Федеральное агентство по образованию

Государственное образовательной учреждение
высшего профессионального образования

«ВОСТОЧНО-СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ» (ГОУ ВПО «ВСГТУ»)

ЕСТЕСТВЕННО-ЯЗЫКОВЫЕ СИСТЕМЫ

Курс лекций

Улан-Удэ, 2006
Издательство ВСГТУ

УДК 004.8 (075.8)
ББК 32.813 я73
Е155

Рецензент: Найханова Л.В., к.т.н., доцент, заведующая кафедрой систем информатики ВСГТУ

Печатается по решению редакционно-издательского совета ВСГТУ

Курс лекций предназначен для студентов старших курсов специализации «Искусственный интеллект» специальностей 230105 «Программное обеспечение вычислительной техники и автоматизированных систем» и 010503 «Математическое обеспечение и администрирование информационных систем». Данный курс содержит теоретический материал по одноименной дисциплине, в нем изложены основные понятия систем общения на естественном языке, рассматриваются архитектура и основные классы ЕЯ-систем, вопросы создания основных компонент ЕЯ-систем и их методы организации.

И.С. Евдокимова. Естественно-языковые системы: курс лекций. – Улан-Удэ: Изд-во ВСГТУ, 2006. – 92 с.: илл.

ISBN 5-89230-182-6

ББК 32.813 я73

Содержание

ВВЕДЕНИЕ

В конце 60-х годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название «обработка естественного языка» (Natural Language Processing). Задачей данного направления является исследование методов и разработка систем, обеспечивающих реализацию процесса общения с компьютерными системами на естественном языке (систем ЕЯ - общения или ЕЯ-систем). Следует отметить, что проблематика коммуникативного взаимодействия, и в частности ЕЯ-общения, находится в центре внимания многих наук, например, лингвистики, психологии, логики и философии. Однако все они исследуют лишь отдельные аспекты процесса общения. В отличие от них искусственный интеллект, как прикладная дисциплина, вынужден моделировать в рамках ЕЯ-систем все или, по крайней мере, основные аспекты ЕЯ-общения, правда, может быть, не на столь глубоком уровне.

Проблема взаимодействия человека с компьютером существует с момента появления вычислительной техники. На начальном этапе непосредственное взаимодействие с ЭВМ осуществляли только программисты, а специалисты других областей - потребители результатов, полученных на компьютере, выступали в роли косвенных конечных пользователей, т. е. общались с компьютером через программистов. По мере расширения сферы использования компьютера и увеличения масштабов их применения конечные пользователи стали вовлекаться в процесс непосредственного взаимодействия с компьютером, что привело к появлению массовой категории пользователей - прямых конечных пользователей, работающих в диалоговом режиме. К пользователям этой категория относятся специалисты в различных проблемных областях, которые решают свои профессиональные задачи, непосредственно используя компьютер, т. е. прикладные программы и пакеты (прикладные системы), входящие в программное обеспечение компьютера. Как правило, эти пользователи не обладают знаниями в области компьютерной обработки данных и не умеют программировать. Поэтому часто их называют неподготовленными конечными пользователями. В дальнейшем термины «пользователь» и «конечный пользователь» будут использоваться в смысле «неподготовленный конечный пользователь».

Сложность создания средств общения, предназначенных для конечных пользователей, обусловлена в значительной степени отсутствием единой теории языкового общения, охватывающей все аспекты взаимодействия коммуникантов. Поэтому при разработке средств общения конечных пользователей на процесс взаимодействия часто налагаются различные «спонтанные» ограничения, последствия которых не до конца осознаются разработчиками. Эти ограничения приводят к тому, что многие человеко-машинные системы, на разработку которых тратятся огромные средства, не удовлетворяют требованиям конечных пользователей.

Естественно-языковые системы используются для поиска в текстах, распознавания речи, голосового управления и обработки данных. Их доля на рынке составляет около 14%. В данном направлении выделяются следующие категории информационных продуктов:

- средства, обеспечивающие естественно-языковый интерфейс к базам данных;
- средства естественно-языкового поиска в текстах и содержательного сканирования текстов (Natural Language text retrieval and Contents Scanning Systems);
- масштабируемые средства для распознавания речи (Large-Vocabulary Talkwriter);
- средства голосового ввода, управления и сбора данных (Voice Input and Control Products and Data Collection Systems);
- компоненты речевой обработки (Voice-Recognition Programming Tools).

Программные продукты первой категории преобразуют естественно-языковые запросы пользователя в SQL-запросы к базам данных. Средства естественно-языкового поиска в текстах осуществляют по запросам пользователей поиск, фильтрацию и сканирование текстовой информации. В отличие от продуктов предыдущей группы, где поиск осуществляется в базах данных, имеющих четкую и заранее известную структуру, средства данной категории осуществляют поиск в неструктурированных текстах, оформленных в соответствии с правилами грамматики того или иного естественного языка. Средства для распознавания речи распознают голосовую информацию и преобразуют ее в последовательность символов. Они ориентированы на работу со словарями объемом от 30000 до 70000 слов. В отличие от этого, средства голосового ввода ориентированы на работу со словарем около 1000 слов и существенно ограничены в возможностях распознавания. Программные средства этого типа предназначены для ввода голосовых команд, управляющих работой некоторого продукта, например, программы сбора данных в тех приложениях, в которых у исполнителей заняты руки.

Компоненты речевой обработки предназначены для программистов, которых хотят добавить возможности по распознаванию речи в разрабатываемые ими приложения.

Данный курс лекций посвящен естественно-языковым системам, относящимся к системам первой и второй категории.

РАЗДЕЛ I. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ СИСТЕМ

Лекция 1. Архитектура естественно-языковых систем

Прежде чем приступить к рассмотрению архитектуры ЕЯ-систем, остановимся на определении некоторых исходных понятий, которые будут широко использоваться далее.

Общение - коммуникативное взаимодействие.

Диалог - процесс достижения его участниками определенных согласованных целей путем обмена связанными высказываниями, выраженными в языке, о некотором реальном или гипотетическом мире (проблемной области).

Говоря об общении человека с компьютером, обычно считают, что цель компьютера состоит в том, чтобы способствовать достижению целей пользователя, которые определяются его информационными потребностями. Поэтому применительно к диалогу между пользователем и компьютером под *общением понимают процесс обмена взаимосвязанными высказываниями, выраженными в языке, направленный на достижение целей пользователя, т.е. на удовлетворение информационных потребностей пользователя (ИПП).*

В общем случае процесс общения не может быть сведен к обмену изолированными парами высказываний «вопрос-ответ». Высказывания участников общения образуют связный текст - *дискурс,* имеющий, как правило, достаточно сложную структуру. Связность дискурса обеспечивается как лингвистическими (родовидовыми, анафорическими, модальными, стилистическими согласованиями, согласованиями пресуппозиций и т.п.), так и экстралингвистическими (ситуативными) средствами, т. е. с помощью временных, причинно-следственных и других связей, существующих в проблемной области.

Следует подчеркнуть, что разговорный ЕЯ гораздо более компактен, чем литературный, «письменный» язык, так как при общении широко используются *разнообразные умолчания* (эллипсисы, анафоры, пресуппозиции и др.), восстанавливаемые (раскрываемые) участниками исходя из текущих целей диалога. Если участники достигли цели, поставленной в начале общения, то говорят, что общение завершилось *успехом (глобальным успехом),* в противном случае - *неудачей (глобальной неудачей).* В процессе общения могут возникать различные локальные неудачи, вызванные, например, неправильностью (нарушением грамматических норм) высказываний участников, непониманием друг друга из-за различных представлений о теме диалога или о проблемной области на языке общения и т.п. Большая часть локальных неудач не приводит к глобальной неудаче, которые преодолеваются участниками общения гибкой (в ходе диалога) корректировкой текущих целей.

Цели, преследуемые участниками общения, определяют структуру диалога, которая может рассматриваться на трех уровнях:

- глобальном;
- тематическом;
- локальном.

На *глобальном уровне* определяются общие свойства решаемых пользователями задач.

На *тематическом уровне* структура диалога зависит от конкретных особенностей решаемой задачи - от алгоритма ее решения (разбиения задач на подзадачи) и распределения ролей (активная или пассивная роль) между участниками общения при решении отдельных подзадач. На *локальном уровне* рассматриваются отдельные шаги диалога, образуемые взаимосвязанными высказываниями его участников.

Шаг диалога трактуется как пара «действие-реакция», где высказывание активного (т.е. владеющего инициативой) участника соответствует действию, а пассивного - реакции. Основными параметрами структуры диалога на этом уровне являются:

- инициатор шага и вид инициирования (вид действия);
- способ влияния действия на реакцию;
- способ спецификации задачи (подзадачи), решаемой на данном шаге.

Действие и реакция, образующие шаг диалога, могут в общем случае не соответствовать соседним (во временной последовательности) высказываниям участников. Соответствие нарушается при перехватах инициативы. *Перехват инициативы* возникает в тех случаях, когда пассивный участник вместо цели (подцели), предложенной активным участником, выбирает иные цели (подцели), в частности, подцели, предусматривающие преодоление локальных неудач. Например, вместо ответа на вопрос (что соответствовало бы стандартной реакции) второй участник может задать встречный вопрос (т. е. совершить действие и тем самым взять на себя активную роль) и лишь после получения ответа на него, ответить на первоначально заданный вопрос (и тем самым вернуть инициативу). Таким образом, перехват инициативы как бы разрывает первоначально инициированный шаг диалога и открывает поддиалог - происходит смена цели (темы) диалога, в котором инициативой владеет ранее пассивный участник.

Переходя к рассмотрению человеко-машинного общения, подчеркнем, что согласно современным представлениям, взаимодействие конечных пользователей с компьютером происходит на всех стадиях существования человеко-машинной системы, т. е. на стадиях использования, разработки и развития приложений. Традиционные средства общения, которые вплоть до настоящего времени широко применяются на практике, ориентированы, как правило, либо только на использование заранее разработанных и неизменяемых приложений, либо на использование, разработку и развитие. В первом случае процесс взаимодействия сводится к трем этапам:

- определение параметров работы системы (вход в систему):
- определение решаемой задачи и исходных данных;
- получение результатов решения задачи.

Такой процесс принципиально не может удовлетворить пользователей с изменяющейся информационной потребностью, не знающих способа представления и использования в системе информации, которой обмениваются участники общения.

Во втором случае взаимодействие осуществляется с помощью процедурного языка программирования, что не удовлетворяет большинство конечных пользователей, обычно не умеющих (и не желающих) программировать.

Низкая эффективность, а часто неприемлемость традиционных средств общения в большинстве случаев вызвана тем, что в них не учитываются важнейшие особенности процесса общения, направленного на удовлетворение реальных информационных потребностей пользователя (ИПП). Эти особенности, независимо от специфики решаемых пользователями задач, сводятся к следующим:

1. *Изменяемость.* Информационная потребность пользователя не может быть заранее четко определена в спецификациях на разработку системы общения, напротив, ИПП неизбежно изменяется в ходе разработки и эксплуатации системы.

2. *Несовпадение взглядов на мир.* Представления, имеющиеся у пользователя и системы о языке общения и проблемной области, относительно которой ведется общение, могут не совпадать. Исходя из этого, процесс общения должен предусматривать разъяснение смысла неизвестных терминов, обнаружение и устранение несовпадающих представлений, а также предупреждение ошибочных толкований, т.е. установление общих точек зрения на обсуждаемые в процессе общения сущности.

3. *Связность общения.* Процесс общения не может быть ограничен обменом изолированными парами «вопрос-ответ», так как в большинстве реальных случаев ИПП не может быть выражена в виде одного вопроса (предложения). Часто требуется определить ситуацию, в которой возникла ИПП, т.е. предпослать запросу на решение некоторой задачи контекст, в котором эту задачу необходимо решать. Кроме того, процесс удовлетворения ИПП - решение некоторой задачи, в большинстве реальных приложений требует взаимодействия, основанного на смешанной инициативе участников. Поэтому процесс общения должен иметь сложную, разветвленную структуру и состоять из обмена связанными высказываниями.

4. *«Неправильность» высказываний пользователя.* Для выражения ИПП пользователь может применить как «правильные» предложения, т.е. такие, которые будут однозначно поняты и верно обработаны системой, так и «неправильные». Неправильности могут быть вызваны, во-первых, тем, что пользователь обычно не в состоянии учесть все ограничения системы общения в части ее возможностей и знаний, во-вторых, использованием умолчаний, характерных для естественного общения и допускающих неоднозначное толкование высказываний, и, в-третьих, отклонением предложений от грамматической нормы.

Недостатки традиционных средств общения потребовали создания средств нового поколения, которые должны быть способны настраиваться на ИПП и адаптироваться к их изменению, представлять и объяснять свою точку зрения на проблемную область, а также учитывать точку зрения пользователя, поддерживать связный диалог и уметь обрабатывать «неправильные» высказывания. Разработка этих средств ведется в настоящее время по двум основным направлениям. *Первое направление,* развиваемое преимущественно специалистами по системам обработки данных, заключается в повышении уровня и увеличении непроцедурности формализованных языков общения. Типичными представителями таких языков являются, например, APL, NOMAD, MAPPER. *Второе направление* развивается в рамках искусственного интеллекта и предполагает использование конечными

пользователями для взаимодействия с компьютером естественного языка, семантически и прагматически ограниченного проблемной областью, относительно которой ведется общение. Рассматриваемые далее ЕЯ-системы разработаны в рамках второго направления.

Традиционные средства общения не позволяют обеспечить взаимодействие конечных пользователей с компьютером. Чтобы быть полноправным участником общения, ЕЯ-система должна выполнять некоторые обязательные функции. К этим функциям относятся:

- ведение диалога - определение его структуры и ранга роли, которую система и пользователь выполняют на текущем шаге диалога;
- понимание - преобразование поступающих от пользователя высказываний на естественном языке в высказывания на языке внутреннего представления;
- обработка высказываний - формирование или определение заданий на решение задач или подзадач на данном шаге диалога;
- генерация - формирование выходных высказываний на ЕЯ.

Приведенные функции имеют обобщенный характер. Поэтому необходимо подчеркнуть, что при реализации конкретных ЕЯ-систем суть этих функций может в значительной степени варьироваться. В соответствии с выделенными функциями обобщенная схема ЕЯ-системы (рис.1) может быть представлена в виде трех компонентов: диалоговый; компонент понимания высказываний; компонент генерации высказываний.

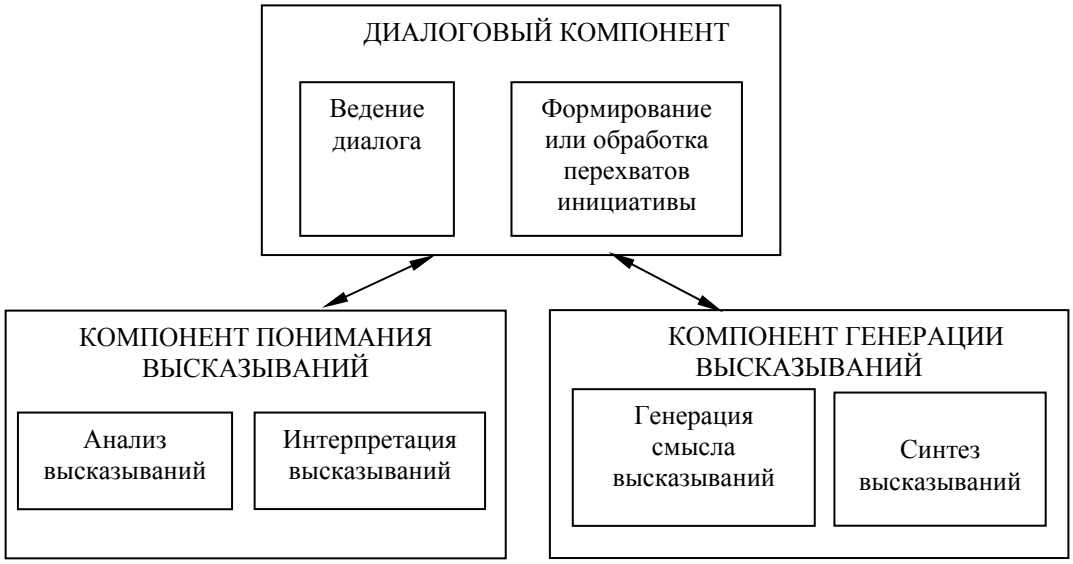


Рис. 1. Обобщенная схема ЕЯ-системы

Диалоговый компонент

К основным задачам диалогового компонента относятся:

- ведение диалога;
- формирование или обработка перехватов инициативы.

Ведение диалога состоит в том, чтобы обеспечивать целесообразные (т.е. способствующие достижению конечных целей пользователя) действия системы на текущем шаге диалога. В связи с тем, что возможности существующих ЕЯ-систем не позволяют им самостоятельно формировать целесообразное поведение, в систему обычно вводится

информация, определяющая общую и тематическую структуру диалога. По структуре и текущему состоянию диалога *диалоговый компонент* формирует (если инициатива принадлежит системе) или определяет (если инициатива принадлежит пользователю) задание, выполняемое системой на текущем шаге (например, генерация вопроса, понимание ответа и его обработка, генерация утверждения и т. п.).

Ведение диалога выполняется по одной из двух схем:

- диалог ведет пользователь;
- диалог ведет система.

В первом случае инициатива в основном (за исключением сообщений об ошибках) находится у пользователя, а система только реагирует на его требования, определяя по виду требования тип задания. Разбиение задачи на подзадачи и распределение ролей осуществляет пользователь, т.е. для системы весь диалог сводится к выработке реакции на текущие высказывания пользователя.

Можно сказать, что в этом варианте работы функции диалогового компонента практически вырождены. Во втором случае инициатива в основном принадлежит системе. Система ведет диалог в соответствии с имеющимися у нее представлениями о структуре диалога (т.е. о разбиении задач на подзадачи и о том, кто из участников, когда и какую подзадачу решает) и о способе обмена высказываниями. Если роли участников неизменны, однозначны и предопределены заранее, то структуру диалога называют *жесткой.* В простейшем случае такая структура диалога сводится к двум взаимосвязанным высказываниям участников (вопрос-ответ) с указанием участника, владеющего инициативой. Развитием жесткой структуры является *альтернативная структура,* которая задает множество возможных (но заранее предписанных) направлений течения диалога. Выбор одного из возможных направлений осуществляет пассивный участник. Если роли участников общения распределяются в ходе общения, то структуру диалога *называют гибкой.* Гибкие структуры подразделяются по степени свободы выбора момента перехвата (предопределенные моменты, произвольные моменты) и по способу перехвата (предопределенный способ перехвата, произвольный способ).

Вторая задача диалогового компонента вызвана тем, что реакции одного участника могут не соответствовать ожиданиям другого. В зависимости от того, кто осуществляет перехват инициативы, система либо формирует перехват, либо обрабатывает его. Формирование происходит в тех случаях, когда система определяет, что текущая ситуация не соответствует ситуации, предусмотренной структурой диалога. Если же перехват инициативы осуществляет пользователь, то задача системы - обработать его, т. е. распознать наличие перехвата инициативы, определить новую тему (цель), на которую перешел пользователь, и перейти на структуру диалога, соответствующую новой теме.

