают данный подход комплексная структура данных, которая содержит всю информацию о воспроизведении звуков, изучается с точки зрения конкретных ограничений. Но как выше указано, каждое из этих ограничений имеет особую внутреннюю модель, и полный анализ не может быть произведен. Для проведения анализа в целом структура данных должна иметь взаимодействие между разными процессами, а также средства для интеграции. Несмотря на то, что структура включает в себя несколько весьма различных источников знаний и ее вклад в понимание речи очень общий, она также имеет большое количество степеней свободы, которые могут быть использованы для тщательного системного воспроизведения. В отличие от этого, техника, основанная на цепях Маркова, имеет математическую поддержку. Чтобы иметь возможность сфокусированного исследования ограничений взаимодействия и интеграции в контексте, необходимо применять обе системы. Те системы, которые описывают ограничение взаимодействия, сфокусированы во многом на воспроизведении знаний, и они относительно слабо контролируемы, а системам с математической поддержкой, которые в свою очередь имеют великолепную технику для установления параметров и оптимизации изучения, не достает использования комплексной структуры данных, необходимых для характеристики ограничений высокого уровня, таких как синтаксис. Оба направления в настоящий момент находятся в процессе развития.

Распознавание речи на ПК

Функции распознавания речи (правда, далеко не для всех языков) в настоящее время уже реализованы в ряде коммерческих продуктов, включая операционные системы и различные приложения. Работы в этой области ведут компании IBM, Intel, Microsoft, Nuance Communications, Philips Speech

Processing и др. Из российских разработчиков стоит упомянуть “Центр речевых технологий” (ЦРТ).

Существующие алгоритмы распознавания живой речи позволяют реализовать пригодные для массового использования системы трансляции голосовых команд (отдельных слов), а также специализированные диктографы, оперирующие словарями объемом в несколько сотен слов. Например, в конце минувшего года упомянутый выше ЦРТ представил готовую к внедрению систему голосовой навигации VoiceNavigator, которая позволяет управлять банкоматом посредством голосовых команд. Ожидается, что уже в ближайшее время данное решение будет реализовано в банкоматах Сбербанка России.

К сожалению, у большинства подобных систем точность распознавания в значительной мере варьируется в зависимости от языка и особенностей дикции пользователя. Создавать высокоэффективную систему преобразования живой речи в текст, способную заменить компьютерную клавиатуру при вводе больших объемов текста произвольного содержания, пока не удалось.

Распознавание речи для мобильных устройств

Проблемы, возникающие при создании систем распознавания речи для мобильных устройств, во многом схожи с описанными в предыдущем разделе. Однако есть и своя специфика, определяемая как особенностями аппаратной речи, так и условиями эксплуатации. Во-первых, производительность процессоров, используемых в мобильных устройствах, гораздо ниже по сравнению с настольными ПК и полноразмерными ноутбуками. Вследствие этого разработчики вынуждены применять менее требовательные к аппаратным ресурсам алгоритмы распознавания. Во-вторых, мобильные устройства часто используются на улице и в общественных местах, что значительно снижает качество звукового сигнала и требует применения эффективных решений для подавления шумов и

фильтрации помех. При этом использование ресурсоемких программных решений невозможно в силу ограниченных возможностей аппаратной части.

В настоящее время системы распознавания речи в мобильных устройствах применяются главным образом для реализации функции голосового управления.

Тема 12. Синтез речи.

1.Задача синтеза речи.

2.Методы синтеза.

3.Конвертация текста в речь.

4.Анализ текста.

5.Синтез самой речи.

6.Оценка синтетической речи.

1. Задача синтеза.

Существуют различные методы синтеза речи. Выбор того или иного метода определяется различными ограничениями. Рассмотрим те 4 вида ограничений, которые влияют на выбор метода синтеза.

Возможности синтезированной речи зависят от того, в какой области она будет применяться. Когда необходимо произносить ограниченное число фраз (и их произнесение линейно не меняется), необходимый речевой материал просто записывается на пленку. С другой стороны, если задача состоит в стимулировании познавательного процесса при чтении вслух, используется совершенно другой ряд методик.

Голосовой аппарат человека.

Все системы синтеза речи должны производить на выходе какую-то речевую волну, но это не произвольный сигнал. Чтобы получить речевую волну определенного качества, сигнал должен пройти путь от источника в речевом тракте, который возбуждает действие артикуляторных органов,

которые действуют как изменяющиеся во времени фильтры. Артикуляторные органы также накладывают ограничения на скорость изменения сигнала. Они также имеют функцию сглаживания: гладкого сцепления отдельных базовых фонетических единиц в сложный речевой поток.

Структура языка.

Ряд возможных звуковых сочетаний определяется природой той или иной языковой структуры. Было обнаружено, что единицы и структуры, используемые лингвистами для описания и объяснения языка, могут также использоваться для характеристики и построения речевой волны. Таким образом, при построении выходной речевой волны используются основные фонологические законы, правила ударения, морфологические и синтаксические структуры, фонотактические ограничения.

Технология.

Возможности успешно моделировать и создавать устройства для синтеза речи в сильной степени зависят от состояния технико-технологической стороны дела. Речевая наука сделала большой шаг вперед благодаря появлению различных технологий, в том числе: рентгенография, кинематография, теория фильтров и спектров, а главным образом - цифровые компьютеры. С приходом интегральных сетевых технологий с постоянно возрастающими возможностями стало возможно построение мощных, компактных, недорогих устройств, действующих в реальном времени. Этот факт, вместе с основательными знаниями алгоритмов синтеза речи, стимулировал дальнейшее развитие систем синтеза речи и переход их в практическую жизнь, где они находят широкое применение.

2. Методы синтеза.

Различные подходы могут быть сгруппированы по областям их применения, по сложности их воплощения.

Синтезаторы делят на два типа: с ограниченным и неограниченным словарем. В устройствах с ограниченным словарем речь хранится в виде слов и предложений, которые выводятся в определенной последовательности

при синтезе речевого сообщения. Речевые единицы, используемые в синтезаторах подобного типа, произносятся диктором заранее, а затем преобразуются в цифровую форму, что достигается с помощью различных методов кодирования, позволяющих компрессировать речевую информацию и хранить ее в памяти синтезирующего устройства. Существует несколько методов записи и компоновки речи.