Компонент понимания высказываний

Компонент понимания высказываний предназначен для выделения смысла входного высказывания и выражения этого смысла на внутреннем языке системы. Под *смыслом высказывания* обычно понимается вся та семантика - прагматическая информация, которую

пользователь хотел передать системе. Внутреннее представление смысла должно содержать, по крайней мере, следующую информацию:

- сущности проблемной области, вовлекаемые в зону рассмотрения данным высказыванием;
- свойства и отношения, приписанные этим сущностям;
- коммуникативные измерения говорящего, выраженные в данном высказывании.

Выявление смысла высказывания в общем случае требует его рассмотрения в контексте всего диалога.

Традиционно задача понимания высказываний подразделяется на два этапа: *анализ и интерпретацию.* На этапе анализа выделяются описания сущностей, упомянутых во входном высказывании, выявляются свойства этих сущностей и отношения между ними. Диалог обычно выполняется отдельным блоком-анализатором, служащим ядром компонента понимания. Анализаторы, разрабатываемые для ЕЯ-систем, различаются по ряду параметров (табл. 1):

- тип анализируемых предложений;
- выделяемые описания сущностей;
- глубина проникновения в смысл;
- используемые для анализа средства.

Таблица 1

Общая характеристика анализаторов ЕЯ- систем

<table>
<tr><th colspan="3">Параметр</th><th>Возможные значения</th></tr>
<tr><td colspan="3">Типы анализируемых предложений</td><td>Повествовательные, вопросительные, отрицательные, полные, неполные, простые, сложные, распространенные, нераспространенные и др.</td></tr>
<tr><td rowspan="5">Выделяемые описания сущностей</td><td colspan="2">Понятия</td><td>Конкретные (индивидуальные), абстрактные, метапонятия</td></tr>
<tr><td rowspan="3">Отношения</td><td>Предикаты</td><td>Вспомогательные, предикаты-состояния, предикаты-действия, функциональные и др.</td></tr>
<tr><td>Кванторы</td><td>Отсутствие кванторов, кванторы существования всеобщности, кванторы отрицания</td></tr>
<tr><td>Модальности</td><td>Отсутствие модальностей, объективная и субъективная модальности</td></tr>
<tr><td colspan="2">Пресуппозиции</td><td>Отсутствие пресуппозиций, экзистенциональные, семантические, прагматические</td></tr>
<tr><td colspan="3">Глубина проникновения в смысл</td><td>Множество ключевых слов, имя события и описания участников события (их роли и характеристики), сценарий с отсылками к связанным подсценариям, пространственно-временное или причинное представление ситуации</td></tr>
<tr><td colspan="3">Используемые средства</td><td>Морфологический, синтаксический, семантический, прагматический или проблемный анализ</td></tr>
</table>

При рассмотрении таблицы 1 следует учитывать, что тот или иной параметр в случае конкретного анализатора может принимать одно или несколько из указанных в таблице

значений.

Интерпретация заключается в отображении входного высказывания на знания системы. Основными задачами данного этапа являются:

- буквальная интерпретация высказывания в контексте диалога;
- интерпретация высказывания на намерения говорящего.

Буквальная интерпретация состоит в том, чтобы, учитывая контекст диалога, идентифицировать образы тех сущностей области интерпретации, которые имел в виду говорящий. В качестве области интерпретации могут использоваться:

- проблемная область;
- область языка общения (если высказывание пользователя касается языка общения);
- область системы (если пользователь интересуется возможностями и состояниями системы. Интерпретация на эту область особенно важна при возникновении «непонимания» между пользователем и системой);
- область пользователя (если высказывание содержит сведения о знаниях или намерениях пользователя);
- область дискурса (если в высказывании содержатся ссылки на предыдущие или последующие высказывания).

Вторая задача интерпретации состоит в том, чтобы, применяя имеющиеся у системы методы вывода, определить, как обрабатываемое высказывание соотносится с целями и планами участников общения. Строго говоря, данная задача решается совместно диалоговым компонентом и компонентом понимания высказываний. Как отмечалось, одной из функций диалогового компонента является определение отношения входного высказывания к текущей цели. При выполнении этой функции текущая цель (т.е. ее описание в форме внутреннего представления) передается компоненту понимания, который пытается интерпретировать на нее входное высказывание. Решение данной задачи в общем случае представляет собой чрезвычайно сложную проблему. Положение усугубляется тем, что одно и то же высказывание может использоваться для достижения целей, относящихся к различным областям.

Компонент генерации высказываний

Компонент генерации высказываний решает в соответствии с результатами, полученными остальными компонентами системы, две основные задачи: генерация смысла, т. е. определение типа и смысла выходного высказывания системы во внутреннем представлении, и синтез высказывания, т.е. преобразование смысла в высказывание на естественном языке.

Первая из указанных задач является сложной и мало изученной. Тип высказывания зависит от состояния системы и результатов, полученных предыдущими компонентами. Так, если система должна генерировать ответ на вопрос, то необходимо определить по состоянию системы, будет ли ответ прямой или косвенный. Прямой ответ (т. е. по существу заданного вопроса) может быть дан только в том случае, если обработка вопроса завершилась успешно.

Если в процессе обработки вопроса возникают какие-то затруднения, то более уместным может быть косвенный ответ, раскрывающий суть возникших затруднений и объясняющий невозможность прямого ответа.

В общем случае при решении задачи формирования смысла выходного высказывания необходимо учитывать прагматический аспект, т. е. цели участников общения. Однако в большинстве существующих систем данная задача решается с помощью достаточно простых подходов. В промышленных системах общения генерация смысла обычно заключается в редактировании значений атрибутов и (или) выборе шаблона ответа. В экспериментальных системах для выражения смысла генерируется полное семантическое представление, включающее одно или несколько связанных событий (понятий) с одним или несколькими исполнителями на каждую роль.

Вторая задача компонента генерации высказываний состоит в синтезе естественно-языкового выражения, соответствующего внутреннему представлению выходного высказывания. Данная задача подразделяется на этапы семантического, синтаксического и морфологического синтеза. Сложность задачи синтеза определяется требованиями к естественности и выразительной мощности выходных высказываний. Под *естественностью* обычно понимается степень близости к естественному языку, т.е. наличие таких свойств, как синонимия и омонимия слов и словосочетаний, свободный порядок слов и др. Под *выразительной мощностью* понимается возможность выразить разнообразные понятия, отношения, кванторы, процедуры и т.п. Естественность и выразительность выходных высказываний в существующих системах могут быть различными. Например, высказывания могут фактически не синтезироваться, а выбираться из заранее подготовленного списка, либо имеется шаблон ответа, в который подставляются некоторые слова, представляющие собой значения искомых атрибутов, при этом может использоваться морфологический синтез. Большая естественность достигается, если выходное высказывание формируется из семантического представления события (или понятия) с применением морфологии, синтаксиса (для определения порядка и формы слов) и семантики (для выбора лексем и обеспечения семантической сочетаемости слов в синтезируемом высказывании).

Знания ЕЯ-систем

Для понимания принципов построения ЕЯ-систем важен также вопрос об используемых в системе знаниях, поскольку именно знания, представленные в различных формах, являются той базой, на которой осуществляется решение рассмотренных выше задач.

Знания ЕЯ-систем можно классифицировать по различным основаниям. Будем придерживаться классификации, представленной на рисунке 2. На верхнем уровне выделяются:

- собственно знания;
- способ представления знаний.

Собственно знания классифицируются по областям и по видам знаний. Наиболее существенными с точки зрения процесса ЕЯ-общения являются следующие области знаний:

- проблемная область;
- область языка;
- область системы;
- область пользователя;
- область диалога (дискурса).

Разнообразие областей определяет множество возможных интерпретаций входных высказываний. К основным видам знаний относятся факты (фактические знания) и реляционные знания. Факты представляют собой возможные знания о сущностях, составляющих некоторую область знаний. Операционные знания составляет информация о способах изменения фактических знаний. Иначе говоря, эти знания задают процедуры преобразования. Часто для обозначения этих знаний используется термин «процедурные знания», однако следует иметь в виду, что операционные знания могут быть представлены как в процедурной, так и в декларативной форме.

Способ представления знаний включает два аспекта: способ организации знании и модель представления. Способы организации знаний различаются по уровням представления и уровням детальности. По уровням представления выделяют знания нулевого уровня (конкретные и абстрактные знания) и знания более высоких уровней (метазнания). Первый уровень составляют знания о том, как в системе представлены знания нулевого уровня. Число уровней представления может быть продолжено. Разделение знаний по уровням представления обеспечивает возможность гибкой настройки и адаптации ЕЯ-системы.

Организация знаний по уровням детальности позволяет рассматривать знания с различной степенью подробности. Количество уровней детальности зависит от специфики решаемых задач, количества знаний и способа их представления. Обычно выделяется не менее трех уровней, отражающих общую организацию знаний, логическую и физическую организацию частных структур знаний. Введение нескольких уровней детальности обеспечивает дополнительную гибкость системы, так как изолирует изменения одного уровня от других.

Модели представления знаний обычно подразделяются на логические и эвристические, на декларативные и процедурные.

Декларативная модель основывается на предположении, что проблема представления некоей предметной области решается независимо от того, как эти знания потом будут использоваться. Поэтому модель как бы состоит из двух частей: статических описательных структур знаний и механизма вывода, оперирующего этими структурами и практически независимого от их содержательного наполнения. При этом в какой-то степени оказываются раздельными синтаксические и семантические аспекты знания, что является определенным достоинством указанных форм представления из-за возможности достижения их определенной универсальности.

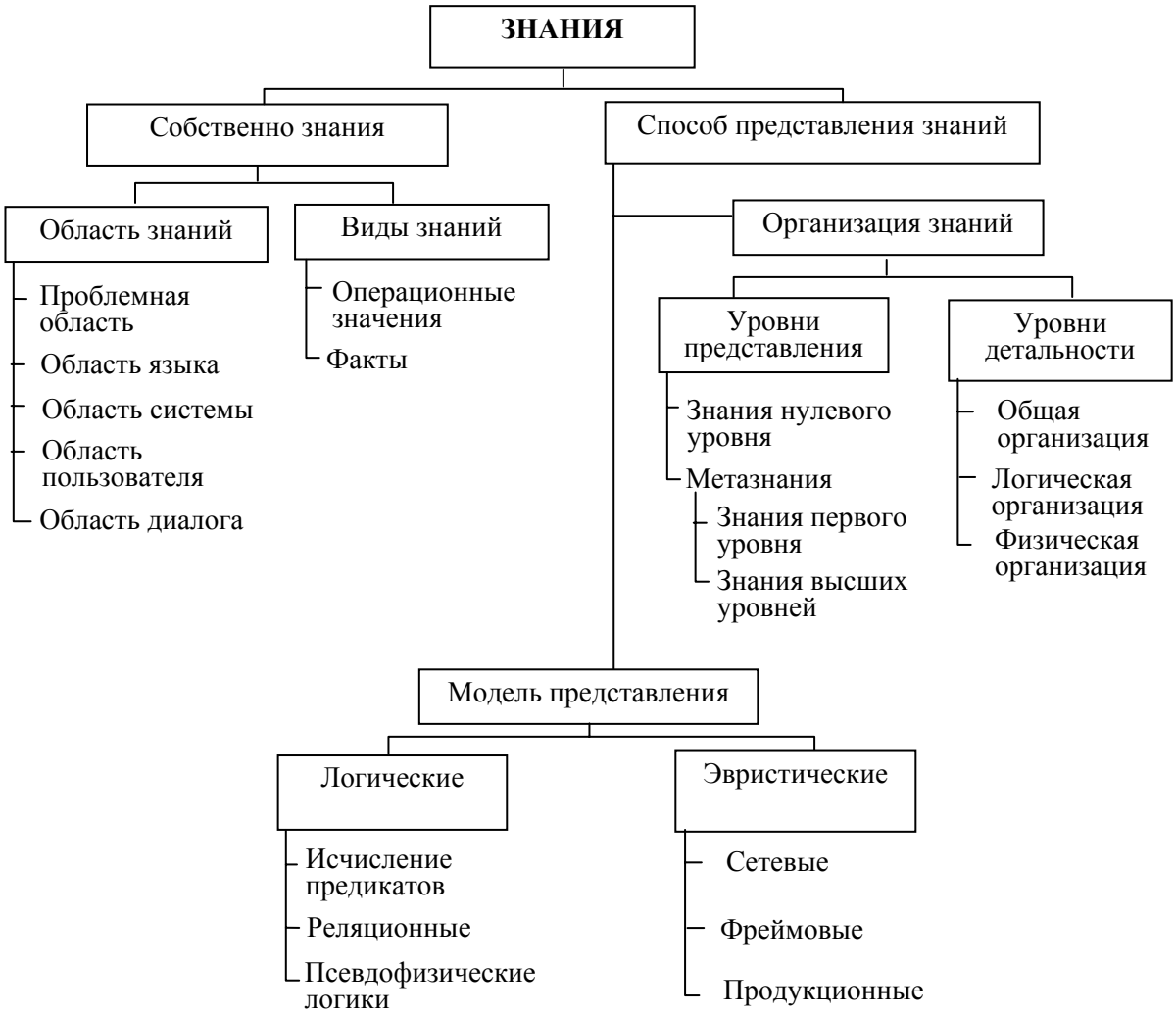


Рис.2. Классификация знаний ЕЯ-систем

В декларативных моделях не содержатся в явном виде описания выполняемых процедур. Эти модели представляют собой обычно множество утверждений. Предметная область представляется в виде синтаксического описания ее состояния (по возможности полного). Вывод решений основывается в основном на процедурах поиска в пространстве состояний.

В процедурном представлении знания содержатся в процедурах - небольших программках, которые определяют, как выполнять специфичные действия (как поступать в специфичных ситуациях). При этом можно не описывать все возможные состояния среды или объекта для реализации вывода. Достаточно хранить некоторые начальные состояния и процедуры, генерирующие необходимые описания ситуаций и действий.

Семантика непосредственно заложена в описание элементов базы знаний, за счет чего повышается эффективность поиска решений. Статическая база знаний мала по сравнению с процедурной частью. Она содержит так называемые "утверждения", которые приемлемы в данный момент, но могут быть изменены или удалены в любой момент. Общие знания и правила вывода представлены в виде специальных целенаправленных процедур,

активизирующихся по мере надобности. Процедуры могут активизировать друг друга, их выполнение может прерываться, а затем возобновляться. Возможно использование процедур - "демонов", активизирующихся при выполнении операций введения, изменения или удаления данных.

Средством повышения эффективности генерации вывода в процедурных моделях является добавление в систему знаний о применении, т.е. знаний о том, каким образом использовать накопленные знания для решения конкретной задачи. Эти знания, как правило, тоже представляются в процедурной форме.

Главное преимущество процедурных моделей представления знаний заключается в большей эффективности механизмов вывода за счет введения дополнительных знаний о применении, что, однако, снижает их общность. Другое важное преимущество заключено в выразительной силе. Эти системы способны смоделировать практически любую модель представления знаний. Выразительная сила процедурных систем проявляется в расширенной системе выводов, реализуемых в них. Большинство расширенных форм выводов может быть охарактеризовано понятием "предположение об отсутствии" и сводится к схеме: "Если А (предварительное условие) - истинно и нет доказательств против В, то предложить В". Подобные правила вывода оказываются полезными в основном в двух случаях:

1. Неполнота знаний. Если в системе представления отдельные факты не представлены или невыводимы, правила вывода позволяют гипотетически признавать их верными при условии, что в системе нет или в ней невыводимы доказательства противного.

2. Вывод в условиях ограниченности ресурсов. Из-за ограниченности ресурсов процессы вывода не могут завершиться, а должны быть оставлены для получения результатов. В этом случае правила определяют дальнейшие действия системы.

Системы представления, содержащие подобные правила, оказываются немонотонными, т.е. добавление новых утверждений может запретить генерацию вывода, который первоначально мог быть получен. Добавление новых фактов может привести к возникновению противоречий. В некоторых системах кроме самих утверждений содержатся также записи причин, по которым были приняты эти утверждения. При добавлении новых фактов осуществляется проверка того, сохраняются ли справедливость утверждений и соответствие причинам.

Можно выделить ряд общих для всех систем представления знаний (СПЗ) черт [7, 8]: все СПЗ имеют дело с двумя мирами - представляемым и представляющим. Вместе они образуют систему для представления. Существует также ряд общих для всех СПЗ проблем. К ним можно отнести, в частности, проблемы: приобретения новых знаний и их взаимодействие с уже существующими, организации ассоциативных связей, неоднозначности и выбора семантических примитивов, явности знаний и доступности, выбора соотношения декларативной и процедурной составляющих представления, что влияет на экономичность системы, полноту, легкость кодировки и понимания.

Рассмотрим различные формы моделей представления знаний.

Продукционные модели представляют собой набор правил в виде "условие - действие",

где условия являются утверждениями о содержимом БД (фактов), а действия есть некоторые процедуры, которые могут модифицировать содержимое БД. Продукционные модели из-за модульного представления знаний, легкого расширения и модификации нашли широкое применение в ЭС и ЕЯ-системах.

Другая важная схема представления знаний - семантические сети, представляющие собой направленный граф, в котором вершинам ставятся в соответствие конкретные объекты, а дугам, их связывающим, - семантические отношения между этими объектами. Семантические сети могут использоваться как для декларативных, так и для процедурных знаний.

Перспективной формой представления знаний являются фреймы, которые быстро завоевали популярность у разработчиков систем ИИ благодаря своей универсальности и гибкости.

Принципиальным методом для логического представления знаний является использование логики предикатов первого порядка (исчисление предикатов). При таком подходе знания о некоторой предметной области могут рассматриваться как совокупность логических формул. Изменения в модели представления знаний происходят в результате добавления или удаления логических формул.

В редукционных моделях осуществляется декомпозиция исходной задачи на ряд подзадач, решая которые последовательно определяют решение поставленной задачи.

Логические представления легки для понимания и располагают правилами вывода, необходимыми для операций над ними. Однако в логических моделях представление знаний отношения между элементами знаний выражаются ограниченным набором средств используемой формальной системы, что не позволяет в полной мере отразить специфику предметной области. Недостатком логического представления является также тенденция потреблять большие объемы памяти ЭВМ.

Ряд понятий человеческих знаний оказывается трудно, а иногда и невозможно описать количественно, используя детерминированные или стохастические методы. Трудности возникают при создании моделей не полностью определенных, неточных, нечетких знаний. Это связано с тем, что человеческому мышлению присуща лингвистическая неопределенность; знания и понятия, которыми оперирует человек, часто имеют качественную природу, они ситуативны, бывают неполными. Для формализации знаний такого типа используется аппарат теории нечетких множеств, создание которого связано с именем известного американского ученого Л. Заде.