Волновой метод кодирования.

Самый легкий путь - просто записать материал на пленку и по необходимости проигрывать. Этот способ обеспечивает высокое качество синтезируемой речи, т.к. позволяет воспроизводить форму естественного речевого сигнала. Однако этот путь синтеза не позволяет реализовать построение новой фразы, т.к. не предусматривает обращение к различным ячейкам памяти и вызов из памяти нужных слов. В зависимости от используемой технологии этот способ может представлять задержки в доступе и иметь ограничения, связанные с возможностями записи. Никаких знаний об устройстве речевого тракта и структуре языка не требуется. Единственно серьезное ограничение в данном случае имеет объем памяти. Существуют способы кодирования речевого сигнала в цифровой форме, позволяющие в несколько раз уплотнять информацию: простая модуляция данных, импульсно-кодовая модуляция, адаптивная дельтовая модуляция, адаптивное предикативное кодирование. Данные способы могут уменьшить скорость передачи данных от 50кбит/сек (нормальный вариант) до 10кбит/сек, в то время как качество речи сохраняется. Естественно, сложность операций кодирования и декодирования увеличивается со снижением числа бит в секунду. Такие системы хороши, когда словарь сообщений небольшой и фиксированный. В случае же, когда требуется соединить сообщения в более длинное, сгенерировать высококачественную речь трудно, т.к. значения параметров речевой волны нельзя изменить, а они могут не подойти в новом контексте. Во всех системах синтеза речи

устанавливается некоторый компромисс между качеством речи и гибкостью системы. Увеличение гибкости неизбежно ведет к усложнению вычислений.

Параметрическое представление.

С целью дальнейшего уменьшения требуемой памяти для хранения и обеспечения необходимой гибкости было разработано несколько способов, которые абстрагируются от речевой волны как таковой, а представляют ее в виде набора параметров. Эти параметры отражают наиболее характерную информацию либо во временной, либо в частотной области. Например, речевая волна может быть сформирована сложением отдельных гармоник заданной высоты и заданными спектральными выступами на данной частоте. Альтернативный путь состоит в том, чтобы форму речевого тракта описать в терминах акустики и искусственным путем создать набор резонансов. Этот метод синтеза экономичнее волнового, т.к. требует значительно меньшего объема памяти, но при этом он требует больше вычислений, чтобы воспроизвести исходный речевой сигнал. Данный способ дает возможность манипулировать теми параметрами, которые отвечают за качество речи (значение формант, ширина полос, частота основного тона, амплитуда сигнала). Это дает возможность склеивать сигналы, так что переходы на границах совершенно не заметны. Изменения таких параметров как частота основного тона на протяжении всего сообщения дают возможность существенно изменять интонацию и временные характеристики сообщения. Наиболее популярным в настоящее время методами кодирования в устройствах, использующих параметрическое представление сигналов, является метод, основанный на формантных резонансах и метод линейного предсказания (LPC - linear predictive coding). Для синтеза используются единицы речи различной длины: параграфы, предложения, фразы, слова, слоги, полуслоги, дифоны. Чем меньше единица синтеза, тем меньшее их количество требуется для синтеза. При этом, требуется больше вычислений, и возникают трудности коартикуляции на стыках. Преимущества этого метода: гибкость, немного памяти для хранения исходного материала,

сохранение индивидуальных характеристик диктора. Требуется соответствующая цифровая техника и знание моделей речеобразования, при этом, лингвистическая структура языка не используется.

Синтез по правилам.

Описанные выше методы синтеза ориентированы на такие речевые единицы, как слова, предварительно введенные в устройство с голоса диктора. Данный принцип лежит в основе функционирования синтезаторов с ограниченным словарем. В синтезаторах с неограниченным словарем элементами речи являются фонемы или слоги поэтому в них применяется метод синтеза по правилам, а не простая компоновка. Данный метод весьма перспективен, т.к. обеспечивает работу с любым необходимым словарем, однако качество речи значительно ниже, чем при использовании метода компоновки.

При синтезе речи по правилам также используются волновой и параметрический методы кодирования, но уже на уровне слогов.

Метод параметрического представления требует компромисса между качеством речи и возможностью изменять параметры. Исследователи обнаружили, что для синтеза речи высокого качества необходимо иметь несколько различных произношений единицы синтеза (например, слога), что ведет к увеличению словаря исходных единиц без каких бы то ни было сведений о контекстной ситуации, оправдывающей тот или иной выбор. По этой причине процесс синтеза получает еще более абстрактный характер и переходит от параметрического представления к разработке набора правил, по которым вычисляются необходимые параметры на основе вводного фонетического описания. Это вводное представление содержит само по себе мало информации. Это обычно имена фонетических сегментов (например, гласные и согласные) со знаками ударения, обозначениями тона и временных характеристик. Таким образом, метод синтеза по правилам использует малоинформационное описание на входе (менее 100 бит/сек). Этот метод дает полную свободу моделирования параметров, но необходимо

подчеркнуть, что правила моделирования несовершенны. Синтезированная речь хуже натуральной, тем не менее, она удовлетворяет тестам по разборчивости и понятности. На уровне предложения и параграфа правила предоставляют необходимую степень свободы для создания плавного речевого потока.

3. Конвертация текста в речь.

Синтез по правилам требует детального фонетического транскрибирования на входе. Хотя для запоминания этой информации требуется мало памяти, чтобы извлечь из нее необходимые параметры, необходимы знания эксперта. Для конвертации неограниченного английского текста в речь необходимо сначала проанализировать его с целью получения транскрипции, которая затем синтезируется в выходную речевую волну. Анализ текста по своей природе задача лингвистическая и включает в себя определение базовых фонетических, слоговых, морфемных и синтаксических форм, плюс - вычленение семантической и прагматической информации. Системы конвертации текста в речь являются наиболее комплексными системами синтеза речи, включающие в себя знания об устройстве речевого аппарата человека, лингвистической структуре языка, а также которые должны учитывать ограничения, накладываемые областью применения системы, технико-технологической базой. Необходимо заметить, что и текст и речь являются поверхностными представлениями базовых лингвистических форм, поэтому задача преобразования текста в речь состоит в выявлении этих базовых форм, а затем в воплощении их в речи.