Неточность, неопределенность или неполнота, заключенные в смысловых значениях или выводах, присущи естественным языкам с их сложной структурой и многообразием понятий. Различают несколько типов неопределенности в прикладных системах ИИ. Первый связан с ненадежностью исходной информации - неточность измерений, неопределенность понятий и терминов, неуверенностью экспертов в своих заключениях.

Второй - обусловлен нечеткостью языка представления правил, например в экспертных системах. Неопределенность возникает также, когда вывод базируется на неполной

информации, т.е. нечетких посылках. Еще один тип неопределенности может появляться при агрегации правил, исходящих от разных источников знаний или от разных экспертов. Эти правила могут быть противоречивыми или избыточными.

Различие между декларативным и процедурным представлением можно выразить как различие между «знать что» и «знать как». Каждое представление имеет свои достоинства и недостатки. В заключение необходимо отметить, что деление моделей представления знаний на декларативные и процедурные весьма условно, так как стремление наиболее полно использовать достоинства обоих видов представления знаний привело к разработке смешанных представлений, т.е. декларативных представлений с присоединенными процедурами (например, фреймовые модели и модели, использующие расширенные семантические сети).

Оптимальное решение задачи выбора: первый прототип реализуется на специализированных средствах, и в случае достаточной эффективности этих средств на них могут быть написаны действующий прототип, и даже промышленная система. Однако в большинстве случаев прототип следует "переписать" на традиционных средствах.

Программные средства создания ЕЯ-систем

Рассмотрим наиболее известные и широко применяемые программные средства искусственного интеллекта.

Язык программирования Лисп. Самое популярное средство для программирования систем ИИ. Создан в 60-х годах американским ученым Дж. Маккарти и его учениками. Наиболее известными диалектами этого языка являются InterLisp, QLisp, CommonLisp. На языке Лисп написаны многие ЭС (Mycin, Internist, Kee), системы естественно-языкового общения (Margie, Shrdlu, Дилос), интеллектуальные ОС (Flex).

Популярность Лиспа объясняется тем, что он с помощью довольно простых конструкций позволяет писать сложные и изящные системы обработки символьной информации. Правда все Лисп-системы имеют низкую вычислительную эффективность.

Существенной особенностью языка Лисп является то, что здесь "данные" и "программы" внешне ничем не отличаются друг от друга. Это дает возможность писать на Лиспе программы, манипулирующие не только "данными", но и "программами". Именно данное свойство позволяет Лиспу стать изящным средством программирования систем ИИ.

Язык программирования FRL (Frame Representation Language). Относится к классу фрейм-ориентированных языков. Фрейм в FRL - это совокупность поименованных, ассоциативных списков, содержащая до пяти уровней подструктур. Подструктурами фреймов могут быть слоты, аспекты, данные, комментарии и сообщения.

Важным свойством FRL является наличие в нем встроенного механизма "наследования свойств". Т.е. все понятия предметной области в БЗ организовываются в виде иерархической классификационной системы, где каждое общее (родовое) понятие связывается с более конкретным (видом). Применяется механизм наследования свойств.

На сегодняшний день большинство FRL-систем написаны на Лиспе.

Язык программирования Пролог. Наиболее известные Пролог - системы: MProlog,

CProlog, Prolog-2, Arity Prolog, Turbo Prolog, Strawberry Prolog. Пролог все чаще в последнее время стал привлекаться к разработке ЭС. Математической основой этого языка являются исчисление предикатов преимущественно первого порядка, метод резолюций Робинсона, теория рекурсивных функций. За счет наличия большого набора встроенных предикатов язык Пролог можно отнести к универсальным языкам программирования и даже к языкам системного программирования. Важнейшей особенностью языка является наличие реляционной базы данных.

Язык программирования OPS. Язык относится к числу продукционных. Являясь универсальным языком, он, в первую очередь, предназначен для разработки систем ИИ, и, в частности экспертных систем. Архитектура языка OPS типична для продукционных систем: база правил, рабочая память и механизм вывода. Отличительные черты семейства языков OPS: программное управление стратегией вывода решений, развитая структура данных и принципиальная эффективность реализации.

Язык программирования Рефал (рекурсивных функций алгоритмический язык). Это машинно-независимый алгоритмический язык, ориентированный на так называемые "символьные преобразования": перевод с одного языка на другой, алгебраические выкладки и т.п. Рефал - универсальный метаязык для преобразования объектов языковой природы. Важнейшим приложением Рефала является его использование в качестве метаязыка для построения системных макрокоманд и специализированных языков. В качестве конкретных областей применения Рефала следует отметить, в частности, создание специализированных языков общения с ЭВМ, автоматическую генерацию программ, перенос программ на языки высокого уровня и их адаптацию при переходе от одного типа ЭВМ к другому.

Проблема выбора программных инструментальных средств вызывает бурные дискуссии между сторонниками специализированных языков ИИ и традиционных языков высокого уровня. Над решением данной проблемы работает целый ряд компаний, специализирующихся на ИИ и коммерческих ЭС, а также большинство крупных фирм по производству ЭВМ.

Лекция 2. Основные классы ЕЯ-систем

В лекции 1 были определены и в общем виде рассмотрены основные функциональные компоненты ЕЯ-систем: ведение диалога, понимание высказываний и генерация высказываний. В зависимости от назначения прикладных систем, в состав которых входят конкретные реализации ЕЯ-систем, задачи, решаемые отдельными функциональными компонентами (как по постановке, так и по методам решения), могут в значительной степени варьироваться. Исходя из этого, а также с учетом истории развития ЕЯ-систем, различают следующие основные классы систем общения:

- интеллектуальные вопрос - ответные системы;
- системы общения с базами данных;
- диалоговые системы решения задач;
- системы обработки связных текстов;
- системы машинного перевода.

Исторически ЕЯ-системы происходят от информационно-поисковых систем, с одной стороны, и систем машинного перевода − с другой. Поэтому на начальном этапе ЕЯ-системы представляли собой макеты информационно-поисковых систем, демонстрирующие принципиальную возможность ввода данных (фактов) и обработки запросов на естественном языке. Такие системы часто назывались интеллектуальными вопрос-ответными системами. Название можно, по-видимому, объяснить стремлением их разработчиков подчеркнуть, что в отличие от обычных информационно-поисковых систем и систем машинного перевода того времени в данных системах широко используются концепции, выработанные в исследованиях по искусственному интеллекту.

Основное внимание при разработке интеллектуальных вопрос-ответных систем уделялось не столько возможностям их практического использования в реальных задачах, сколько развитию моделей и методов, позволяющих осуществлять перевод ЕЯ-высказываний, относящихся к узким и заранее фиксированным проблемным областям, в формальное представление, а также обратный перевод. Накопленный опыт разработки интеллектуальных вопрос-ответных систем позволил, с одной стороны, углубить понимание процесса ЕЯ-общения и, следовательно, поставить новые проблемы (в том числе и специфичные для общения в различных классах проблемных областей), требующие дальнейшей проработки, а с другой - оценить перспективы практического применения ЕЯ-систем.

Первые предпосылки для практического использования ЕЯ-систем создало появление баз данных (БД). В связи с этим возникла проблема обеспечения доступа к информации, хранящейся в БД, широкому классу неподготовленных конечных пользователей, к которым относят специалистов в той или иной предметной области, как правило, не обладающих знаниями о логической структуре БД, о системе представления информации в БД и не умеющих пользоваться формализованными языками запросов. Для решения этой проблемы стали создаваться системы общения с базами данных, основная задача которых (в простейшем случае) заключается в выполнении перевода запросов неподготовленных конечных пользователей с ЕЯ на формализованные языки запросов к БД.

Диалоговые системы решения задачи в отличие от систем общения с БД берут на себя не только функции ЕЯ-доступа к БД, но и функции интеллектуального монитора, обеспечивающего решение заранее определенных классов задач (например, планирование путешествий, боевых операций, составление контрактов и т. п.). В этом случае разбиение задач на подзадачи и распределение ролей между участниками, т. е. определение, кто из участников (пользователь или система) решает ту или иную подзадачу, осуществляется не пользователем (как в случае применения систем общения с БД), а диалоговой системой. Решение подзадач, «порученных» системе, может осуществляться как на основе использования собственных знаний и механизмов вывода, так и в результате обращения к прикладным программам и пакетам, не входящим в состав ЕЯ-системы. Основным направлением практического использования ЕЯ-систем данного класса является реализация ЕЯ-общения с экспертными системами.

Возникновение последнего класса ЕЯ-систем — систем обработки связных текстов, обусловлено возрастанием объема хранимой в ЭВМ текстовой информации (газетные статьи, сообщения о различных событиях, патенты, авторские свидетельства и т. п.) и необходимостью извлечения из нее разнообразных сведений (например, о структуре некоторых объектов, о действующих лицах некоторых событий, о мотивах их поступков и т. д.).

Каждый из классов ЕЯ-систем обладает специфическими особенностями, которые хорошо заметны при рассмотрении характера задач, решаемых основными функциональными компонентами этих систем (рис. 3).

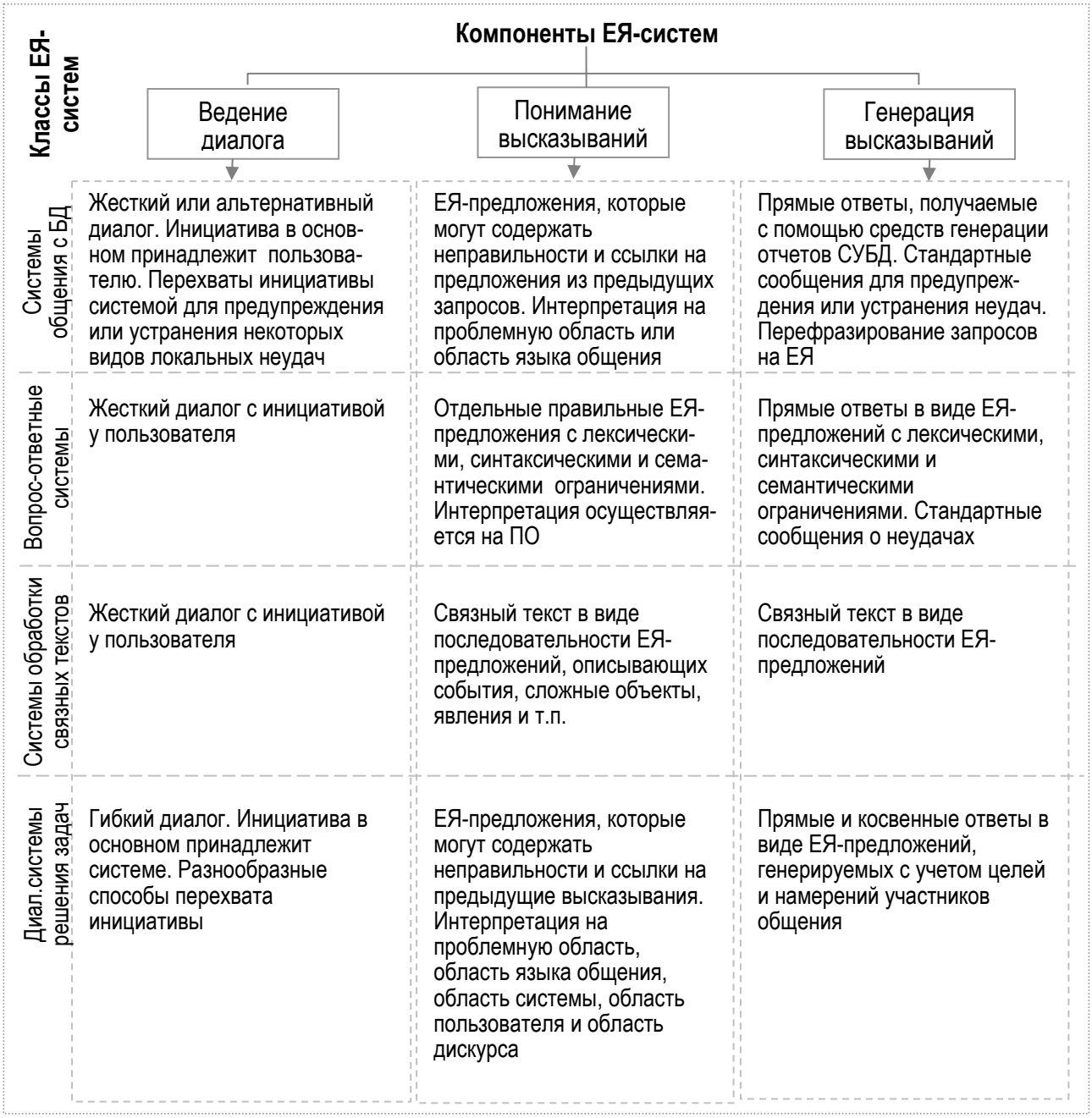


Рис.3. Сравнительная характеристика основных классов ЕЯ-систем

Приведенная выше классификация ЕЯ-систем охватывает лишь функционально полные системы, т. е. такие, в которых представлены все основные функциональные компоненты.

Однако помимо функционально полных систем ведется интенсивная разработка систем, которые можно назвать фрагментарными. Цель их создания - исследование или реализация новых методов решения достаточно узких задач (например таких, как анализ, интерпретация, определение целей пользователя и т. п.).

Благодаря модульности структуры ЕЯ-систем и, как правило, универсальному (т. е. не зависящему от специфики прикладных областей) характеру языка внутреннего представления, фрагментарные системы могут успешно использоваться в качестве отдельных функциональных блоков, встраиваемых (хотя бы на логическом уровне) в различные функционально полные ЕЯ-системы. Рассмотрим основные отличительные характеристики каждого класса ЕЯ-систем на примере существующих систем.

Интеллектуальные вопрос-ответные системы

При разработке интеллектуальных вопрос-ответных систем основное внимание уделяется языковому аспекту, т. е. максимальному приближению языка общения к литературному естественному языку. Наиболее значительной из отечественных систем данного класса является система ПОЭТ, созданная коллективом исследователей под руководством Э.В. Попова, во многом определившая применяемые в последующих системах методы анализа и генерации высказываний на русском языке.

Система ПОЭТ воспринимает вопросительные предложения русского языка с практически несущественными ограничениями на допустимые синтаксические конструкции и пунктуацию. Типичными примерами запросов, допускаемых системой ПОЭТ, могут служить «Сколько каменного угля перевезено железнодорожным транспортом в 1978 году?» или «Каков удельный вес перевозок железнодорожным транспортом в общем объеме перевозок всеми видами транспорта в 1975 году?»

Процесс понимания входных высказываний осуществляется в системе ПОЭТ по полной схеме: морфологический анализ, синтаксический анализ, семантический анализ и семантическая интерпретация (рис. 4). При этом последние три этапа выполняются в общем случае параллельно, за счет чего достигается коррекция неверных путей анализа, и в конечном счете сокращается время обработки запросов.

Все знания о языке общения разделяются в системе ПОЭТ на лингвистические и проблемные. Первые хранятся в различных зонах словаря (морфологической и синтактико-семантической), а вторые - в семантической сети. При этом в системе различаются абстрактная (описывающая общие понятия и категории) и конкретная (описывающая конкретные сущности) семантические сети. Описания базовых событий представлены в словаре системы в виде моделей управления. Выделение участников событий и определение выполняемых ими ролей осуществляются на основе метода фильтров. При этом активно используется как грамматическая, так и синтактико-семантическая информация.

На этапе семантического анализа синтаксическая структура входного высказывания, представленная в виде дерева зависимостей, преобразуется в семантический граф, состоящий из вершин-понятий, связанных друг с другом через вершины-события и характеристики. Каждая вершина семантического графа определяется каноническим представлением, а дуги

имеют глубинный смысл. Вся числовая и параметрическая информация выносится из графа в дополнительные таблицы. Там же указываются и временные соотношения между событиями.

На этапе интерпретации семантический граф запроса сопоставляется с семантической сетью. В результате происходит вычленение контекста, имеющего отношение к запросу, получение содержательной информации из конкретной сети, формирование обращений к базе данных за числовой информацией и получение способа обработки этой информации (суммирование, вычисление процента и т. п.). Семантический граф ответа вырабатывается на базе графа запроса путем внесения в него смысловой информации, полученной на этапе интерпретации.

Система ПОЭТ является ЕЯ-системой с генерацией ответов на русском языке. Формирование ответов выполняется следующим образом. По семантическому графу ответа строится дерево зависимостей. Затем каждой вершине приписывается морфологическая информация и определяется порядок слов. На этом заканчивается синтаксический синтез. На этапе морфологического синтеза по таблицам окончаний и морфологической информации, приписанной вершинам дерева зависимостей, осуществляется окончательная генерация поверхностной структуры ответа. Генерация полного ответа, например, «В 1978 году железнодорожный транспорт перевез NNN млн. тонн каменного угля», позволяет пользователю убедиться в правильности понимания системой заданного вопроса.

С помощью первых вопрос-ответных систем была показана принципиальная возможность получения ответов на ЕЯ-вопросы, относящиеся к ограниченным проблемным областям. Так, система ПОЭТ могла отвечать па вопросы о перевозках различных народнохозяйственных грузов. ДИСПУТ - об обслуживании контейнерных перевозок к морскому порту, LUNAR - о свойствах образцов лунных пород, LIFER - о дислокации и характеристиках судов военно-морских сил и т. п. Для большинства интеллектуальных вопрос-ответных систем была характерна жесткая структура диалога, при которой каждое высказывание пользователя воспринималось как очередной запрос (который, как правило, не мог быть связан с предыдущим). Система играла пассивную роль - она могла лишь отвечать на запросы и выдавать сообщения о неудачах, когда очередной запрос по каким-либо причинам не мог быть проанализирован или обработан. Обработка высказываний сводилась в большинстве случаев к вызову (в соответствии с типом запроса) одной из имеющихся в распоряжении системы специализированных программ и передаче ей в виде параметров условий поиска информации в БД имен сущностей, значения которых должны быть обработаны или выданы в качестве ответа, и т. п.

Первые эксперименты с интеллектуальными вопрос-ответными системами показали, что, несмотря на возможность понимания запросов на ЕЯ, данные системы налагают достаточно жесткие (и в общем случае трудновыполнимые) ограничения на процесс общения. Эти ограничения стимулировали дальнейшие исследования, направленные, в первую очередь, на повышение гибкости процесса общения.