Система преобразования текста в речь.

Разработка такой системы началась в конце 60-х гг. Изначально предполагалось разработать читающую машину для слепых, но система может применяться в любых ситуациях, где необходимо преобразовать текст в речь. Система имеет блок морфологического анализа, правила преобразования буква-звук, правила лексического ударения, просодический и фонематический синтез.

4. Анализ текста.

Преобразование символов в стандартную форму. В самых различных текстах можно обнаружить символы и аббревиатуры, которые не принадлежат к категории " правильно образованных слов". Такие символы как "%" и "&", аббревиатуры типа "Mr" и "Nov" должны быть преобразованы в нормальную форму. Были разработаны подробные руководства по транскрибированию чисел, дат, сум денег. Иногда возникают двусмысленные ситуации, такие как, например, использование знака дефиса в конце строки. Человек в таких случаях, чтобы определить подходящее произношение, обращается к контексту и к практическим знаниям, которые не поддаются алгоритмизации.

Морфологический анализ

В вводном тексте границы слов легко определяются. Можно хранить произношение всех английских слов. Размер словаря будет большим, но в таком подходе есть несколько привлекательных сторон. Во-первых, в любом случае необходим словарь слов, произношение которых является исключением из общих правил. Такими являются, например, заимствованные слова (parfait, tortilla). Более того, все механизмы преобразования цепочки букв в фонетические значки допускают ошибки. Интересный класс исключений составляют часто употребительные слова. Например, звук /th/ в начале слова произносится как глухой фрикативный в большинстве слов (thin, thesis, thimble). Но в наиболее частотных, таких как короткие функциональные слова the, this, there, these, those, etc. начальный звук произносится как звонкий. Также /f/ всегда произносится глухо, за исключением слова "of". Другой пример. В словах типа "shave", "behave" конечный /e/ удлиняет предшествующий гласный, но в таком частом слове как "have" это правило не действует. Наконец, конечный /s/ в "atlas", "canvas" глухой, но в функциональных словах is, was, has он произносится звонко. Таким образом, приходим к выводу, что все системы должны иметь такой словарь исключений. Что касается нормальных слов, то здесь имеется два

варианта. Первый крайний случай состоит в том, чтобы составить полный словарь. Хотя число слов ограничено, составить абсолютно полный словарь невозможно, т.к. постоянно появляются новые слова. Кроме того, в словарь необходимо будет внести все изменяемые формы слова. Другой крайний подход состоит в установлении ряда правил, которые бы преобразовывали цепочки букв в фонетические значки. Хотя эти правила очень продуктивны, нельзя избежать ошибок, что ведет к созданию словаря исключений. Чтобы правильно определить фонетическую транскрипцию слова, нужно правильно разбить слово на структурные составляющие. Было обнаружено, что важную роль в определении произношения играет морфема, минимальная синтаксическая единица языка. Система использует морфемный лексикон, что может рассматриваться как некоторый компромиссный подход между двумя крайними, упомянутыми выше. Многие английские слова можно расчленить на последовательность морфов, таких как префиксы, корни, суффиксы. Так слово "snowplows" имеет два корня и окончание, "relearn" имеет приставку и корень. Такие морфы являются атомными составляющими слова и они относительно стабильны в языке, новые морфы формируются в языке очень редко. Эффективный лексикон может иметь не более 10,000 морфов. Морфемный словарь действует вместе с процедурами анализа. Этот подход эффективен и экономичен, т.к. хранение морфемного словаря не занимает много места, а хранить все изменяемые формы слова не нужно. Так как морфы являются основными составляющими слова, проиллюстрируем их полезность при определении произношения. При соединении морфов они часто меняют свое произношение. Например, при образовании множественного числа существительных "dog" и "cat" конечный /s/ будет звонким в первом случае и глухим во втором. Это пример морфофонемного правила, касающегося реализации морфемы множественного числа в различных окружениях. Становится очевидным, что для эффективного и легкого определения произношения нужно распознать составляющие морфемы слова и обозначить их границы. Еще один плюс морфемного

анализа - обеспечение подходящей базы для использования правил преобразования буква-звук. Большинство таких правил рассматривают слово как неструктурированную последовательность букв, используя окно сканирования для нахождения согласных и гласных кластеров, которые преобразуются в фонетические значки. Буквы "t" и "h" в большинстве случаев выступают как единый согласный кластер, но в слове "hothouse" кластер /th/ разрывается границей двух разных морфем. Гласный кластер /ea/ представляет много трудностей для алгоритмов буква-звук, но в слове changeable он явно разрывается. В системе MITalk морфемный анализ всегда проводится перед правилами преобразования букв в звуки. Лежащие в основе слова морфы не всегда очевидны. Например, некоторые морфы множественного числа не всегда легко определить: mice, fish. Подобные формы заносятся в словарь. При помощи морфемного лексикона и соответствующего алгоритма анализа 95-98% слов анализируется удовлетворительно. В результате им приписывается фонетическая транскрипция и часть речи.

Правила "буква-звук" и лексическое ударение

В системе нормализованный вводный текст подвергается морфологическому анализу. Может быть, что целое слово есть в словаре морфов, как, например, слово "snow". С другой стороны, слово может быть проанализировано как последовательность соединенных морфов. В английском языке среднее число морфов в слове, примерно два. В случае, если ни целое слово не может быть найдено в словаре морфов, ни проанализировано как последовательность морфов, в этом случае применяются правила преобразования "буква-звук". Важно подчеркнуть, что этот метод никогда не применяется, если морфемный анализ удался. Конвертация последовательности букв в последовательность звуков при помощи этих правил проходит в три этапа.

Первый этап - отделение префиксов и суффиксов. Возможность отделения аффиксов не такая сильная, как в морфемном анализе, но

действует удовлетворительно. Предполагается, что после отделения префиксов и суффиксов остается одна центральная часть слова, которая состоит из одного морфа, подвергаемого затем правилам преобразования.

Второй этап состоит в преобразовании согласных в фонетические значки, начиная с наиболее длинного согласного кластера до тех пор, пока все отдельные согласные не будут преобразованы.

Последний этап - оставшиеся гласные преобразуются при помощи контекстов. Гласные преобразуются последними, потому что это наиболее трудная задача, зависящая от контекста. Например, гласный кластер /ea/ имеет 14 разных произносительных контекстов и несколько произношений (reach, tear, steak, leather).

В системе правила преобразования букв в звуки действуют в паре с широким набором правил расстановки лексического ударения. Еще 25 лет назад лингвистам не удавалось обнаружить никакой системы расстановки ударений в английских словах. В Настоящее время разработан ряд правил, эффективно справляющихся с этой задачей. Ударения зависят от синтаксической роли слова, например, прилагательное "invalid" отличается от существительного. Таких слов немного, но учитывать их необходимо. Кроме того, на некоторые суффиксы автоматически падают ударения в словах, как, например, в "engineer". Но бывают более сложные случаи, которые разрешаются применением циклических правил.

В системе разработаны несколько наборов таких правил, некоторые из которых включают в себя до 600 правил. Конечно, большинство из них употребляются довольно редко. Подразумеваются, что все сильные и неправильные формы преобразуются на стадии морфологического анализа. Правила же "буква-звук" используются для преобразования новых и неправильно написанных слов. Например, слово "recieved" получает правильную транскрипцию, благодаря этим правилам преобразования.

Парсинг.

Каждая схема преобразования неограниченного текста в речь должна включать синтаксический анализ. Необходимо определить синтаксическую роль слова, т.к. она часто влияет на произношение и ударение. Кроме того, синтаксический анализ важен для определения правильного тонального контура и временных характеристик. Просодические характеристики важны для синтеза речи, чтобы она звучала живо и естественно. К сожалению, полный синтаксический анализ на уровне сложного предложения (clause-level parsing) осуществить нельзя. Тем не менее, возможно провести синтаксический анализ на уровне фразы (phrase-level parsing), в результате которого определяется большая часть необходимой для синтеза речи структуры, хотя в некоторых ситуациях неизбежны ошибки из-за отсутствия анализа целого предложения. Встречается множество синтаксически двусмысленных предложений, таких как "he saw the man in the park with a telescope", для которых фразовый анализ достаточен.

В английском языке существует ряд синтагматических маркеров, по которым можно формально разграничить фразы: это вспомогательные глаголы, детерминативы в номинативных фразах. Система MITalk широко использует это и проводит высокоточный грамматический анализ (augmented-transition-network grammas). Фразовый анализ показал удовлетворительные результаты, хотя эффективный анализатор предложений несомненно улучшил бы работу системы. Пока анализаторы предложений сталкиваются со значительными трудностями, когда встречают неполное или синтаксически омонимичное предложение. По завершении деятельности блока синтаксического анализа система приписывает словам маркеры функциональных частей речи, отмечает синтаксические паузы как основу для дальнейшего уточнения произношения, временных характеристик, частоты основного тона.

Модификация ударения и фонологические уточнения.

Последняя фаза анализа состоит в некоторых незначительных поправках к имеющейся уже фонетической транскрипции на основе анализа

контекстного окружения. Простой пример определения произношения артикля "the", которое зависит от начального звука последующего слова. Кроме того, на этом этапе используются некоторые эвристические методы проверки правильного соотношения общего контура предложения с контурами отдельных слов. На этом этапе заканчивается подготовка исходного текста собственно к самому процессу синтеза.

5. Синтез самой речи.

Важно осознать, что в системе не используются готовые речевые волны даже в параметрическом представлении. Система не хранит параметрические представления множества морфов или слов. Вместо этого были разработаны правила контроля параметров, так что можно реализовать любую желаемую речевую волну на выходе.

Просодическая рамка.

Первый шаг в создании выходной речевой волны - создание временного контура и частоты основного тона (основные корреляты интонации), на основе которых строится детальная артикуляция отдельных фонетических элементов. Распределение ударения, которое было вычислено на стадии анализа, во многом ответственно за контур временного распределения и тональный контур. Часто интенсивность принимают за коррелят ударения, тогда как главными ключами являются длительность и изменения в тональном контуре. Согласные мало меняются по длительности, в то время как гласные более пластичны и могут легко сжиматься или растягиваться.

Существует также тенденция растягивать слова на границе основных абзацев предложения, и наоборот, сжимать интервалы на относительно невыделенных участках. Кроме того, на основе временной рамки задается частота основного тона (или тональный контур). В утвердительных предложениях обычно высота тона резко поднимается на первом ударном слоге, затем плавно снижается до последнего ударного слога, где она резко

падает. Вопросительные и повелительные предложения имеют различные тональные контуры.

Кроме целостного контура предложения существуют еще локальные ударения. Большее ударение получают слова, выражающие отрицание или сомнение (например, слово might), значение частоты основного тона на них возрастает; новая информация в предложении также больше выделяется ударением. С другой стороны, высота тона используется в семантических и эмоциональных целях, что не может быть выведено из письменного текста. Необходимо лишний раз подчеркнуть важность составления правильного просодического контура, т.к. неправильный просодический контур может привести к трудностям в восприятии.

Синтез фонетических сегментов.

Когда завершено создание просодической рамки, создаются параметры, соответствующие модели речевого тракта. Обычно таких параметров 25, которые изменяются с интервалом 5 - 10 мсек. В настоящее время используются около 100 контекстных правил описания траектории изменения параметров. Когда значения параметров вычислены, они должны быть перенесены на соответствующую модель речевого тракта (обычно это формантная модель или LPC-модель). Выходная дискретная модель создается обычно на частоте 10 Кгц.

6. Оценка синтетической речи.

С точки зрения понятности, разборчивости качество синтезированной речи достаточно хорошее. Был проведен тест, где одна группа испытуемых прослушивала синтезированную речь с письменным вариантом перед глазами, а другая - без. Выяснилось, что результаты прослушивания мало отличаются друг от друга. Тем не менее, синтезированной речи не хватает живости и естественности, поэтому воспринимать ее на протяжении длительного времени трудно. Исследования показали, что фрикативные и назальные звуки требуют дальнейшего улучшения качества.