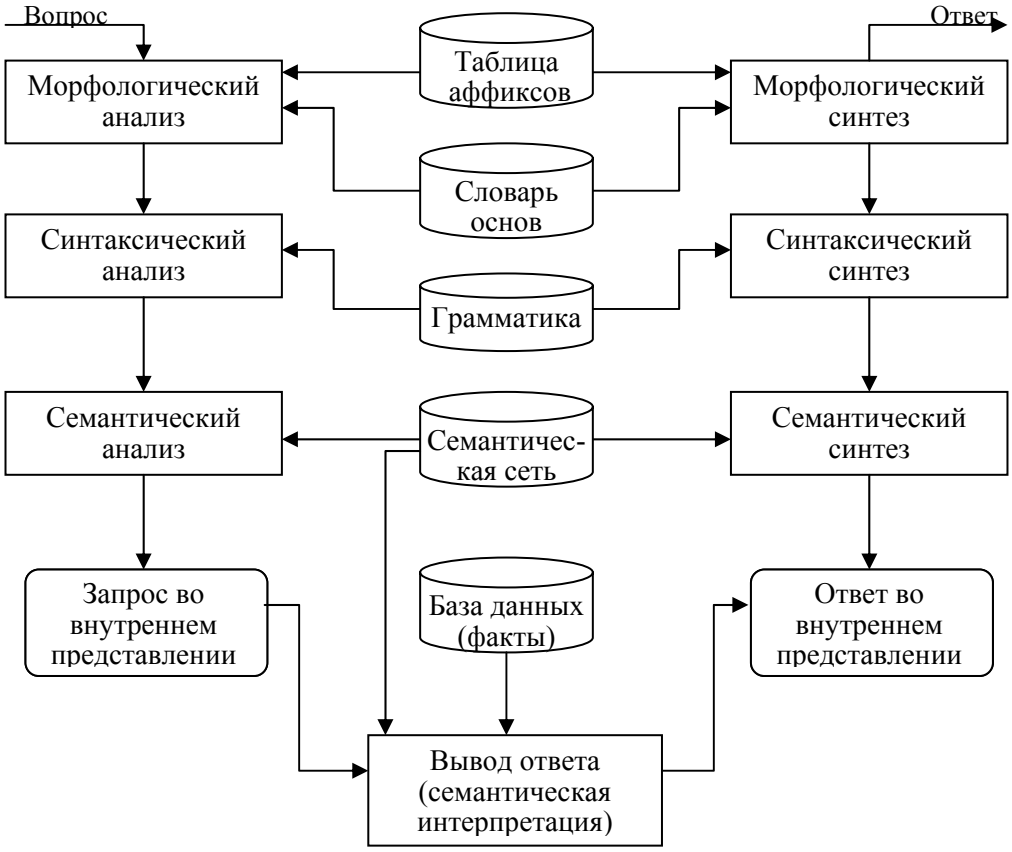


Рис. 4. Схема вопрос-ответной системы ПОЭТ

Системы общения с базами данных

В последнее время возрос интерес к ЕЯ-системам данного класса, что связано с усиливающейся тенденцией хранения информации в структурированных источниках данных. Концепция базы данных лежит в основе подавляющего большинства современных систем обработки данных. Для обеспечения взаимодействия с конечными пользователями системы управления базами данных (СУБД) предоставляют специальные формализованные языки. Однако, как правило, эти языки ориентированы на пользователей, обладающих специальными знаниями. В частности, они должны знать основные приемы программирования, синтаксис языка запросов, логическую структуру БД, термины, используемые в БД для обозначения сущностей предметной области, и т.п. ЕЯ-системы рассматриваемого класса предназначены для использования в качестве посредника (естественно-языкового интерфейса) между неподготовленными конечными пользователями (т. е. специалистами в прикладных проблемных областях, не обладающими указанными выше знаниями) и БД. Другими словами, они должны позволять получать информацию, хранящуюся в БД, по запросам, сформулированным на ЕЯ. На рисунке 5 показана упрощенная схема систем общения с БД.

В системах общения с БД общение ведется в форме связного диалога, т.е. ответы на вопросы пользователя выдаются с учетом его предыдущих вопросов и/или предыдущих ответов системы. Инициатива в диалоге в основном принадлежит пользователю. Перехват

инициативы допускается лишь для уточнения незнакомых системе слов и исправления орфографических ошибок. Язык общения является подмножеством естественного языка, семантически ограниченным предметной областью, отображаемой в БД. В высказываниях пользователя допускаются многие общепринятые синтаксические конструкции. Кроме того, допускаются определенные неправильности: орфографические ошибки, пропуск слов, ошибки в пунктуации, неправильное употребление строчных и прописных букв и ряд типичных диалоговых конструкций: эллипсис и анафорические ссылки. Ответы ЕЯ-системы строятся таким образом, чтобы обеспечить, насколько это возможно, «дружественность» общения.

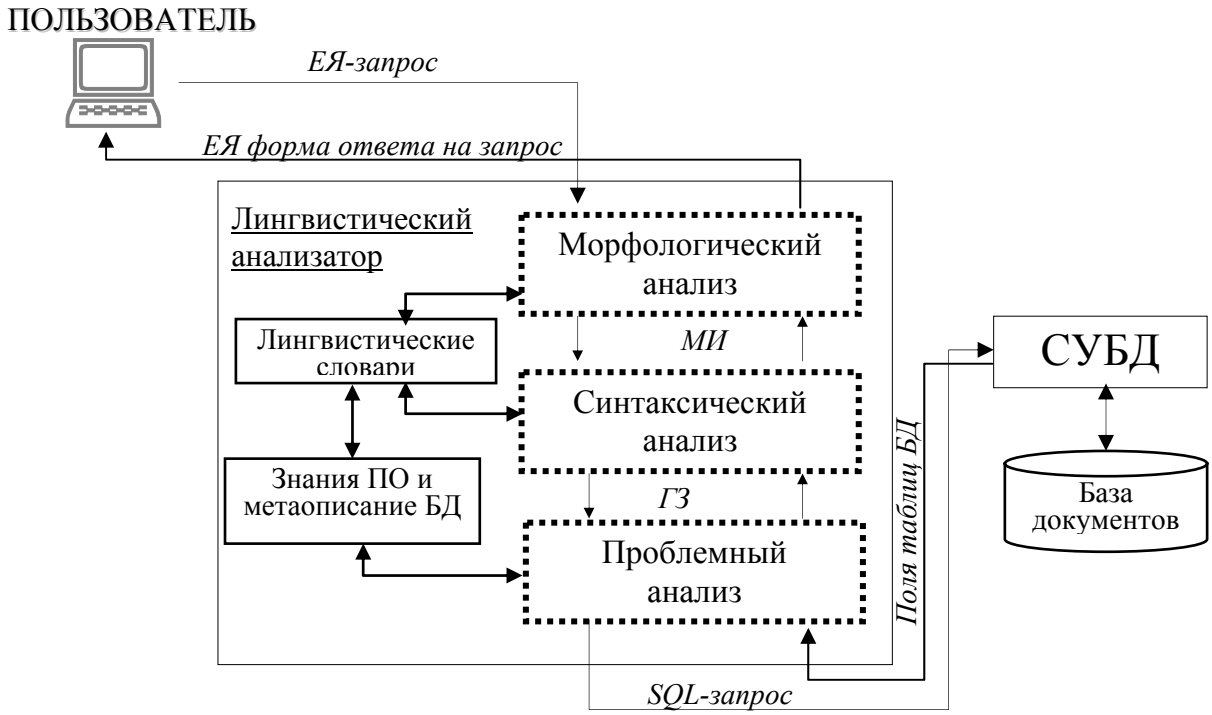


Рис. 5. Структурная схема систем общения с БД

С точки зрения возможностей практического использования, системы общения с БД значительно опережают остальные классы ЕЯ-систем. Существует несколько коммерческих и промышленных ЕЯ-систем общения с БД, которые успешно применяются на практике.

Несмотря на очевидные преимущества, предоставляемые пользователю ЕЯ-системами данного класса, они в настоящее время еще не получили широкого распространения. Это объясняется, по меньшей мере, следующими причинами: сравнительно малое количество коммерческих систем и их ориентация на общение преимущественно с реляционными БД; существующие ЕЯ-системы не позволяют задавать запросы, требующие сложной вычислительной и логической обработки, а также не позволяют управлять форматированием отчетов; значительная трудоемкость сопровождения и настройки ЕЯ-систем при их использовании для общения с большими изменяющимися базами данных. Остановимся подробнее на типах пользовательских интерфейсов к базе данных.

Типы пользовательских интерфейсов к базе данных

Под *пользовательским интерфейсом* понимается система средств, облегчающих поиск, получение, просмотр и обработку информации из БД. Естественно-языковой интерфейс (ЕЯИ) - разновидность пользовательского интерфейса, который принимает запросы на естественном языке, а также использует ЕЯ и для вывода информации (реакции системы на запрос пользователя).

В противоположность ЕЯ-интерфейсам, нетрадиционным с точки зрения распространенности, существуют другие виды пользовательских интерфейсов к БД, которые можно назвать традиционными. Среди них выделяют:

- интерфейсы с формальным языком запросов;
- интерфейсы с графическим построением запросов;
- интерфейсы, основанные на заполнении форм запросов.

В интерфейсах с формальным языком запросов пользователь, для того, чтобы правильно задать запрос, должен, во-первых, знать синтаксис языка запросов (например SQL), а во-вторых, представлять устройство конкретного структурированного источника данных (например, реляционную схему базы данных). При работе с этим типом интерфейсов пользователь должен обладать достаточно высокой квалификацией. Опыт показывает, что такой необходимой квалификацией обладают лишь специалисты, проектирующие и создающие информационные системы. Очевидно, что такие ЕЯ-интерфейсы обладают большей гибкостью - один и тот же запрос можно формулировать различными способами.

Средства графического построения запросов, которыми снабжаются многие "настольные" СУБД (например, MS Access, MS FoxPro), безусловно, обладают большим удобством - пользователь не должен держать в голове названия таблиц, полей и конструкции языка. Однако для работы с такими средствами необходим опыт и представление некоторых понятий, относящихся скорее к математике (например, термин связывания таблиц в реляционной алгебре), а не к предметной области, и иногда достаточно утомительные действия по заполнению форм. Так, в базе данных Microsoft Access для того, чтобы сформулировать выражение AVG(PERSONNEL.SALARY), эквивалентный ЕЯ-фразе "средняя зарплата", требуется около 15 нажатий мышью. Неподготовленный пользователь обычно пасует перед системами, требующими сложных действий. Как и в случае интерфейсов с формальным языком, пользователь должен представлять устройство базы данных.

Интерфейсы, основанные на заполнении форм запросов, являются более дружественными, по сравнению с формальными языками. Сама метафора формы и ее заполнения подразумевает, что пользователь сразу видит набор критериев и параметров поиска, а иногда и список возможных значений полей формы, что сводит к минимуму ошибки при вводе запроса. От предыдущего метода построения пользовательских интерфейсов данный отличается тем, что все необходимые запросы уже написаны разработчиком интерфейса, и пользователь, чтобы получить ответ, должен просто вставить недостающие значения. Так работают многие современные коммерческие приложения -

пользователю информация в системе доступна в виде нескольких типовых "срезов" информационного пространства. К недостатку систем, основанных на таком подходе, как и в предыдущем, также следует отнести необходимость наличия у пользователя опыта работы с подобными системами, а также необходимость создания форм, что требует дополнительных усилий программиста для создания интерфейса.

Поэтому преимущества ЕЯ-интерфейсов достаточно очевидны:

- минимальная предварительная подготовка пользователя. Естественный язык является наиболее привычным и удобным средством коммуникации, и именно в силу этого с ростом эффективности ЕЯ-систем, он будет вытеснять традиционные в данный момент;
- простота задания запросов на ЕЯ. Во многих случаях запрос на ЕЯ получается гораздо короче языка на формальном языке, поскольку ЕЯ-представление более емко, ведь в самой структуре языка содержится понятийная база, которую отражает структура источника данных;
- большая скорость создания произвольного запроса (отсутствует стадия формального задания запроса). Как правило, пользователь сразу может сформулировать корректное ЕЯ-представление запроса, поскольку такое представление является самым естественным для человека, тогда как построение запроса на формальном языке, даже с помощью вспомогательных средств, таит множество ошибок, зачастую исправить которые можно, только проанализировав результат запроса;
- более высокий уровень модели предметной области. Традиционные интерфейсы обычно не обладают моделью предметной области как таковой, и в лучшем случае скрывают от пользователя искусственные средства и особенности структуры, присущие конкретной БД (такие, как связи по идентификаторам между таблицами в реляционных базах данных или синтаксис XML).

Однако ЕЯ-интерфейсы не лишены недостатков:

- неоднозначность естественного языка приводит к множественности смыслов. Специфика естественного языка такова, что часто запрос может иметь несколько смыслов, о которых пользователь в момент задания запроса не предполагает. Формальные же языки лишены проблемы неоднозначности. Это свойство ЕЯ приводит к усложнению ЕЯ-интерфейсов и методов анализа, в противном случае ЕЯ-интерфейс получается слишком примитивным для реального использования;
- недостаточная надежность анализаторов ЕЯ-запросов может привести к неправильному пониманию. Современные ЕЯ-интерфейсы далеко не всегда позволяют диагностировать причины неудач понимания. Причины этих неудач могут быть как в лингвистической сфере, так и в концептуальной. Например, запрос к кадровой базе данных "Кто получает больше Иванова" может привести к непониманию, если ЕЯ-интерфейс не умеет распознавать вложенные запросы (а в данном случае надо сначала получить значение зарплаты Иванова, а затем

сравнить с ней зарплату сотрудников). Это случай лингвистической проблемы. Второй пример - "Как зовут жен сотрудников?" - может привести к неудаче понимания, если ЕЯ-интерфейс не поймет, что имя супруга/супруги - это реальный атрибут сотрудника, но отсутствующий в данной базе данных. В данном случае налицо будет концептуальная проблема - ЕЯ-интерфейс должен уметь отличать реальную предметную область, которую имеет в виду пользователь, задавая ЕЯ-запрос, от той ее части или трансформации, которая представлена в данном источнике данных;

- пользователь может иметь завышенные или заниженные ожидания от ЕЯ-интерфейса. Сравнительный анализ типов пользовательских интерфейсов (основанных на формах, с формальным языком запросов, графические) показывает, что в целях построения ЕЯ-интерфейсов превалирует желание максимально приблизить интерфейс к потребностям неподготовленного пользователя. Это несколько поднимает планку требований к дружественности и надежности ЕЯ-интерфейсов, поскольку пользователь, впервые столкнувшись с системой, понимающей естественный язык, слабо представляет, насколько интеллектуальна система. При этом ожидания к степени понимания ЕЯ может отличаться от реальных способностей системы в обе стороны - т.е. пользователь может спрашивать систему о том, чего она "не знает", а может "по привычке" использовать простейшие шаблонные формулировки запросов.

Для сравнения подходов к построению ЕЯ-интерфейсов введем метрику показателей, характеризующих качество ЕЯ-интерфейсов к структурированным источникам данных.

Критерии качества ЕЯ-интерфейсов

Рассмотрим такую качественную интегральную характеристику, как надежность. Под надежностью здесь понимается способность ЕЯ-интерфейса правильно понимать намерения пользователя по получению информации из источника, при условии, что пользователь корректно выразил потребности в виде ЕЯ-запроса. Надежность отражает правильность принципов, лежащих в методе ЕЯ-анализа, а также правильность (корректность) построения ЕЯ-интерфейса к конкретной БД.

Любой ЕЯ-интерфейс имеет некоторое пространство правильно понимаемых запросов. Чем больше это пространство, тем большей полнотой обладает ЕЯ-интерфейс. Полнота - характеристика, тесно связанная с гибкостью интерфейса. Поскольку пространство ЕЯ-запросов весьма неоднородно, следует говорить о различных типах запросов, т.е. групп запросов, имеющих сходное строение. Гибкость - показатель того, насколько разнообразные типы запросов может понимать ЕЯ-интерфейс. Речь в основном идет о так называемых "трудных" типах запросов, в числе которых - вложенные, эллипсис, анафорические.

Другой важной характеристикой является дружественность интерфейса, которую можно определить как меру того, насколько ЕЯ-интерфейс удобен в работе, насколько корректно он может сообщать о проблемах понимания, может ли он помогать в перефразировании непонятных системе запросов и т.д.

Все эти критерии можно объединить в схему, отражающую составляющие качества ЕЯИ (рис. 6).

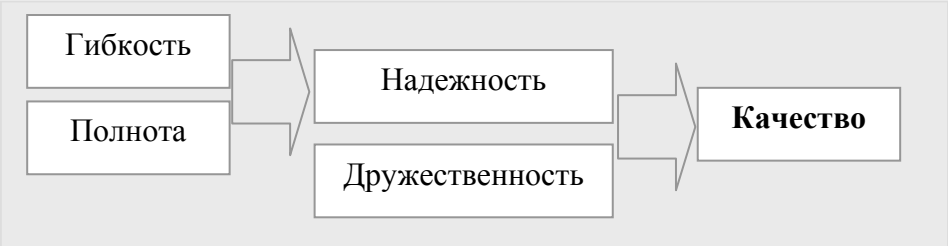


Рис. 6. Иерархия качественных характеристик ЕЯ-интерфейса

Подходы к анализу ЕЯ-запросов к БД

В данном разделе дается краткий обзор методов и подходов анализа ЕЯ применительно к теме построения ЕЯ-интерфейсов к структурированным источникам данных.

Подходов к решению задачи понимания естественно-языковых запросов несколько. Наиболее распространенными являются подходы, основанные на синтаксическом, семантическом анализах и шаблонах. Первый подход основан на использовании синтаксических конструкций. Синтаксическое представление запроса строится на основе подлежащего, сказуемого, прямого дополнения и т.п., которые определяются с помощью морфологических характеристик (часть речи, род, падеж, лицо и т.д.). Это представление ничего не говорит о глубоком смысле запроса.

В результате анализа запроса дерево синтаксического разбора непосредственно отображается в выражение на языке запросов к базе данных. Типичная система, основанная на синтаксическом анализе - LUNAR [1, 2].

Синтаксически-ориентированные системы используют грамматику, описывающую возможные синтаксические структуры пользовательских запросов. Следующий пример показывает упрощенную грамматику систем наподобие LUNAR:

S → NP VP NT; NP → Det N

Det → "кто" | "какой"| "какие"

N → "студент" | "специальность" | "группа" | "вуз" | …

N → N"; N" → "student" | "spech" | "group" | "вуз" | …

VP → V N

VP → V NT

NT → N T

Y → "630" | "620"

Данная грамматика указывает на то, что предложение (S) состоит из группы подлежащего (NP), следующего за группой сказуемого (VP) и т.п. Группа подлежащего состоит из детерминанта (Det), следующего за подлежащим, детерминантом может быть "кто" или "какие", и т.д. Используя эту грамматику, ЕЯИ строит синтаксическую структуру запроса "какие студенты учатся в группе 630", показанную на (рис. 7). ЕЯИ может затем

отобразить дерево синтаксического разбора в следующий запрос к базе данных:

SELECT (student) WHERE group="630".

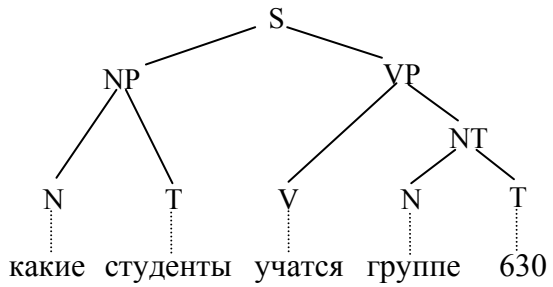


Рис. 7. Дерево синтаксического разбора

Отображение дерева в выражение запроса производится с помощью правил, и целиком основывается на синтаксической информации дерева разбора. Правила отображения могут быть следующими:

- "какие" отображается в SELECT;
- "студент" отображается в student;
- "группа" отображается в group;
- поддерево NT отображается в N="T" (в нашем примере будет group="630");
- поддерево NP отображается в det(N) (SELECT (student));
- поддерево VP отображается в WHERE NT (WHERE group="630");
- S отображается в NP and VP (SELECT (student) WHERE group="630").