Технологии синтеза речи на нынешнем этапе развития обеспечивают достаточно понятное (для восприятия слушателем смысла) воспроизведение печатного текста. При этом обычно имеется возможность настраивать ряд параметров голоса “электронного диктора”: выбирать его пол (мужской или женский), менять высоту и тембр, варьировать темп речи. Вместе с тем из-за отсутствия ряда важных компонентов живой речи (правильного интонирования, естественных пауз, изменения темпа и т.п.) смысл текста, произносимого диктором-роботом, усваивается хуже.

В настоящее время технологии синтеза речи находятся в стадии коммерциализации. Системы синтеза речи широко используются в call-центрах, автоматизированных системах оповещения и обслуживания абонентов телефонных сетей и т.д. Кроме того, функции синтеза речи реализованы в ряде портативных электронных устройств, в частности в автомобильных GPS-навигаторах, ридерах и некоторых других. В минувшем году модуль синтеза речи был встроен в онлайновый переводчик Google, благодаря чему переведенное слово или фразу теперь можно не только прочитать, но и прослушать.

В настоящее время технологии синтеза речи находятся в начальной стадии коммерциализации, однако несовершенство существующих решений пока в значительной мере ограничивает сферу их применения.

Контрольные вопросы.

1. Как вы понимаете «порождение текстов на ЕЯ»?
2. Какие проблемы в задачах порождения текстов на ЕЯ актуальны в настоящее время?
3. Как происходит процесс порождения текста?
4. Кто первым провел исследования по использованию сетей распознавания?
5. Что такое фразовый словарь?
6. Кто первым сформулировал идею автоматического перевода?
7. Когда начались исследования в области автоматического перевода?

8. Как можно охарактеризовать состояние работ в области автоматического перевода в 60-х г.г. прошлого века?
9. Название ведущей Российской компании, ведущей исследования и разработки в области автоматического перевода?
10. Назовите фамилии трех самых известных российских специалистов, внесших наибольший вклад в начальный период исследований проблем машинного перевода и компьютерной лингвистики в целом.
11. Каковы основные недостатки традиционной лингвистики с точки зрения задач компьютерной лингвистики?
12. Назовите ту особенность естественного языка, которая более всего затрудняет алгоритмизацию процедур автоматического перевода.
13. Что обозначает аббревиатура TMS применительно к исследованиям и разработкам в области автоматического перевода?
14. Восемь типов памяти перевода.
15. Аспекты использования памяти переводов.
16. Пути расширения возможностей
17. Что входит в состав среды перевода помимо памяти переводов?
18. В каком порядке следует предоставлять найденные сегменты переводчику?
19. Как реализуется задача добавления нового сегмента в память переводов?
20. Дайте определение пересечения сегментов.
21. Как обычно трактуют понимание речи?
22. Чем отличается понимание речи от распознавания речи?
23. Где применяются СПР.
24. Что определяет фонетическое представление?
25. Структура системы понимания речи.
26. Какие задачи следует решить для успешного распознавания речи?
27. Что включают в себя аспекты влияния акустической структуры?
28. Что является первым уровнем артикуляторного контроля?

29. Назовите 4 вида ограничений, которые влияют на выбор метода синтеза.
30. Назовите методы записи и компоновки речи.
31. Что включает в себя анализ текста?
32. Для чего изначально предназначалась система преобразования текста в речь?
33. Назовите этапы конвертации последовательности букв в последовательность звуков.

Тема 13. Экспертные системы, их особенности. Применение экспертных систем.

1. Определение экспертных систем. Главное достоинство и назначение экспертных систем.
2. Отличие ЭС от других программных продуктов.
3. Отличительные особенности. Экспертные системы первого и второго поколения.
4. Области применения экспертных систем.
5. Критерий использования ЭС для решения задач.
6. Ограничения в применение экспертных систем.
7. Преимущества ЭС перед человеком - экспертом.

1.Определение экспертных систем. Главное достоинство и назначение экспертных систем.

Экспертные системы (ЭС) возникли как значительный практический результат в применении и развитии методов искусственного интеллекта (ИИ)- совокупности научных дисциплин, изучающих методы решения задач интеллектуального (творческого) характера с использованием ЭВМ.

Область ИИ имеет более чем сорокалетнюю историю развития. С самого начала в ней рассматривался ряд весьма сложных задач, которые, наряду с другими, и до сих пор являются предметом исследований:

автоматические доказательства теорем, машинный перевод (автоматический перевод с одного естественного языка на другой), распознавание изображений и анализ сцен, планирование действий роботов, алгоритмы и стратегии игр.

ЭС - это набор программ, выполняющий функции эксперта при решении задач из некоторой предметной области. ЭС выдают советы, проводят анализ, дают консультации, ставят диагноз. Практическое применение ЭС на предприятиях способствует эффективности работы и повышению квалификации специалистов.

Главным достоинством экспертных систем является возможность накопления знаний и сохранение их длительное время. В отличие от человека к любой информации экспертные системы подходят объективно, что улучшает качество проводимой экспертизы. При решении задач, требующих обработки большого объема знаний, возможность возникновения ошибки при переборе очень мала.

При создании ЭС возникает ряд затруднений. Это, прежде всего, связано с тем, что заказчик не всегда может точно сформулировать свои требования к разрабатываемой системе. Также возможно возникновение трудностей чисто психологического порядка: при создании базы знаний системы эксперт может препятствовать передаче своих знаний, опасаясь, что впоследствии его заменят “машиной”. Но эти страхи не обоснованы, т. к. ЭС не способны обучаться, они не обладают здравым смыслом, интуицией. Но в настоящее время ведутся разработки экспертных систем, реализующих идею самообучения. Также ЭС неприменимы в больших предметных областях и в тех областях, где отсутствуют эксперты.

Экспертная система состоит из:

базы знаний (части системы, в которой содержатся факты);

подсистемы вывода (множества правил, по которым осуществляется решение задачи);

подсистемы объяснения;

подсистемы приобретения знаний;

диалогового процессора.