Обычно трудно составить систему правил, трансформирующих дерево разбора напрямую в некоторое выражение на языке запросов к реальным базам данных (например, SQL), поэтому данный подход применятся в основном в комбинации с другими.

Семантически-ориентированный метод анализа ЕЯ-запросов был предложен А.С. Нариньяни. Этот подход, основанный на семантике, гораздо ближе к смыслу запроса. В нем используется синтаксическая информация из предыдущего подхода, а также информация из семантических словарей. Каждое слово в словаре имеет характеристики, позволяющие определять смысловые отношения между ним и другими словами, точнее, их значениями. Полное описание связей между смыслами слов (а одно слово часто имеет несколько смыслов) образует тезаурус, представляющий собой большую сеть со словами и их смыслами в качестве узлов. С помощью таких тезаурусов выполняется построение семантического представления запроса. Основная задача при этом — отсечь ненужные смыслы, постараться выделить с помощью синтаксических связей достоверные семантические конструкции. В больших предложениях, особенно с многозначными словами, это часто приводит к комбинаторному взрыву — перебору множества смыслов и связей между ними, а также многозначности синтаксических конструкций (одному и тому же предложению может быть сопоставлено несколько синтаксических представлений), обработка которых занимает неприемлемо большое время. Это лишь одна проблема, стоящая на пути понимания естественно-языковых запросов в традиционной синтаксически-

ориентированной парадигме. Вторая сложность — типичные естественно-языковые запросы, которые, как правило, не имеют правильных синтаксических конструкций. На это влияют вольное словоизменение и словообразование в виде неологизмов сетевой общественности, большой процент имен собственных и сокращений, игнорирование правил пунктуации, что приводит к тому, что от естественного языка во всем его многообразии иногда остается лишь лексика, причудливым образом исковерканная. И, наконец, необходимые в этом подходе семантические словари — очень трудоемкая составляющая, для многих предметных областей они просто отсутствуют, а их разработка требует высокой квалификации.

Третий подход к анализу естественно-языковых запросов основан на шаблонах. Он появился самым первым и с точки зрения программной реализации наиболее прост. Суть его в том, что возможные запросы покрываются набором шаблонов-конструкций, позволяющих отождествляться с запросом и выдавать в результате предопределенные конструкции.

Для примера рассмотрим таблицу базы данных, содержащую информацию о странах (табл. 2):

Таблица 2

Пример таблица базы данных

Страна	Столица	Язык
Россия	Москва	русский
Италия	Рим	итальянский
...	...	...

Простейшую основанную на шаблонах систему можно построить с помощью следующих правил:

Шаблон: ... "столица"..., <страна>

Действие: Вывести столицу в строке, в которой поле страна = <страна>

Шаблон: . . . "столица" . . . "страна"

Действие: Вывести столицу и страну для каждой строки

Согласно первому правилу, если ЕЯ-запрос содержит слово "столица" перед названием страны (т.е. значением поля Страна), то система найдет записи, содержащие это название страны, и выведет соответствующую столицу. Например, для запроса "Какая столица Италии?" будет использовано первое правило, и ответ будет "Рим". То же самое правило применится для запроса "Напечатать столицу Италии", "Подскажите мне, пожалуйста, столицу Италии?" и т.д. Во всех этих случаях ответ будет одинаковым.

В соответствии со вторым правилом, любой ЕЯ-запрос, в котором слово "столица" предшествует слову "страна", вернет столицу каждой страны в соответствии с содержимым таблицы. Так, запросы "Какая столица в каждой из стран?", "Вывести на печать столицу любой страны.", "Столицы и страны, пожалуйста." будут подходить под второе правило.

Главным достоинством шаблонного подхода является его простота - здесь отсутствуют сложные модули синтаксического разбора и интерпретации, такую систему, основанную на шаблонах, легко реализовать. Однако простота подхода имеет оборотную сторону - такие системы сложно портировать, и их надежность понимания запросов оставляет желать

лучшего. Такой подход заключается в необходимости предусмотреть все возможные способы выражений на естественном языке, т.е. исчислить грамматику. К сожалению, современный пользовательский язык совсем не похож на литературный, и поисковые запросы синтаксическими шаблонами в чистом виде покрыть довольно трудно. Если же основываться на семантической грамматике, придется для каждой новой предметной области писать шаблоны заново.

Также выделяются системы с семантической грамматикой и системы с промежуточным языком представления.

В системах с *семантической грамматикой* ответ на ЕЯ-запрос также делается разбором запроса и отображением дерева в выражение на формальном языке. Отличие в том, что грамматические категории не обязательно соответствуют синтаксическим концептам. Ниже показана возможная семантическая грамматика, используя которую, ЕЯИ строит структуру запроса "which rock contains magnesium", показанную на рисунке 8.

Заметим, что некоторые категории грамматики на (Substance, Radiation, Specimen_question) не соответствуют синтаксическим конструкциям (группе подлежащего, подлежащему, предложению). Семантическая информация о предметной области жестко привязана к семантической грамматике. Категории семантической грамматики обычно выбираются так, чтобы усилить семантические ограничения. Например, приведенная грамматика не допускает следования слова "light" после "contains" (синтаксически же эта фраза корректна - "contains light").

Грамматические категории могут быть выбраны также таким образом, чтобы облегчить отображение дерева запроса в запрос к базе данных. Семантическая грамматика была введена как инженерная методология, позволяющая просто включать семантические знания в систему. Однако поскольку семантическая грамматика содержит жестко привязанные знания о конкретной предметной области, системы, основанные на этом подходе, трудно портируются на другие предметные области - каждая ПО требует своей грамматики. Например, приведенная выше грамматика абсолютно неприменима для ЕЯ-интерфейса к кадровой базе данных.

Многие современные ЕИЯ к базам данных сначала преобразуют ЕЯ-запрос в логический запрос на некотором *промежуточном языке представления*. Промежуточный логический запрос выражает значение запроса в терминах модели предметной области, независимой от структуры базы данных. Затем логический запрос преобразуется в запрос на языке запросов к базе данных, этот запрос исполняется в базе данных. Многие современные ЕЯ-интерфейсы к БД используют не один, а несколько промежуточных языков запросов [3, 4, 5]. Принцип анализа следующий: ЕЯ-запрос сначала обрабатывается синтаксически анализатором с использованием набора синтаксических правил для построения дерева синтаксического разбора, аналогичного показанного на рисунке 8. Семантический интерпретатор последовательно трансформирует дерево синтаксического разбора в язык промежуточного представления, используя семантические правила, рассмотренные ранее.

S → Specimen question | Spacecraft question
Specimen question → Specimen Emits info | Specimen Contains info
Specimen → "which rock" | "which specimen"
Emits info → "emits" Radiation
Radiation → "radiation" | "light"
Contains info → "contains" Substance
Substance → "magnesium" | "calcium"
Spacecraft question → Spacecraft Depart info | Spacecraft Arrive info
Spacecraft → "which vessel" | "which spacecraft"
Depart info → "was launched on" Date | "departed on" Date
Arrive info → "returns on" Date | "arrives on" Date

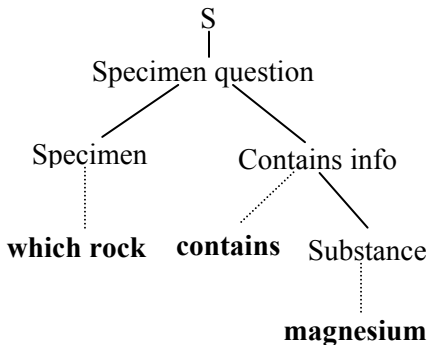


Рис. 8. Дерево разбора в семантической грамматике

К настоящему времени существующие естественно-языковые системы используют в основном два последних подхода. Второй подход реализован в достаточно распространенной системе *ЗАПСИБ,* разработанной в середине 80-х годов [10]. Система позволяет вести общение на ограниченном подмножестве естественного языка. Развитием проекта является система *InterBase*, вышедшая в 1990 году. Система основана на семантически-ориентированном анализе и продолжает ряд естественно-языковых технологий лаборатории искусственного интеллекта ВЦ АН Новосибирска, затем фирмы «Интеллектуальные технологии», а теперь РосНИИ искусственного интеллекта. В 2001 году эта система была переработана и получила название *InBASE* в виде коммерческого продукта. В настоящее время система представляет собой библиотеку COM-компонентов и среду настройки естественно-языковых интерфейсов. Существенным отличием от старой версии является появление промежуточного уровня запросов — Q-языка, являющегося подмножеством языка объектных запросов OQL и уровня описания предметной области в виде диаграммы классов UML. В полном соответствии с особенностями семантически-ориентированной парадигмы *InBASE* позволяет строить естественно-языковые интерфейсы ко многим языкам — для русского и для английского используется один и тот же Л-процессор. Интересной особенностью *InBASE* является возможность моделирования предметной области на естественном языке: с помощью класса словарных статей «Толкование» смысл слова можно описать простой фразой. Это позволяет настраивать естественно-языковые интерфейсы людям, не обладающим навыками инженеров знаний. Основным недостатком данной системы является то, что кортежи базы данных продублированы в словарях – отдельных файлах. В базах данных больших объемов этот недостаток может стать проблемой.

Ярким представителем третьего подхода является система *English Query*. Система *English Query* от Microsoft основана на синтаксически-ориентированных шаблонах, связываемых с моделью предметной области, и через нее - со схемой базы данных. При настройке необходимо задать модель базы данных и предметной области, а затем для каждого отношения в базе данных (а отношением считается и связь между классом и его

атрибутом, например, между товаром и его ценой) задать синтаксический шаблон английской грамматики, выбираемый из списка. Этот продукт позволяет строить естественно-языковые интерфейсы только для английского языка и работает только с Microsoft SQL Server, в этом смысле это лишь утилита, поставляемая с SQL-сервером, именно так она и позиционируется. В целом же этот продукт очень интересен. Например, в нем есть встроенная обучаемая база знаний, с которой можно пообщаться на английском языке, — она запоминает факты, правила и отвечает на вопросы по этой базе. К сожалению, эта замечательная способность не совмещена с пониманием запросов к базе данных.

База знаний, необходимая для выполнения анализа запроса, содержит метаописание базы данных и знания проблемной среды. Модуль метаописания БД состоит из описания концептуальной схемы базы данных на естественном языке: сущностей, атрибутов и связей между сущностями. Модуль словарей содержит знания для проведения морфологического, синтаксического анализов и трансляции естественно-языковых запросов к базе данных. Модуль знаний проблемной среды содержит описания понятий и терминов предметной области.

Диалоговые системы решения задач

Основное отличие ЕЯ-систем данного класса от ЕЯ-систем общения с БД состоит в той роли, которую играет система в процессе решения задач пользователя. Системы общения с БД лишь облегчают получение из БД информации. Они, как правило, не имеют знаний о задаче, для решения которой пользователю нужна эта информация. При общении с диалоговыми системами решения задач пользователь и система меняются ролями. Цель системы состоит в получении решения задачи на основе использования как собственных знаний и механизмов вывода, так и данных, получаемых из ответов пользователя и из прикладных программ, которые могут вызываться диалоговой системой для непосредственного решения каких-то подзадач.

В системах данного класса требуется выполнить или упорядочить для последующего выполнения (т. е. спланировать) действия, позволяющие получить решение некоторой типовой, стереотипной задачи. Каждый класс подобных стереотипных задач характеризуется тем, что входящие в него задачи имеют одинаковую и хорошо определенную структуру и отличаются друг от друга лишь значениями органического числа параметров. Поэтому для инициирования процесса решения задачи пользователю достаточно сообщить системе преследуемую им цель (т. е. идентифицировать тип задачи) и задать ограничения на значения каких-то параметров решаемой задачи. Если какие-то параметры пропущены или заданы (с точки зрения системы) неправильно, то система перехватывает инициативу, инициируя диалоги по уточнению параметров. В ходе этих диалогов пользователь также может перехватывать инициативу, задавая системе вопросы для того, чтобы использовать полученные сведения при формировании ответов на предыдущие вопросы системы.

Предопределенность решаемой задачи и наличие детальных сведений о ее структуре приводят к тому, что основные функции ЕЯ-системы могут успешно выполняться в более сложной (чем в случае ЕЯ-систем общения с БД) постановке. Так, вместо жесткой структуры

диалога в диалоговых системах решения задач может использоваться альтернативная или гибкая структура с произвольным перехватом инициативы. Понимание входных высказываний осуществляется с учетом текущего состояния диалога и имеющихся у системы целей. Благодаря этому упрощается понимание высказываний, содержащих неправильности, и в то же время повышается непроцедурность общения (так как система может рассматривать высказывания пользователя как определения условий текущих подзадач). Высказывания системы строятся в виде фраз естественного языка. Их генерация осуществляется, как правило, в соответствии с коммуникативными намерениями, которые определяются компонентом ведения диалога. В связи с этим содержание высказываний системы может в значительной степени варьироваться. Это могут быть результаты решения задач, вопросы, касающиеся каких-то параметров задач, объяснения действий системы и имеющихся у нее представлений о проблемной области и т. п.

Основной областью практического использования диалоговых систем решения задач является обеспечение ЕЯ-доступа к различным прикладным системам, предназначенным для решения задач реальных объемов и сложности. При этом диалоговая система выступает в качестве интерфейса между прикладной системой и конечным пользователем, не знающим входного языка прикладной системы и имеющим лишь самое общее представление об алгоритме решения задачи. В этом случае процесс решения задачи распадается на следующие этапы:

- информирование пользователя о возможностях прикладной системы;
- получение от пользователя исходных данных (описаний задач, подзадач и их параметров), их уточнение и формирование заданий на входном языке прикладной системы;
- собственно решение подзадач прикладной системой;
- предоставление пользователю результатов решения задачи.

Большинство из существующих в настоящее время ЕЯ-систем данного типа предназначены для общения пользователя (клиента) с экспертными системами (ЭС) в процессе кооперативного решения задачи. Вместе с тем разрабатываются и ЕЯ-системы, не прибегающие в процессе решения задач к помощи пользователей (эти системы могут найти применение в простых проблемных областях).

В настоящее время разработано достаточно много систем данного класса, но все они ориентированы на решение определенного круга вопросов в конкретной предметной области. Например:

- система Snuka – обеспечивает общение на английском языке с экспериментальной ЭС Knobs, решающей задачи планирования военных операций (система позволяет вводить в ЭС компоненты плана, получать ответы на вопросы пользователя о предметной области, анализировать высказывания пользователя и, по желанию пользователя, автоматически генерировать полный план);
- система Xcalibur - обеспечивает общение на английском языке с

экспериментальной ЭС Xsel, выполняет функции консультанта, помогающего пользователю выбрать нужные ему компоненты вычислительной техники и формирующего с помощью системы R1 заказ на конфигурацию технических средств;

- система Advisor выполняет функции консультанта, способного отвечать на вопросы студентов о различных дисциплинах и давать советы относительно возможности или необходимости изучения той или иной дисциплины и др.

На рисунке 9 показана примерная схема диалоговых систем решения задач (система Advisor), в которой отражены компоненты данного класса ЕЯ-систем и их взаимосвязь.

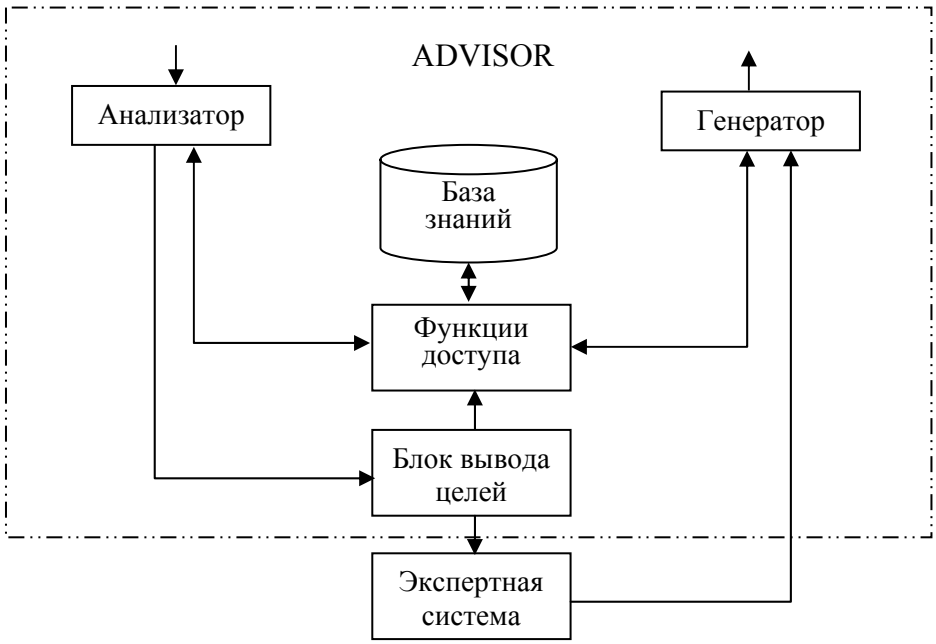


Рис. 9. Структурная схема системы ADVISOR

Существующие ЕЯ-системы данного класса пока не отвечают требованиям, диктуемым условиями промышленной эксплуатации (например, требование простоты настройки ЕЯ-системы на класс решаемых задач и на прикладную систему). В то же время следует обратить внимание на тенденцию применения в качестве подсистем ЕЯ-системы хорошо зарекомендовавших себя и допускающих настройку фрагментарных систем. Эта тенденция особенно заметна в ЕЯ-системах, решающих задачи с помощью собственных механизмов вывода.

Системы обработки связных текстов

Системы данного класса моделируют процесс понимания законченных описаний определенных фрагментов действительности (историй, рассказов, эпизодов и т. п.), выраженных в виде текста на естественном языке, т. е. последовательности связанных друг с другом предложений. Понимание текста трактуется как извлечение из него всей существенной с точки зрения системы информации и присоединение ее к собственной базе знаний. После этого система может отвечать на вопросы относительно фактов, событий,

явлений и прочих сущностей, которые явно или косвенно описаны во введенных текстах. Очевидно, что в практическом плане модели и методы, развиваемые в системах обработки связных текстов, могут быть полезны при создании интеллектуальных систем автоматического индексирования и реферирования.

Для примера рассмотрим системы Researcher и Tailor, которые образуют единый комплекс (рис. 10), позволяющий пользователю получать сведения из рефератов-патентов, описывающих сложные физические объекты. Система Researcher получает рефераты патентов, стрит на их основе базу знаний и делает обобщения их различных патентов, которые могут служить для изучения содержащихся в рефератах сведений, относящихся к различным объектам. Вопрос-ответные функции выполняет система Tailor.