При построении подсистем вывода используют методы решения задач искусственного интеллекта.

Экспертные системы (ЭС) - это яркое и быстро прогрессирующее направление в области искусственного интеллекта (ИИ). Причиной повышенного интереса, который вызывают к себе ЭС на протяжении всего своего существования, является возможность их применения к решению задач из самых различных областей человеческой деятельности. Пожалуй, не найдется такой проблемной области, в которой не было бы создано ни одной ЭС или по крайней мере, такие попытки не предпринимались бы.

ЭС выдают советы, проводят анализ, выполняют классификацию, дают консультации и ставят диагноз. Они ориентированы на решение задач, обычно требующих проведения экспертизы человеком-специалистом. В отличие от машинных программ, использующий процедурный анализ, ЭС решают задачи в узкой предметной области (конкретной области экспертизы) на основе дедуктивных рассуждений. Такие системы часто оказываются способными найти решение задач, которые неструктурированны и плохо определены. Они справляются с отсутствием структурированности путем привлечения эвристик, т. е. правил, взятых «с потолка», что может быть полезным в тех системах, когда недостаток необходимых знаний или времени исключает возможность проведения полного анализа.

Главное достоинство ЭС - возможность накапливать знания, сохранять их длительное время, обновлять и тем самым обеспечивать относительную независимость конкретной организации от наличия в ней квалифицированных специалистов. Накопление знаний позволяет повышать квалификацию специалистов, работающих на предприятии, используя наилучшие, проверенные решения.

Практическое применение искусственного интеллекта на машиностроительных предприятиях и в экономике основано на ЭС,

позволяющих повысить качество и сохранить время принятия решений, а также способствующих росту эффективности работы и повышению квалификации специалистов.

2. Отличие ЭС от других программных продуктов.

Основными отличиями ЭС от других программных продуктов являются использование не только данных, но и знаний, а также специального механизма вывода решений и новых знаний на основе имеющихся. Знания в ЭС представляются в такой форме, которая может быть легко обработана на ЭВМ. В ЭС известен алгоритм обработки знаний, а не алгоритм решения задачи. Поэтому применение алгоритма обработки знаний может привести к получению такого результата при решении конкретной задачи, который не был предусмотрен. Более того, алгоритм обработки знаний заранее неизвестен и строится по ходу решения задачи на основании эвристических правил.

Решение задачи в ЭС сопровождается понятными пользователю объяснениями, качество получаемых решений обычно не хуже, а иногда и лучше достигаемого специалистами. В системах, основанных на знаниях, правила (или эвристики), по которым решаются проблемы в конкретной предметной области, хранятся в базе знаний. Проблемы ставятся перед системой в виде совокупности фактов, описывающих некоторую ситуацию, и система с помощью базы знаний пытается вывести заключение из этих фактов.

Качество ЭС определяется размером и качеством базы знаний (правил или эвристик). Система функционирует в следующем циклическом режиме:

выбор (запрос) данных или результатов анализов,

наблюдения,

интерпретация результатов,

усвоение новой информации, выдвижении с помощью правил временных гипотез,

и затем выбор следующей порции данных или результатов анализов.

Такой процесс продолжается до тех пор, пока не поступит информация, достаточная для окончательного заключения.

В любой момент времени в системе существуют три типа знаний:

Структурированные знания - статические знания о предметной области. После того как эти знания выявлены, они уже не изменяются.

Структурированные динамические знания - изменяемые знания о предметной области. Они обновляются по мере выявления новой информации.

Рабочие знания - знания, применяемые для решения конкретной задачи или проведения консультации.

Все перечисленные выше знания хранятся в базе знаний. Для ее построения требуется провести опрос специалистов, являющихся экспертами в конкретной предметной области, а затем систематизировать, организовать и снабдить эти знания указателями, чтобы впоследствии их можно было легко извлечь из базы знаний.

3. Отличительные особенности. Экспертные системы первого и второго поколения.

Экспертиза может проводиться только в одной конкретной области. Так, программа, предназначенная для определения конфигурации систем ЭВМ, не может ставить медицинские диагнозы.

База знаний и механизм вывода являются различными компонентами. Действительно, часто оказывается возможным сочетать механизм вывода с другими базами знаний для создания новых ЭС. Например, программа анализа инфекции в крови может быть применена в пульмонологии путем замены базы знаний, используемой с тем же самым механизмом вывода.

Наиболее подходящая область применения - решение задач дедуктивным методом. Например, правила или эвристики выражаются в виде пар посылок и заключений типа “если - то”.

Эти системы могут объяснять ход решения задачи понятным пользователю способом. Обычно мы не принимаем ответ эксперта, если на вопрос «Почему?» не можем получить логичный ответ. Точно так же мы должны иметь возможность спросить систему, основанную на знаниях, как было получено конкретное заключение.

5. Выходные результаты являются качественными (а не количественными).

6. Системы, основанные на знаниях, строятся по модульному принципу, что позволяет постепенно наращивать их базы знаний.

Компьютерные системы, которые могут лишь повторить логический вывод эксперта, принято относить к ЭС *первого поколения*. Однако специалисту, решающему интеллектуально сложную задачу, явно недостаточно возможностей системы, которая лишь имитирует деятельность человека. Ему нужно, чтобы ЭС выступала в роли полноценного помощника и советчика, способного проводить анализ нечисловых данных, выдвигать и отбрасывать гипотезы, оценивать достоверность фактов, самостоятельно пополнять свои знания, контролировать их непротиворечивость, делать заключения на основе прецедентов и, может быть, даже порождать решение новых, ранее не рассматривавшихся задач. Наличие таких возможностей является характерным для ЭС *второго поколения,* концепция которых начала разрабатываться 9-10 лет назад. Экспертные системы, относящиеся ко второму поколению, называют партнерскими, или усилителями интеллектуальных способностей человека. Их общими отличительными чертами является умение обучаться и развиваться, т.е. эволюционировать.

В экспертных системах первого поколения знания представлены следующим образом:

1) знаниями системы являются только знания эксперта, опыт накопления знаний не предусматривается.

2) методы представления знаний позволяли описывать лишь статические предметные области.

3) модели представления знаний ориентированы на простые области.

Представление знаний в экспертных системах второго поколения следующее:

1) используются не поверхностные знания, а более глубокие. Возможно дополнение предметной области.

2) ЭС может решать задачи динамической базы данных предметной области.

4. Области применения экспертных систем.

Области применения систем, основанных на знаниях, могут быть сгруппированы в несколько основных классов: медицинская диагностика, контроль и управление, диагностика неисправностей в механических и электрических устройствах, обучение.

Медицинская диагностика.

Диагностические системы используются для установления связи между нарушениями деятельности организма и их возможными причинами. Наиболее известна диагностическая система MYCIN, которая предназначена для диагностики и наблюдения за состоянием больного при менингите и бактериальных инфекциях. Ее первая версия была разработана в Стенфордском университете в середине 70-х годов. В настоящее время эта система ставит диагноз на уровне врача-специалиста. Она имеет расширенную базу знаний, благодаря чему может применяться и в других областях медицины.

Прогнозирование.

Прогнозирующие системы предсказывают возможные результаты или события на основе данных о текущем состоянии объекта. Программная система “Завоевание Уолл-Стрита” может проанализировать конъюнктуру рынка и с помощью статистических методов алгоритмов разработать для вас план капиталовложений на перспективу. Она не относится к числу систем, основанных на знаниях, поскольку использует процедуры и алгоритмы

традиционного программирования. Хотя пока еще отсутствуют ЭС, которые способны за счет своей информации о конъюнктуре рынка помочь вам увеличить капитал, прогнозирующие системы уже сегодня могут предсказывать погоду, урожайность и поток пассажиров. Даже на персональном компьютере, установив простую систему, основанную на знаниях, вы можете получить местный прогноз погоды.

Планирование.

Планирующие системы предназначены для достижения конкретных целей при решении задач с большим числом переменных. Дамасская фирма Informat впервые в торговой практике предоставляет в распоряжении покупателей 13 рабочих станций, установленных в холле своего офиса, на которых проводятся бесплатные 15-минутные консультации с целью помочь покупателям выбрать компьютер, в наибольшей степени отвечающий их потребностям и бюджету. Кроме того, компания Boeing применяет ЭС для проектирования космических станций, а также для выявления причин отказов самолетных двигателей и ремонта вертолетов. Экспертная система XCON, созданная фирмой DEC, служит для определения или изменения конфигурации компьютерных систем типа VAX и в соответствии с требованиями покупателя. Фирма DEC разрабатывает более мощную систему XSEL, включающую базу знаний системы XCON, с целью оказания помощи покупателям при выборе вычислительных систем с нужной конфигурацией. В отличие от XCON система XSEL является интерактивной.

Интерпретация.

Интерпретирующие системы обладают способностью получать определенные заключения на основе результатов наблюдения. Система PROSPECTOR, одна из наиболее известных систем интерпретирующего типа, объединяет знания девяти экспертов. Используя сочетания девяти методов экспертизы, системе удалось обнаружить залежи руды стоимостью в миллион долларов, причем наличие этих залежей не предполагал ни один из девяти экспертов. Другая интерпретирующая система- HASP/SIAP. Она

определяет местоположение и типы судов в тихом океане по данным акустических систем слежения.

Контроль и управление.

Системы, основанные на знаниях, могут применятся в качестве интеллектуальных систем контроля и принимать решения, анализируя данные, поступающие от нескольких источников. Такие системы уже работают на атомных электростанциях, управляют воздушным движением и осуществляют медицинский контроль. Они могут быть также полезны при регулировании финансовой деятельности предприятия и оказывать помощь при выработке решений в критических ситуациях.

Диагностика неисправностей в механических и электрических устройствах.

В этой сфере системы, основанные на знаниях, незаменимы как при ремонте механических и электрических машин (автомобилей, дизельных локомотивов и т.д.), так и при устранении неисправностей и ошибок в аппаратном и программном обеспечении компьютеров.

Обучение.

Системы, основанные на знаниях, могут входить составной частью в компьютерные системы обучения. Система получает информацию о деятельности некоторого объекта (например, студента) и анализирует его поведение. База знаний изменяется в соответствии с поведением объекта. Примером этого обучения может служить компьютерная игра, сложность которой увеличивается по мере возрастания степени квалификации играющего. Одной из наиболее интересных обучающих ЭС является разработанная Д.Ленатом система EURISCO, которая использует простые эвристики. Эта система была опробована в игре Т.Тревевеллера, имитирующая боевые действия. Суть игры состоит в том, чтобы определить состав флотилии, способной нанести поражение в условиях неизменяемого множества правил. Система EURISCO включила в состав флотилии небольшие, способные провести быструю атаку корабли и одно очень

маленькое скоростное судно и постоянно выигрывала в течение трех лет, несмотря на то, что в стремлении воспрепятствовать этому правила игры меняли каждый год.

Большинство ЭС включают знания, по содержанию которых их можно отнести одновременно к нескольким типам. Например, обучающая система может также обладать знаниями, позволяющими выполнять диагностику и планирование. Она определяет способности обучаемого по основным направлениям курса, а затем с учетом полученных данных составляет учебный план. Управляющая система может применяться для целей контроля, диагностики, прогнозирования и планирования. Система, обеспечивающая сохранность жилища, может следить за окружающей обстановкой, распознавать происходящие события (например, открылось окно), выдавать прогноз (вор-взломщик намеревается проникнуть в дом) и составлять план действий (вызвать полицию).

5. Критерий использования ЭС для решения задач.

Существует ряд прикладных задач, которые решаются с помощью систем, основанных на знаниях, более успешно, чем любыми другими средствами. При определении целесообразности применения таких систем нужно руководствоваться следующими критериями.

1. Данные и знания надежны и не меняются со временем.

2. Пространство возможных решений относительно невелико.

3. В процессе решения задачи должны использоваться формальные рассуждения. Существуют системы, основанные на знаниях, пока еще не пригодные для решения задач методами проведения аналогий или абстрагирования (человеческий мозг справляется с этим лучше). В свою очередь традиционные компьютерные программы оказываются эффективнее систем, основанных на знаниях, в тех случаях, когда решение задачи связано с применением процедурного анализа. Системы, основанные на знаниях, более подходят для решения задач, где требуются формальные рассуждения.