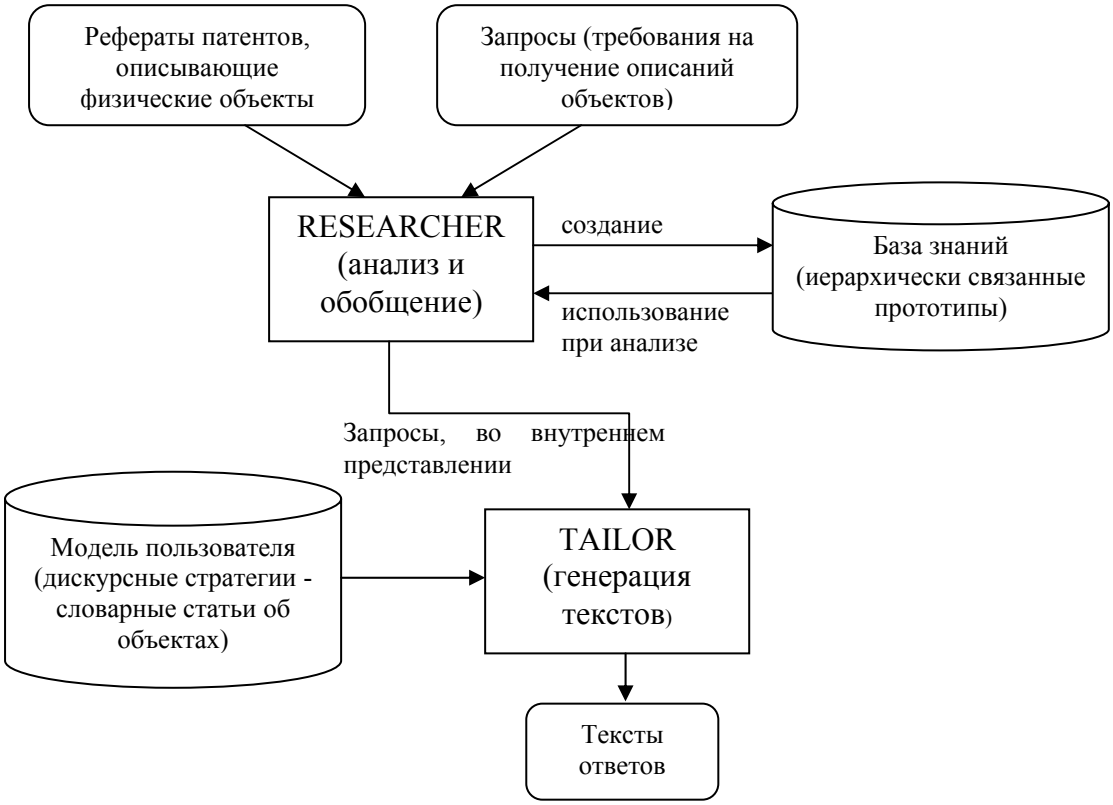


Рис. 10. Схема взаимодействия систем RESEARCHER и TAILOR

Задача понимания связных текстов превосходит по сложности задачи, решаемые ЕЯ-системами ранее рассмотренных классов. Наиболее сложными для понимания являются тексты, описывающие взаимоотношения и поступки активных действующих лиц. Для их понимания система должна обладать громадным объемом знаний о мире, иметь совершенные механизмы вывода. Поэтому в настоящее время системы обработки связных текстов находятся на стадии разработки экспериментальных образцов, которые используются для исследования и оценки методов решения этой крайне сложной и многогранной задачи.

Системы машинного перевода

Машинный перевод (МП), или *автоматический перевод* (АП) - интенсивно

развивающаяся область научных исследований, экспериментальных разработок и уже функционирующих систем (СМП), в которых к процессу перевода с одного естественного языка на другой привлекается вычислительная система. СМП открывают быстрый и систематический доступ к информации на иностранном языке, обеспечивают оперативность и единообразие в переводе больших потоков текстов, в основном научно-технических. Работающие в промышленном масштабе СМП опираются на большие терминологические банки данных и, как правило, требуют привлечения человека в качестве пред-, интер- или постредактора. Современные СМП, и в особенности те, которые опираются при переводе на базы знаний в определенной предметной области, относят к классу систем искусственного интеллекта.

Основные сферы использования МП:

1. В отраслевых службах информации при наличии большого массива или постоянного потока иноязычных источников. Если СМП используются для выдачи сигнальной информации, постредактирование не требуется.

2. В крупных международных организациях, имеющих дело с многоязычным политематическим массивом документов. Поскольку требования к переводу здесь высоки, МП нуждается в постредактировании.

3. В службах, осуществляющих перевод технической документации, сопровождающей экспортируемую продукцию. Структура и язык технической документации достаточно стандартны, что облегчает МП и даже делает его предпочтительным перед ручным переводом, так как гарантирует единый стиль всего массива. Поскольку перевод спецификаций должен быть полным и точным, продукция МП нуждается в постредактировании.

Помимо практической потребности делового мира в СМП, существуют и чисто научные стимулы к развитию МП: стабильно работающие экспериментальные системы МП являются опытным полем для проверки различных аспектов общей теории понимания, речевого общения, преобразования информации, а также для создания новых, более эффективных моделей самого МП.

В классических СМП, осуществляющих непрямой перевод по отдельным предложениям (пофразный перевод), каждое предложение проходит последовательность преобразований, состоящую из трех этапов: АНАЛИЗ→ТРАНСФЕР (межъязыковые операции)→СИНТЕЗ. В свою очередь, каждый из этих этапов представляет собой достаточно сложную систему промежуточных преобразований.

Цель этапа анализа - построить структурное описание (промежуточное представление, внутреннее представление) входного предложения. Задача этапа трансфера (перевода) - преобразовать структуру входного предложения во внутреннюю структуру выходного предложения. К этому этапу относятся и замены лексем входного языка их переводными эквивалентами (лексические межъязыковые преобразования). Цель этапа синтеза - на основе полученной в результате анализа структуры построить правильное предложение выходного языка.

Лингвистическое обеспечение стандартной современной СМП включает:

- словари;
- грамматики;
- формализованные промежуточные представления единиц анализа на разных этапах преобразований.

Помимо стандартных, в отдельных СМП могут иметься и некоторые нестандартные компоненты. Так, экспертные знания о предметной области могут задаваться с помощью специальных концептуальных сетей, а не в виде словарей и грамматик.

Механизмы (алгоритмы, процедуры) оперирования с имеющимися словарями, грамматиками и структурными представлениями относят к математико-алгоритмическому обеспечению СМП.

Одно из необходимых требований к современным СМП - высокая модульность. С лингвистически содержательной точки зрения это означает, что анализ и следующие за ним процессы строятся с учетом теории лингвистических уровней. В практике создания СМП различают четыре уровня анализа (рис. 11).

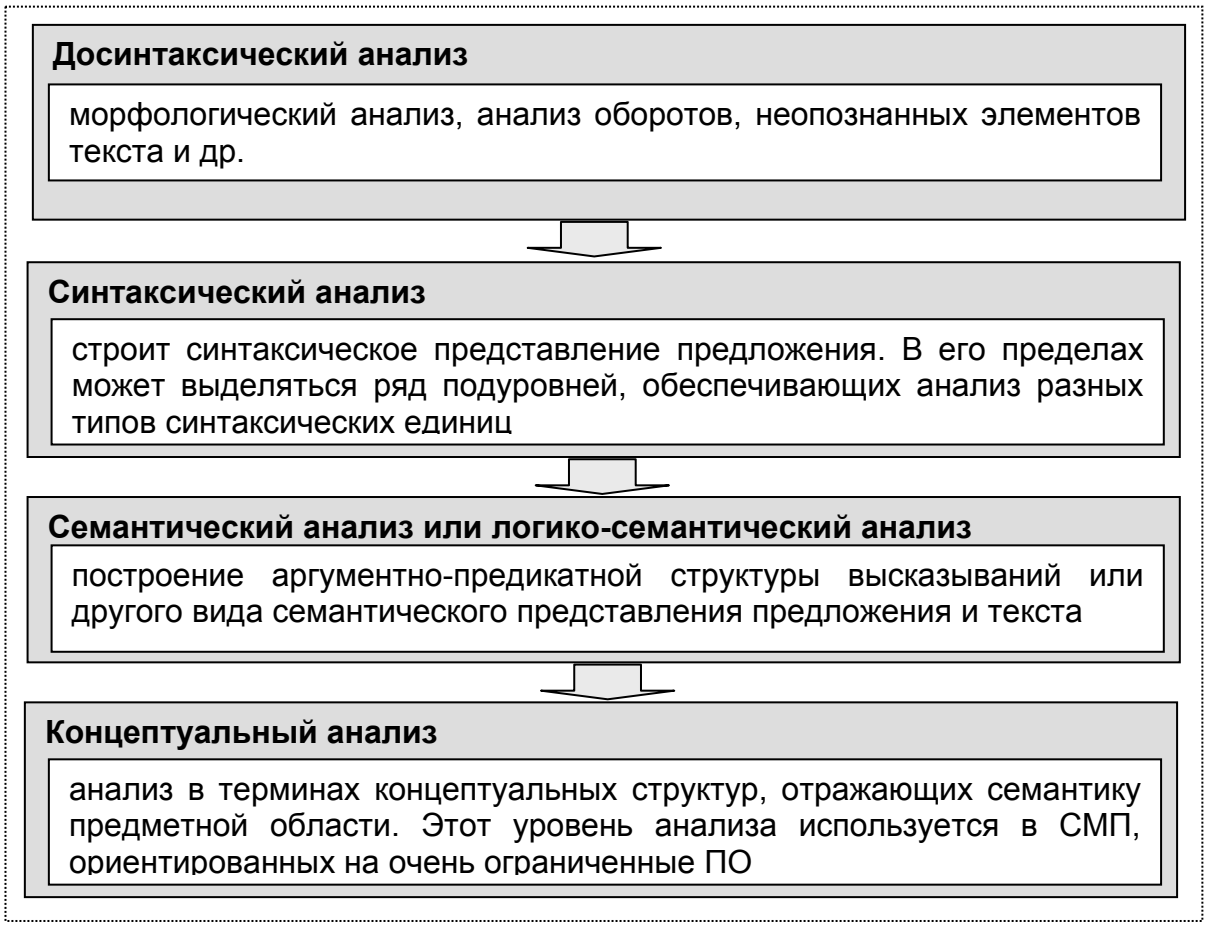


Рис. 11. Основные уровни анализа в СМП

Синтез теоретически проходит те же уровни, что и анализ, но в обратном направлении. В работающих системах обычно реализован только путь от синтаксического представления до цепочки слов выходного предложения.

Лингвистическое разграничение разных уровней может проявляться также в разграничении используемых в соответствующих описаниях формальных средств (набор этих средств задается для каждого уровня отдельно). На практике часто задаются отдельно лингвистические средства морфологического анализа и совмещаются средства двух остальных этапов. Но разграничение уровней может оставаться только содержательным при использовании в их описаниях единого формализма, пригодного для представления информации всех выделяемых уровней.

С технической точки зрения модульность лингвистического обеспечения означает отделение структурного представления фраз и текстов (как текущих, временных знаний о тексте) от «постоянных» знаний о языке, а также языковых знаний - от знаний ПО; отделение словарей от грамматик, грамматик - от алгоритмов их обработки, алгоритмов - от программ.

Словари анализа, как правило, одноязычные. Они должны содержать всю информацию, необходимую для включения данной лексической единицы (ЛЕ) в структурное представление. Часто разделяют словари основ (с морфолого-синтаксической информацией: часть речи, тип словоизменения, подкласс, характеризующий синтаксическое поведение ЛЕ и т. п.) и словари словозначений, содержащие семантическую и концептуальную информацию: семантический класс ЛЕ, семантические падежи (валентности), условия их реализации во фразе и т. д.

Во многих системах разделены словари общеупотребительной и терминологической лексики. Такое разделение дает возможность при переходе к текстам другой предметной области ограничиваться лишь сменой терминологических словарей. Словари сложных ЛЕ (оборотов, конструкции) образуют обычно отдельный массив, словарная информация в них указывает на способ «собирания» такой единицы при анализе. Часть словарной информации может задаваться в процедурной форме, например, многозначным словам могут сопоставляться алгоритмы разрешения соответствующего типа неоднозначности.

Грамматика и словарь задают лингвистическую модель, образуя основную часть лингвистических данных. Алгоритмы их обработки, т. е. соотнесения с текстовыми единицами, относят к математико-алгоритмическому обеспечению системы.

Разделение грамматик и алгоритмов важно в практическом смысле тем, что позволяет менять правила грамматики, не меняя алгоритмов (и соответственно программ), работающих с грамматиками. Но далеко не всегда такое разделение возможно. Так, для системы с процедурным заданием грамматики и тем более с процедурным представлением словарной информации такое разделение нерелевантно. Алгоритмы принятия решений в случае недостаточной (неполнота входных данных) или избыточной (вариантность анализа) информации в большой мере эмпиричны, их формулировка требует лингвистической интуиции.

Наиболее четко разделение грамматик и алгоритмов наблюдается в системах, работающих с *контекстно-свободными* (КС) *грамматиками* (КСГ), где модель языка - *грамматика с конечным числом состояний,* а алгоритм должен обеспечить для произвольно

взятого предложения дерево его вывода по правилам грамматики, и если таких выводов несколько, то перечислить их. Такой алгоритм, представляющий собой формальную (в математическом смысле) систему, называется *анализатором.* Описание грамматики служит для анализатора, обладающего универсальностью, таким же входом, как и анализируемое предложение. Анализаторы строятся для классов грамматик, хотя учет специфических особенностей грамматики может повысить эффективность анализатора.

Грамматики *синтаксического уровня* — наиболее разработанная часть и с точки зрения лингвистики, и с точки зрения их обеспечения формализмами. Укажем основные типы грамматик и реализующих их алгоритмов (в литературе по МП их часто описывают как одну совокупность).

Цепочечная грамматика фиксирует порядок следования элементов, т. е. линейные структуры предложения, задавая их в терминах грамматических классов слов (артикль+существительное+предлог...) или в терминах функциональных элементов (подлежащее+сказуемое). Примером реализации такой языковой модели является *предсказуемостный синтаксический анализ:* идентифицированная грамматическая категория слова предсказывает (с определенной долей вероятности) появление грамматической категории следующего за ним слова. Стратегия анализа — «слева направо»: перебор слов, проверка предсказаний, их изменение и добавление новых предсказаний регулируются механизмом «магазинной памяти» (last in first out).

Грамматика составляющих (или грамматика *непосредственно составляющих* - НСГ) фиксирует лингвистическую информацию о группировке грамматических элементов, например, именная группа (состоит из существительного, артикля, прилагательного и других модификаторов), предложная группа (состоит из предлога и именной группы) и т. д. до уровня предложения. Грамматика строится как набор *правил подстановки,* или *исчисление продукций* вида $A \rightarrow B \rightarrow ...C$. НСГ представляют собой грамматики *порождающего типа* и могут использоваться как при анализе, так и при синтезе: предложения языка порождаются многократным применением таких правил.

Грамматика зависимостей (ГЗ) задает иерархию отношений элементов предложения (главное слово определяет форму зависимых). Анализатор в ГЗ основан на идентификации *хозяев* и их зависимых *(слуг).* Главным в предложении является глагол в личной форме, так как он определяет число и характер зависимых существительных. Стратегия анализа в ГЗ — *сверху вниз* (top-down): сначала идентифицируются хозяева, затем слуги, или *снизу вверх:* (bottom-up): хозяева определяются процессом подстановки.

Новым и, сразу завоевавшим популярность, методом грамматического описания является лексико-функциональная грамматика (ЛФГ). Она устраняет необходимость трансформационных правил. Хотя ЛФГ основывается на КСГ, проверочные условия в ней отделены от правил подстановки и «решаются» как автономные уравнения.

Лекция 3. Методы реализации ЕЯ-систем

Приведем методы реализации основных функциональных компонент, получившие наиболее широкое распространение в практике создания ЕЯ-систем.

Методы реализации диалогового компонента

Диалог можно рассматривать на трех уровнях: общая (глобальная) структура, характеризующая тип диалога и класс решаемых задач; тематическая структура, отражающая структуру конкретной задачи; структура шага диалога (локальная структура), отражающая взаимодействие участников в элементном акте диалога.

На уровне глобальной структуры действия ЕЯ-системы обычно задаются в виде последовательности этапов, определяемых в зависимости от класса решаемых задач. Так, в случае общения с экспертными системами глобальная структура включает следующие этапы: инструктаж, определение задачи, решение задачи, объяснения в ходе решения задачи, выдача результатов решения задачи и их оценка, объяснения после решения задачи, определение причин неудачи и приобретение новых знаний.

Перечисленные этапы не обязательно должны выполняться в каждом конкретном диалоге, но если они выполняются, то в том порядке, который указан в глобальной структуре. Тот или иной этап может не выполняться либо в связи с явным указанием пользователя, либо по умолчанию. Например, если результат решения задачи удовлетворяет пользователя, то этапы объяснения и приобретения могут быть пропущены. В силу простоты и статичности глобальной структуры она, как правило, встраивается в управляющий механизм диалогового компонента (т. е. задается процедурно), однако в ряде систем, ориентированных на многоцелевое применение, глобальная структура задается декларативно с помощью правил, имеющих вид продукций.

Тематическая структура диалога обычно представляется в виде сценария, в рамках которого определяются: структура задачи, решаемой в процессе общения, т. е. разбиение задачи на упорядоченное множество подзадач; распределение подзадач между участками общения, т. е. определение, какие подзадачи решаются системой и какие - пользователем; языковые средства, используемые при обращениях к пользователю. Для задания тематической структуры (сценария) диалога в существующих системах общения применяются различные способы. Они могут быть сгруппированы в три класса:

- сценарий присутствует в системе в «готовом» виде (например, он встраивается в систему при ее создании или вводится в процессе настройки системы на проблемную область);
- сценарий генерируется системой в процессе решения задачи;
- некоторые компоненты сценария присутствуют в системе в готовом виде, а некоторые генерируются.

Последний класс является композицией предыдущих, поэтому ограничимся рассмотрением способов, составляющих первый и второй классы.

Готовый сценарий может быть задан в виде частично упорядоченного множества правил с параметрами, значения которых устанавливаются в процессе решения конкретной задачи. В зависимости от значений параметров между правилами устанавливается отношение строгого порядка, определяющее тематическую структуру конкретного диалога. Использование готовых сценариев целесообразно в тех случаях, когда к системе

предъявляются жесткие требования по быстродействию (это характерно для промышленных и коммерческих систем), а решаемые задачи имеют устойчивую структуру и заранее известны функции (роли) участников общения. Класс подобных задач весьма широк. Это большинство задач общения с базами данных, задачи предупреждения или устранения определенных видов локальных неудач, задачи, возникающие при настройке ЕЯ-систем на проблемную область. Во всех случаях функция ведения диалога ограничивается интерпретацией готового сценария. При этом если сценарий не встроен заранее в диалоговый компонент, а должен вводиться в систему при ее настройке на проблемную область, то для его описания используются специальные языки. Примерами отечественных систем, имеющих языки описания сценариев, могут служить АИСТ и АДС.