4. Должен быть по крайней мере один эксперт, который способен явно сформулировать свои знания и объяснить свои методы применения этих знаний для решения задач.

В целом ЭС не рекомендуется применять для решения следующих типов задач:

- математических, решаемых обычным путем формальных преобразований и процедурного анализа;

- задач распознавания, поскольку в общем случае они решаются численными методами;

- задач, знания о методах решения которых отсутствуют (невозможно построить базу знаний).

6. Ограничения в применение экспертных систем.

Даже лучшие из существующих ЭС, которые эффективно функционируют, как на больших, так и на мини-ЭВМ, имеют определенные ограничения по сравнению с человеком - экспертом.

1. Большинство ЭС не вполне пригодны для применения конечным пользователем. Если вы не имеете некоторого опыта работы с такими системами, то у вас могут возникнуть серьезные трудности. Многие системы оказываются доступными только тем экспертам, которые создавали из базы знаний.

2. Вопросно-ответный режим, обычно принятый в таких системах, замедляет получение решений. Например, без системы MYCIN врач может (а часто и должен) принять решение значительно быстрее, чем с ее помощью.

3. Навыки системы не возрастают после сеанса экспертизы.

4. Все еще остается проблемой приведение знаний, полученных от эксперта, к виду, обеспечивающему их эффективную машинную реализацию.

5. ЭС не способны обучаться, не обладают здравым смыслом. Домашние кошки способны обучаться даже без специальной дрессировки, ребенок в состоянии легко уяснить, что он станет мокрым, если опрокинет на

себя стакан с водой, однако если начать выливать кофе на клавиатуру компьютера, у него не хватит “ума” отодвинуть ее.

6. ЭС не применяют в больших предметных областях. Их использование ограничивается предметными областями, в которых эксперт может принять решение за время от нескольких минут до нескольких часов.

7. В тех областях, где отсутствуют эксперты (например, в астрологии), применение ЭС оказывается невозможным.

8. Имеет смысл привлекать ЭС только для решения когнитивных задач. Теннис, езда на велосипеде не могут являться предметной областью для ЭС, однако такие системы можно использовать при формировании футбольных команд.

9. Человек-эксперт при решении задач обычно обращается к своей интуиции или здравому смыслу, если отсутствуют формальные методы решения или аналоги таких задач.

Системы, основанные на знаниях, оказываются неэффективными при необходимости проведения скрупулезного анализа, когда число “решений” зависит от тысяч различных возможностей и многих переменных, которые изменяются во времени. В таких случаях лучше использовать базы данных с интерфейсом на естественном языке.

7. Преимущества ЭС перед человеком - экспертом.

Системы, основанные на знаниях, имеют определенные преимущества перед человеком-экспертом.

1. У них нет предубеждений.

2. Они не делают поспешных выводов.

3. Эти системы работают систематизировано, рассматривая все детали, часто выбирая наилучшую альтернативу из всех возможных.

4. База знаний может быть очень и очень большой. Будучи введены в машину один раз, знания сохраняются навсегда. Человек же имеет ограниченную базу знаний, и если данные долгое время не используются, то они забываются и навсегда теряются.

Системы, основанные на знаниях, устойчивы к “помехам”. Эксперт пользуется побочными знаниями и легко поддается влиянию внешних факторов, которые непосредственно не связаны с решаемой задачей. ЭС, не обремененные знаниями из других областей, по своей природе менее подвержены “шумам”. Со временем системы, основанные на знаниях, могут рассматриваться пользователями как разновидность тиражирования- новый способ записи и распространения знаний. Подобно другим видам компьютерных программ они не могут заменить человека в решении задач, а скорее напоминают орудия труда, которые дают ему возможность решат задачи быстрее и эффективнее.

6. Эти системы не заменяют специалиста, а являются инструментом в его руках.

Контрольные вопросы.

1. Что отличает экспертные системы от других программ?
2. Какие категории различных типичных проблем, решаемых экспертными системами, можно выделить?
3. Охарактеризуйте экспертную систему MYCIN.
4. Охарактеризуйте экспертную систему DENDRAL.
5. Охарактеризуйте экспертную систему PROSPECTOR.
6. Какие виды инструментальных программных средств для создания экспертных систем существуют?
7. Какие еще экспертные системы вы знаете?
8. Дайте краткую характеристику систем с интеллектуальным интерфейсом, экспертных систем, самообучающихся систем и адаптивных информационных систем.
9. Каким требованиям должны удовлетворять адаптивные систе-мы?
10. Перечислите сферы применения экспертных систем.
11. Назовите основные классы экспертных систем.

12. Какие этапы составляют поиск решения на основе аналогий в системах, основанных на прецедентах.

10. Дайте определение нейронной сети.

11. Экспертная система как один из видов программ искусственного интеллекта.

12. Основные компоненты экспертных систем.

13. Экономические экспертные системы.

14. Их состав и функции.

15. Режимы работы экспертных систем.

16. Области применения экспертных систем.

17. Какие средства автоматизации создания экспертных систем существуют в настоящее время?

18. Перечислите основные функции, которые должна выполнять интеллектуальная информационная технология?

19. В чем особенность и чем определяется эффективность интеллектуальных информационных технологий?

20. Объясните назначение блоков экспертной системы.

21. В чем главное отличие экспертной системы от информаци-онно-поисковой системы?

22. Как моделируются системы искусственного интеллекта?

23. Назовите примеры успешного применения технологии ЭС.

24. Объясните основные причины успеха современной технологии С .

25. Дайте формальное определение продукционной системы (по Е.Посту и А.Ньюэллу).

26. Охарактеризуйте основные режимы работы ЭС.

27. Укажите состав и роли участников разработки ЭС.

28. Перечислите основные компоненты статической ЭС.

29. Поясните отличия архитектуры динамической ЭС от архитектуры стати-ческой ЭС.

30. Перечислите и охарактеризуйте основные этапы разработки ЭС.