Если сценарий диалога не присутствует в системе в готовом виде, а генерируется в процессе решения задачи, то в диалоговый компонент включается специальный механизм вывода (планирования). Метод планирования определяется в зависимости от используемой в конкретной ЕЯ-системе системы представления знаний. Например, если для представления знаний применяется исчисление предикатов, то процесс генерации сценария реализуется методами доказательства теорем. Специфика применения методов планирования для генерации сценариев состоит в том, что в качестве операторов, решающих элементарные подзадачи, рассматриваются не только действия, ведущие к изменению отношений между сущностями проблемной области, но и типовые действия, которые соответствуют определенным речевым поступкам (речевым актам, например таким, как «сообщать», «спрашивать», «предлагать»). Это позволяет естественным образом включить в план решения задачи - в последовательность действий по решению элементарных подзадач, полученную с помощью механизма вывода - речевые акты, т. е. определить языковые средства, используемые для обращений к пользователю.

Генерация сценариев диалога целесообразна в тех случаях, когда структура задачи зависит от контекста ситуации, в которой происходит ее решение, а полный перечень ситуаций не может быть определен заранее. К подобным задачам относится большинство задач общения с экспертными системами, а также многие задачи, возникающие при обработке связных текстов (в последнем случае методы планирования позволяют устанавливать цели и строить планы действий (в том числе и речевых) участников событий, описываемых в текстах).

При задании локальной структуры шаг диалога состоит из действия и реакции и характеризуется следующими параметрами: инициатор и тип инициирования (вид) действия; способ влияния действия на реакцию; способ спецификации подзадачи, решаемой на данном шаге. Последний параметр будем характеризовать двумя подпараметрами: ограниченностью пространства выбора функции (и/или параметров), используемой для решения подзадачи, и однозначностью определения функции. Данные параметры позволяют выделить шесть основных типов шагов диалога для ЕЯ-систем (табл. 3).

На локальном уровне диалога задача диалогового компонента состоит в определении параметров текущего шага. Инициатором шага в общем случае может быть как пользователь,

так и система. Действия пользователя подразделяются на задания и команды. Задания предусматривают формулировку решаемой задачи (подзадачи) и ввод необходимых для ее решения параметров. При этом однозначность определения функции, обеспечивающей решение этой задачи, не гарантируется. Например, пользователь может специфицировать задачу, для решения которой в системе отсутствует подходящая функция (либо такая функция есть, но она не может быть выполнена на данном шаге), или задать значения параметров не в том виде, в котором требуется системе. Команды обычно служат для перехвата инициативы или для выполнения технологических действий, например листания отчета, получения твердой копии и т. п. Фиксированность набора команд позволяет системе легко идентифицировать команду и однозначно определять требуемую для выполнения команды функцию.

Таблица 3
Основные типы шагов диалога

Инициатор шага	Вид действия	Вид реакции	Способ спецификации подзадач	
			ограниченность пространства выбора	однозначность определения
Пользователь	задание	отчет или диагностическое сообщение	неограниченное	неоднозначное
Пользователь	команда	выполнение команды или диагностическое сообщение	ограниченное	однозначное
Система	простой вопрос	ответ на простой вопрос	фиксированное	то же
Система	вопрос с фиксированной структурой ответа	ответ в заданной структуре	ограниченное	то же
Система	вопрос со свободной структурой ответа	произвольный ответ	неограниченное	неоднозначное
Система	альтернативный вопрос (предложение выбора)	выбор альтернативы	ограниченное	однозначное

Если инициатива принадлежит системе, то вид действия определяется исходя из того, насколько диалоговому компоненту известна функция (и/или параметры), с помощью которой осуществляется решение подзадачи на данном шаге. Если функция известна, а неизвестны лишь некоторые параметры этой функции, то используется *простой вопрос*. Например, пользователь задал системе вопрос: «Какой домашний адрес у Петрова?». В этом случае для более релевантного поиска система уточняет один из параметров, путем задания простого вопроса: «Введите номер отдела, в котором работает Петров».

Альтернативные вопросы и вопросы с фиксированной структурой ответа применяются в тех случаях, когда на данном шаге возможно выполнение одной из нескольких функций, а выбор осуществляется в зависимости от реакции пользователя. Альтернативные вопросы

ограничивают выбор явно, т.е. пользователю предъявляется множество возможных ответов (например, для предыдущего примера: «Петров работает в 1, 2 или 3-ем отделе?»). Вопросы с фиксированной структурой ответа предусматривают неявное определение функции. Эти вопросы содержат анкету для ввода ответа. Анкета состоит из именованных полей, предназначенных для ввода соответствующих значений. Определение функции осуществляется в зависимости от того, какие поля анкеты будут заполнены пользователем при вводе ответа.

Вопросы со свободной структурой ответа не содержат никаких сведений относительно ожидаемых свойств ответа. В принципе эти вопросы в существенно меньшей степени ограничивают (вплоть до отсутствия каких бы то ни было ограничений) область выбора функции и параметров. Естественно, что при этом не гарантируется однозначность их определения. С этой точки зрения вопросы со свободной структурой ответа аналогичны заданиям, разница между ними — только в инициаторе действия.

Из рассмотрения основных типов шагов диалога следует, что при действиях, не ограничивающих возможные реакции (заданиях и вопросах со свободной структурой ответа), не гарантируется однозначная спецификация задачи. Кроме того, неоднозначность спецификации может иметь место и в тех случаях, когда действие системы ограничивает реакцию пользователя (простой вопрос и вопрос с фиксированной структурой ответа), но вследствие использования естественного языка ответ неправильно понимается системой. Во всех подобных случаях требуется приведение ситуации к однозначной. На практике для этого применяется перефразирование или изменение темы диалога.

Перефразирование заключается в переформулировании на естественном языке высказывания (задания или ответа) пользователя. При неоднозначном понимании все возможные (с точки зрения системы) варианты выдаются в виде альтернативного вопроса. Частным случаем перефразирования является «эхо». При этом повторяется часть высказывания пользователя, в которой система сомневается (например, альтернативный вопрос системы: «Выдать домашний адрес Петрова из 1-го отдела?»).

При изменении темы диалога текущий шаг диалога прерывается, и для достижения однозначного понимания внутри него создается поддиалог из одного или нескольких шагов. Поддиалог может быть заранее запланирован в сценарии диалога либо инициирован в результате перехвата инициативы ранее пассивным участником. В последнем случае в зависимости от того, кто из участников осуществляет перехват инициативы, диалоговый компонент либо формирует перехват, либо обрабатывает его.

Как правило, действия участников по перехвату инициативы ограничены моментом перехвата, способом перехвата и целями, которые участники могут преследовать, перехватив инициативу. Обычно перехват инициативы разрешается в те моменты, когда активный участник находится в ожидании реакции пассивного участника. Наиболее распространенный способ перехвата инициативы системой - простые и альтернативные вопросы. В первом случае целью системы является уточнение параметра, а во втором - функции, которая должна быть выполнена для решения подзадачи на прерванном шаге диалога. Для перехвата

инициативы пользователем обычно применяются специальные стандартные команды, смысл которых заранее известен системе. Появление в высказывании пользователя подобной команды сигнализирует системе, как о наличии перехвата инициативы, так и о цели перехвата.

Если способ перехвата инициативы пользователем не ограничен специальными командами, то диалоговый компонент должен определять по смыслу очередного высказывания его отношение к текущей цели (теме). Если взаимосвязь высказывания с текущей целью установить не удается, то это высказывание должно рассматриваться как перехват инициативы. При этом возникает достаточно сложная задача определения цели, которую, перехватив инициативу, намерен преследовать пользователь. Данная задача в настоящее время еще не имеет удовлетворительного решения, однако следует подчеркнуть, что способность диалогового компонента обрабатывать перехваты инициативы с учетом целей участников является одним из необходимых условий для организации гибкого диалога, гарантирующего достижение в процессе общения глобального успеха.

Методы реализации компонента понимания высказываний

Понимание высказываний включает анализ и интерпретацию.

В методах анализа обычно выделяются анализ слов, предложений и текстов. *Анализ слов* сводится к морфологическому анализу, обнаружению и исправлению орфографических ошибок. Цель *морфологического анализа* состоит в получении основ (под основой понимается словоформа с отсеченным окончанием) со значениями грамматических категорий (например, часть речи, род, число, падеж) для каждой из словоформ высказывания, поступившего на вход ЕЯ-системы. Методы морфологического анализа были детально разработаны еще при создании первых ЕЯ-систем и более подробно рассмотрены в главе III. Примерами отечественных ЕЯ-систем с достаточно полной для практических потребностей реализацией морфологического анализа могут служить ПОЭТ, TULIPS и АИСТ.

Методы обнаружения и исправления орфографических ошибок подразделяются на два класса в зависимости от того, используют они словари основ или нет. К методам, не использующим словари, относятся частотные и полиграммные. *Частотные методы* основаны на сортировке слов по частоте их встречаемости в текстах. Предполагается, что частота встречаемости слов, содержащих ошибки, низкая. Однако низкая частота встречаемости и у правильных, но редко встречающихся слов, что значительно снижает эффективность частотных методов. В *полиграммных методах* для поиска ошибок применяют списки возможных сочетаний букв в словах (обычно анализируются пары и тройки идущих подряд букв). Полиграммными методами целесообразно пользоваться в системах с открытым (пополняемым) словарем наряду с методами, основанными на словарях.

Методы, в которых используются словари, разделяются в зависимости от типа применяемой стратегии на абсолютные и относительные. К *абсолютным* относится «исторический» метод, основанный на словаре встречаемых ранее ошибок. Данный метод реализован, например, в системе SPEEDCOP. Эффективность исторического метода

существенно зависит от размера текстов, на основе которых порожден словарь ошибок. *Относительный метод* состоит в нахождении в словаре таких слов, которые наиболее похожи на искаженное слово, и выборе среди них правильного. Обычно искаженное слово подвергается определенной обработке для получения из него правильных слов. Обработка, как правило, включает действия по пропуску, перестановке и вставке букв. При этом для уменьшения списка новых слов применяются частотные и полиграммные методы.

Анализ предложений обычно сводится к синтаксическому и семантическому анализу, выполняемому отдельным функциональным блоком-анализатором (parser). Наиболее распространенные методы анализа предложений, так же как и методы морфологического анализа, были разработаны еще при создании первых ЕЯ-систем и предназначались для обработки только «правильных», т. е. не содержащих отклонений от грамматической нормы, предложений. Обычно при описании анализаторов основное внимание уделяется распределению функций между синтаксическим и семантическим анализом и порядку их выполнения. Однако с точки зрения современных требовании к ЕЯ-системам более важным является вопрос о том, насколько существующие анализаторы могут быть приспособлены к обработке «неграмматичностей», т. е. характерных для диалогов между людьми высказываний с отклонениями от грамматической нормы (лексические и грамматические ошибки, пропуски, повторы, шумы, эллипсис, идиомы и т.п.). Сравним по этому параметру следующие типы анализаторов: традиционные, концептуальные, анализаторы, использующие сопоставление по образцам и анализаторы, использующие разнообразные стратегии. Более подробно существующие подходы, методы и алгоритмы синтаксического и семантического анализов рассмотрены в следующих разделах.

Традиционные анализаторы

Наиболее распространенным способом анализа ЕЯ-предложений является разбор сверху вниз, слева направо, основанный на некоторой фиксированной грамматике. В последние годы подобные методы обычно выполнялись с применением ATN-техники, т.е. с помощью расширенных сетей переходов. Такие анализаторы осуществляют разбор предложения либо в общих грамматических категориях, либо в терминах категорий, имеющих значение в некоторой ограниченной области. Анализаторы этого типа чрезвычайно «хрупки», т. е. они терпят неудачу при разборе предложений, содержащих минимальные отклонения от нормы.

Хрупкость традиционных анализаторов обусловлена тем, что их алгоритм осуществляет поиск сверху вниз среди разборов, допускаемых грамматикой, того разбора, который соответствует обрабатываемому предложению. Если некоторый частный разбор при сопоставлении ему очередного слова противоречит используемой грамматике, то для анализатора это сигнал того, что на более раннем этапе поиска сделан ошибочный выбор. Таким образом, неудача на некотором шаге разбора является сигналом для выбора очередного из возможных разборов, т. е. принципиальные затруднения возникают при обработке предложений, содержащих отклонения от грамматики.

Для преодоления указанных недостатков традиционных анализаторов были

предложены способы, позволяющие ослаблять действие грамматических правил. Однако это возможно только в ограниченном классе грамматических отклонений. Кроме того, предпринимались попытки добавить в ATN-сеть дополнительные специфические дуги, которые имеют дело с проблематичными входными предложениями. Некоторые из этих дуг выполняют функцию сопоставления по образцу (см. ниже). Тот факт, что для обработки неграмматичных предложений приходится осуществлять радикальные преобразования ATN-техники, говорит о ее малой пригодности для обработки высказываний, имеющих место в реальных, естественных диалогах.

Один из возможных подходов к преодолению хрупкости традиционных анализаторов состоит в одновременном применении нескольких подграмматик. Каждая из подграмматик предназначена для анализа частных конструкций какого-либо одного вида. Применение подграмматик осуществляется независимо, поэтому неудача одной подграмматики не влияет на возможности других. Впервые подобный подход был реализован в системе PLANES, которая имеет подграмматики для каждого типа известных систем сущностей. При данном подходе предложение в процессе разбора разбивается на несколько независимых фрагментов. В этом случае в задачу анализатора входит построение общей (объединенной) интерпретации предложения. Если проблемная область достаточно ограничена (как это имеет место в системе PLANES), то интерпретация фрагментов всегда уникальна, однако в общем случае эта задача не имеет единственного решения и может стать трудноразрешимой.

Концептуальные анализаторы

Анализаторы данного типа используют методы разбора, направляемые значениями базовых событий, обнаруженных в анализируемых предложениях. Наиболее известными разновидностями данного подхода являются анализаторы, основанные на модели концептуальной зависимости и на модели управления. Анализатор первого типа был впервые реализован в системе MAPGE, а второго - в системе ПОЭТ. Концептуальные анализаторы не разрабатывались специально для анализа неграмматичных предложений. Однако заложенные в них идеи в принципе позволяют этим алгоритмам работать в условиях пропусков и повторов слов. Такие системы, как FRUMP, IPP, RESEARCHER, SNUKA с концептуальными анализаторами обладают иммунитетом к ошибкам, так как они игнорируют непонятные им слова, а понятные приспосабливают (даже если в них есть ошибки) к базовым событиям обрабатываемого предложения.

Анализаторы, использующие сопоставление по образцам

Анализаторы данного класса основаны на том, что в простейшем случае анализ сводится к сопоставлению предложения с некоторым множеством образцов, представляющих собой последовательности из одного или нескольких слов. Подобные анализаторы широко применялись в ранних ЕЯ-системах. Многие методы анализа, основанные на сопоставлении по образцам, содержат в образце не только константы, но и переменные. При этом предполагается, что переменные образца могут сопоставляться с любой строкой символов. Гибкость анализаторов определяется гибкостью процесса

сопоставления. Различаются следующие формы сопоставления: синтаксическое, параметрическое, семантическое и принуждаемое. Разнообразие форм сопоставления позволяет анализировать входные предложения, отклоняющиеся от традиционной грамматики в произвольной степени, однако глубина проникновения в смысл подобных анализаторов обычно невелика.

Возможности методов анализа, основанных на сопоставлении по образцам, можно увеличить, используя частичное сопоставление предложения с одним или несколькими образцами. Примером такой системы может служить Flex-P. Следует отметить, что возможности частичного сопоставления исследованы мало. Однако бесспорно, что этот метод анализа весьма эффективен при обработке предложений с отклонениями от грамматической нормы. Кроме того, сопоставление по образцам позволяет успешно обрабатывать идиомы. Так как идиомы не могут быть обработаны путем интерпретации образующих их слов, то для их обработки неизбежно привлечение механизмов, подобных сопоставлению по образцам. По этой причине некоторые системы, например LUNAR, основанные на традиционных методах анализа, для обработки идиом включают стадию преданализа, на которой применяется механизм сопоставления по образцам.

Методы анализа, использующие сопоставление по образцам, имеют определенные ограничения. Если все предложение сопоставляется с одним образцом, то ясно, что такой подход не позволяет обеспечить точную обработку сложных предложений. Если предложение сопоставляется с несколькими образцами, каждый из которых подходит к отдельному фрагменту предложения, то возникает задача определения границ фрагментов. Кроме того, после сопоставления фрагментов одного предложения с несколькими образцами возникает задача получения общей интерпретации всех фрагментов данного предложения.

Анализаторы, использующие разнообразные методы

Все анализаторы, относящиеся к рассмотренным выше классам, основываются на каком-либо одном методе. Исследования показали, что использование в одном анализаторе нескольких специфических методов позволяет обеспечить гибкость процесса анализа, необходимую для обработки неграмматичных конструкций. Примерами таких анализаторов могут служить CASPAR и MULTIPAR. В настоящее время данный подход еще не получил такого распространения, как рассмотренные ранее подходы. Это объясняется, на наш взгляд, его сравнительной новизной и недостаточной исследованностью.

Перейдем к описанию методов анализа связного текста (дискурса). Связность дискурса достигается как лингвистическими средствами, имеющими языковое выражение, так и экстралингвистическими (ситуационными) средствами - «умолчаниями», не имеющими языкового выражения и основанными на общности знаний участников общения о цели общения и проблемной области. На этапе анализа связного текста, как правило, решается задача выявления связей между предложениями, выражаемых лингвистическими средствами, а на этапе интерпретации - ситуационными.

К основным лингвистическим средствам связи предложений относятся ссылки и эллипсис. В проблеме установления ссылок могут быть выделены две задачи:

1) поиск в предыдущих предложениях (контексте) сущности (референта), обозначаемой данной ссылкой;

2) определение соответствия между референтом и ссылкой.

Простейшим методом решения первой задачи является поиск референта в заданном количестве предыдущих предложений. Однако отсутствие критерия для определения количества просматриваемых предыдущих предложений приводит на практике, как к увеличению времени поиска, так и к ошибкам в установлении ссылок. Решение второй задачи является тривиальным для простейших видов ссылок (т. е. в случае тождества референта и ссылки) и весьма трудным для случаев несовпадения референта и ссылки. Отсутствие удовлетворительных методов решения обеих задач в общей постановке на этапе анализа текста стимулировало попытки их решения на этапе интерпретации (см. ниже).

Задача обработки эллиптических конструкций решается на этапе анализа также в ограниченной постановке. Под эллипсисом понимается сжатая форма высказывания, смысл которой определяется либо предыдущими высказываниями (текстовый эллипсис), либо ситуацией, имеющей место в проблемной области (ситуативный эллипсис). С формально-грамматической точки зрения высказывания, содержащие эллипсис, выглядят как неполные (т. е. содержащие пропуски слов) предложения. На этапе анализа может быть обработан (т. е. восстановлен) только текстовый эллипсис. Сущность методов восстановления текстового эллипсиса, например, состоит в подстановке фрагментов предыдущих высказываний в текущее высказывание, содержащее эллипсис. При этом пропущенные слова в текущем высказывании как бы восстанавливаются из текста предыдущего высказывания. Восстановление ситуационного эллипсиса осуществляется на этапе интерпретации.

На этапе интерпретации решаются две основные задачи:

1) буквальная интерпретация высказываний в контексте диалога;

2) интерпретация на цели участников общения.

Методов решения этих задач в общей постановке не существует. Однако применительно к простым проблемным областям их решение существенно упрощается.

К простым проблемным областям относятся различные области, над которыми решаются простые задачи информационного обслуживания, например: справочные задачи (о погоде, о товарах, о литературе), задачи резервирования (мест, билетов, товаров) и т. п. Все эти задачи обладают «прозрачной» структурой, т. е. они оперируют ограниченным множеством сущностей, которые являются параметрами предлагаемого вида обслуживания. Структура простых задач информационного обслуживания соответствует структуре фрейма. При этом параметры обслуживания соответствуют слотам фрейма, а весь вид обслуживания представляется некоторым фреймом. Работа ЕЯ-системы при обслуживании пользователя состоит в заполнении (означивании) слотов и, возможно, в выполнении некоторых манипуляций над фреймами с заполненными слотами.

Для решения обеих задач интерпретации в рамках фрейм-представлений используется единый механизм означивания фреймов. При этом структура целей участников общения (т. е. разбиение целей на подцели) определяется структурой фрейма, описывающего данную

задачу, а подцели состоят в заполнении слотов, т. е. в идентификации сущностей через описания, представленные в результатах анализа входных высказываний.

В общем случае процесс идентификации некоторой сущности может иметь три исхода:

1 исход: однозначный - данному описанию сопоставляется единственная сущность;

2 исход: многозначный - описанию сопоставляется более чем одна сущность;

3 исход: неудовлетворительный - описанию не сопоставляется ни одна сущность.

Последние два исхода рассматриваются как неудачи буквальной интерпретации и служат сигналами о необходимости установления подцелей более глубокого уровня, предусматривающих устранение неудачи. При этом в диалоговый компонент кроме сообщения о неудаче и типе неудачи передаются исходные данные, позволяющие сформировать (с помощью компонента генерации высказываний) действие системы по перехвату инициативы и открытию уточняющего поддиалога, преследующего новую подцель.

В случае многозначной интерпретации вместе с сообщением о неудаче передаются возможные варианты сопоставления, что позволяет системе генерировать альтернативный вопрос. При неудовлетворительной интерпретации от компонента понимания требуется указать причины, по которым рассматриваемое описание является неудовлетворительным. В большинстве случаев причины неудовлетворительной интерпретации сводятся к неудовлетворительным (неполным или неоднозначным) результатам анализа входных высказываний и к несоответствию пресуппозиций пользователя знаниям системы.

При решении задач интерпретации важную роль играет имеющееся в системе представление общей точки зрения на то, о чем идет речь в текущий момент. Эту точку зрения часто называют фокусом (или фокусом внимания). Разделяемый участниками фокус позволяет им повысить компактность диалога за счет того, что сущности, находящиеся в фокусе, могут либо вообще не упоминаться в высказывании (эллипсис), либо упоминаться в виде кратких описаний (ссылок). Покажем, как используется фокус для установления референтов ссылок и восстановления эллипсиса.

В работе предлагается приравнять фокус к текущей подцели, преследуемой участниками общения. Подобная трактовка рассматривает фокус как набор сущностей, на которые в текущий момент направлено внимание участников, при этом текущая подцель может быть определена как множество сущностей, соответствующих этой подцели. Ясно, что данная трактовка фокуса обеспечивает решение задач установления референтов ссылок и восстановления эллипсиса. Поясним это на примере следующего диалога (П – пользователь, С – система):

П1: Какая зарплата у Петрова?

С1: Вы имеете в виду Алексея Петрова или Николая Петрова?

П2: Я имею в виду Николая.

В данном диалоге пользователь ссылается на Николая Петрова (П2). Референт этой ссылки может быть легко идентифицирован. Действительно, высказывание П1 генерирует подцель «заполнить слот личность». Вследствие многозначной интерпретации система

генерирует подцель «выбрать между двумя личностями». Ответ пользователя выбирает одну из них. С таким же успехом пользователь мог бы ответить: «Я имел в виду первого из них». Заметим, что приведенные рассуждения опираются на тот факт, что высказывание пользователя (П2) не изменило текущую подцель и, следовательно, не изменило текущий фокус.

Если бы в приведенном выше диалоге ответ пользователя (П2) содержал эллипсис, например ответ имел бы вид «Николая» или «Первого из них», то эллипсис мог бы быть легко восстановлен. Действительно, если предположить, что пользователь согласен с фокусированной подцелью системы (выбрать между двумя личностями), то естественно считать, что сущность, упомянутая в ответе, определяет выбор пользователя. Отметим, что при данном методе не возникает необходимость достраивать эллиптическое высказывание до полного высказывания (как это требуется при восстановлении эллипсиса на этапе анализа).

Еще раз подчеркнем, что изложенные выше методы базируются на фрейм-представлениях и широко применяются в ЕЯ-системах, ориентированных на решение простых задач информационного обслуживания. Что же касается методов интерпретации, используемых в более сложных проблемных областях (например, понимание связных текстов, описывающих разворачивающиеся по времени события со многими действующими лицами), то они находятся в стадии становления и пока не поддаются обобщенному описанию, так как сильно зависят как от условий конкретных задач, решаемых ЕЯ-системой, так и от специфики применяемых средств представления знаний.

Методы реализации компонента генерации высказываний

Процесс генерации высказываний состоит из генерации смысла высказывания и синтеза высказывания на ЕЯ. Первый этап часто называется внелингвистическим (или концептуальным) синтезом, а второй - лингвистическим синтезом. Результатом выполнения первого этапа является внутреннее представление смысла генерируемого высказывания. Как правило, на первом этапе решаются следующие задачи:

- определение информации, которая должна быть сообщена пользователю;
- определение уровня общности информации, включаемой в высказывание;
- выделение из множества аспектов, описывающих сущности, о которых говорится в высказывании, аспектов, интересующих и понятных пользователю;
- разбиение сообщаемой информации на части, соответствующие будущим предложениям, и установление последовательности этих частей;
- определение лексем и построение семантического представления высказывания.

На втором этапе обычно решаются следующие основные задачи: построение синтаксической структуры отдельных предложений; приписывание морфологической информации вершинам синтаксических структур отдельных предложений - морфологический синтез словоформ.

До последнего времени в большинстве исследований по генерации высказываний на естественном языке (ЕЯ) наибольшее внимание уделялось проблемам, связанным с решением задач второго этапа. В ходе этих исследований были развиты методы прямой

трансляции формализованного представления смысла в предложения на английском языке, разработаны системные грамматики и механизмы их употребления для порождения текста, предложены критерии для отбора слов из общего словаря системы. Для генерации высказываний на русском языке были разработаны и внедрены в ряде ЕЯ-систем и систем машинного перевода методы морфологического и синтаксического синтеза.

По сравнению с задачами второго этапа, для решения которых существуют апробированные методы, задачи, решаемые на первом этапе генерации, изучены гораздо меньше. Поэтому в большинстве действующих ЕЯ-систем решение задач генерации смысла осуществляется, как правило, в упрощенной или сильно ограниченной постановке, что приводит к упрощению, а иногда и к вырождению задач синтеза высказываний на ЕЯ.

Наиболее простым методом генерации высказываний является метод, который основан на использовании заранее заготовленных шаблонов, содержащих текст на ЕЯ и переменные, вместо которых подставляются конкретные данные (описания сущностей). Как показала практика, данный метод особенно удобен для генерации высказываний типа диагностического сообщения, простого вопроса, альтернативного вопроса и вопроса с фиксированной структурой ответа в определенных стандартных ситуациях, не зависящих от проблемной области, в которой используется ЕЯ-система. В подобных ситуациях текущей целью системы является предупреждение или устранение типовых локальных неудач, вызванных, например, такими факторами: наличие в высказывании пользователя незнакомых системе слов; несоответствие пресуппозиций в высказывании пользователя знаниям системы; многозначная или неудовлетворительная интерпретация высказывания пользователя и т. п.

При методе шаблонов генерация высказываний осуществляется, строго говоря, совместно с диалоговым компонентом, который идентифицирует стандартную ситуацию (т. е., генерирует соответствующую этой ситуации цель) и определяет, какая информация (т. е. описания каких сущностей) должна быть включена в высказывание. Компонент генерации высказываний в этом случае выбирает подходящий формат для представления высказывания и конкретизирует его соответствующими данными, возможно, производя при этом морфологический синтез словоформ.

Аналогичный метод может применяться и при генерации большинства отчетов, т. е. высказываний, выражающих результаты решения задач, полученные ЕЯ - системой. Отличие состоит в том, что вместо заранее заготовленных шаблонов используются шаблоны, которые порождаются компонентом генерации высказываний на основе метазнаний, т. е. описаний данных, включаемых в ответ. Заметим, что в ряде случаев, например в системах ЕЯ-общения с базами данных, результатом решения задачи часто является множество данных, формат представления которых определяется средствами генерации отчетов СУБД. Очевидно, что в этих случаях необходимость в компоненте генерации высказывании ЕЯ-системы вообще не возникает.

Генерация высказываний, которые не могут быть описаны с помощью шаблонов, представляет собой более сложную и недостаточно исследованную задачу. К таким

высказываниям, в частности, относятся объяснения, генерируемые в качестве ответов на вопросы, областью интерпретации которых являются абстрактные знания и метазнания системы. Такие вопросы часто задаются неопытными конечными пользователями для того, чтобы получить общее представление о знаниях и возможностях ЕЯ-системы, например: «Какого рода данные содержатся в базе данных?», «Что такое издержки производства?», «Какая разница между общезаводскими накладными расходами и издержками производства?» и т. п. Основная особенность подобных вопросов заключается в том, что они не дают точного представления о необходимой информации в ответе. Как правило, содержание ответа (текста объяснения) зависит от ситуации, в которой задан вопрос, а зачастую и от познаний пользователя в данной области.

Для генерации объяснений, т. е. ответов на вопросы метауровня и абстрактного уровня, требуется достаточно тонкая классификация целей создания ЕЯ-текстов. Организация текста (т.е. информации, сообщаемой пользователю, уровень ее общности, разбиение на части, соответствующие будущим предложениям и т.п.) определяется так называемыми дискурсными целями, такими, как, например: дать определение, описать, сравнить, подтвердить, уточнить и т.д. Каждой дискурсной цели может быть сопоставлен определенный способ организации текста, называемый дискурсной стратегией. Предполагается, что, выявив дискурсные стратегии, используемые людьми, и выразив их в подходящем формальном представлении, можно получить достаточно надежный и эффективный механизм генерации смыслов порождаемых текстов.

Один из возможных методов представления дискурсных стратегий применяется в системе TEXT. Данная система генерирует объяснения в виде отдельных абзацев и может отвечать на вопросы следующих типов: вопросы на определения (дефиниции) объектов; вопросы о различии между объектами; вопросы на описания объектов; вопросы относительно имеющейся в базе данных информации. Дискурсные стратегии построения объяснений были получены эмпирически в результате изучения коротких определений и сопоставлений в различных справочниках и энциклопедиях. Каждая дискурсная стратегия представляется в системе TEXT в виде грамматики (дерева составляющих), терминалами которой служат так называемые риторические предикаты (например, идентификация объекта как члена какого-то класса, представление свойств (атрибутов) объекта и т. п.), определяющие тип информации из системной модели данных, которая может быть использована для их конкретизации.

Изложенные выше методы не претендуют на универсальность. Отсутствие общих методов решения задач генерации (в основном это касается задач генерации смысла высказываний) объясняется как сложностью и недостаточной изученностью проблемы, так и тем обстоятельством, что в силу исключительной способности человека понимать «плохо построенные» ответы наибольшее внимание при разработке ЕЯ-систем уделяется, как правило, компоненту понимания высказывания и диалоговому компоненту.

Лекция 4. Гибкость и настройка ЕЯ-систем

Рассмотренные в предыдущей лекции методы позволяют создавать ЕЯ-системы,

обеспечивающие эффективное удовлетворение информационных потребностей конечных пользователей относительно ограниченных проблемных областей. Ограниченность проблемной области имеет принципиальное значение, так как позволяет наложить вполне «естественные» для каждой из проблемных областей ограничения на лексику, синтаксис, семантику и прагматику языка общения. Это дает возможность рассматривать процесс ЕЯ-общения не во всем многообразии и сложности, а в более узкой (и, следовательно, более простой) постановке. Вместе с тем, чтобы обеспечить общение относительно различных проблемных областей, каждая из которых характеризуется собственными, отличными от других областей ограничениями, а также проблемных областей, изменяющихся во времени, ЕЯ-система должна допускать возможность изменения (модификации) своих знаний, зависящих от специфики проблемной области, в которой функционирует система.

Процесс извлечения знаний из некоторого источника знаний и передачи их ЕЯ-системе называют *приобретением знаний.* В качестве источника знаний может выступать как человек, так и текст, в котором содержатся сведения о проблемной области. Если знания ЕЯ-системы не могут приобретаться без участия ее разработчиков (или программистов), то такие системы называются *жесткими.* Очевидно, что жесткие ЕЯ-системы не удовлетворяют требованиям, предъявляемым к средствам общения конечных пользователей.

Гибкие ЕЯ-системы допускают возможность приобретения знаний без привлечения программистов или разработчиков. Степень гибкости ЕЯ-системы определяется, по крайней мере, следующими факторами: составом (номенклатурой) приобретаемых знаний; пределами, в которых допускается модификация знаний; инициатором процесса приобретения знаний (пользователь или система); требованиями к уровню профессиональной подготовки лиц, участвующих в процессе приобретения знаний.

Возможности приобретения знаний в гибких ЕЯ-системах основаны на использовании различных способов представления проблемно-независимых и проблемно-ориентированных знаний. Предполагается, что знания ЕЯ-системы могут быть разделены на проблемно-независимые, используемые в любом из возможных приложений, и проблемно-ориентированные, определяемые спецификой конкретных приложений. Проблемно-независимые знания вводятся в ЕЯ-систему при ее разработке. Большая их часть представляется процедурно (т. е. в виде программ). Поэтому они не могут модифицироваться без участия программистов или разработчиков системы. Проблемно-ориентированные знания представляются декларативно (т. е. в виде фреймов, семантических сетей, выражений на языке исчисления предикатов, продукций и т.д.). Наиболее важное общее свойство декларативных представлений состоит в том, что механизм интерпретации декларативных знаний (т. е. программы ЕЯ-системы) не зависит от содержания этих знаний. Поэтому модификация декларативно представленных проблемно-ориентированных знаний может осуществляться без участия программистов или разработчиков системы.

Таким образом, состав приобретаемых знаний, а также пределы, в которых допускается модификация знаний, в каждой ЕЯ-системе зависят от масштабов использования в ней декларативно представленных знаний. Следует отметить, что применение декларативных

представлений ведет в общем случае к ухудшению временных и объемных характеристик ЕЯ-системы. Поэтому при разработке конкретных ЕЯ-систем, особенно ориентированных на примышленное или коммерческое использование, часто приходится идти на компромиссы, т. е. искусственно «уменьшать» гибкость (за счет сокращения состава декларативно представленных знаний) для получения приемлемых технических характеристик.

Различают два основных способа приобретения знаний: *настройка* и *адаптация.* Настройка обычно выполняется при первоначальном развертывании ЕЯ-системы на конкретном объекте эксплуатации, а также при значительных изменениях проблемной области, относительно которой ведется общение. Как правило, настройка ЕЯ-системы осуществляется в отсроченном режиме. Состав и объем приобретаемых в процессе настройки знаний, а также требования к уровню профессиональной подготовки персонала, осуществляющего настройку, зависят от особенностей конкретной ЕЯ-системы.

Адаптация заключается в оперативном приобретении знаний ЕЯ-системой в процессе решения задач пользователя. Инициатором адаптации обычно является система. При этом, поскольку адаптация выполняется в процессе взаимодействия с пользователем, то никаких специальных требований к его знаниям не предъявляется.

Настройка ЕЯ-систем. В существующих ЕЯ-системах предусматривается настройка по различным параметрам (видам проблемно-ориентированных знаний). К основным из них относятся:

- знания о языке общения и сущностях проблемной области;
- знания о пользователе, решаемых им задачах и структуре диалога, направленного на решение этих задач;
- знания об особенностях прикладной системы, взаимодействие с которой поддерживается ЕЯ-системой.

Большинство ЕЯ-систем имеет определенную прикладную ориентацию (общение с базой данных, диалоговое решение задач, обработка связного текста). Это позволяет исключить необходимость настройки по некоторым из параметров и улучшить временные и объемные характеристики системы. Так, общение с базой данных характеризуется относительной устойчивостью решаемых пользователями задач. Поэтому в большинстве ЕЯ-систем для общения с базой данных предусматривается настройка по первому и третьему параметрам. Для систем обработки связных текстов характерна жесткая «привязка» к средствам хранения информации, поэтому системы данного класса допускают настройку по первому и (в значительной степени) по второму параметрам. Системы диалогового решения задач в принципе должны иметь возможность настраиваться по всем трем указанным параметрам. Однако в настоящее время большинство из них ориентируется на заранее определенные средства решения задач (в частности, на определенные экспертные системы). Поэтому настройка систем диалогового решения задач, как правило, допускается по первому и второму параметрам.

В зависимости от требований, предъявляемых к уровню профессиональной подготовки лиц, осуществляющих настройку, системы ЕЯ-общения могут быть подразделены на

следующие классы:

- системы, для настройки которых требуется знание лингвистики;
- системы, настраиваемые администратором базы данных;
- системы, настраиваемые специалистом в прикладной проблемной области (прикладным специалистом) или непосредственно конечным пользователем.

Системы, составляющие последний класс, получили название «переносимых» (transportable).

Необходимость знания лингвистики для настройки системы характерна для ранних ЕЯ-систем (ПОЭТ, МИВОС, LIFER, SOPHIE и т. п.). Настройка подавляющего большинства современных ЕЯ-систем может выполняться администратором базы данных или прикладным специалистом (т. е. людьми, от которых не требуются знания лингвистики).

Типичным примером ЕЯ-системы, настройка которой осуществляется администратором базы данных, может служить система INTELLECT. Данная система осуществляет поиск и выдачу информации из одного файла базы данных по запросам пользователя на английском языке. Настройка системы заключается в пополнении (или изменении) ее словаря. Словарь содержит слова двух типов: функциональные - они поставляются вместе с системой, и слова, которые вводятся в процессе настройки или адаптации.

К функциональным словам относятся: *что, когда, в, все, какие, сколько, напечатать* и т. п. Семантика этих слов заранее известна системе и не может изменяться при использовании системы в любом из приложений. Фактически функциональные слова являются ключевыми при синтаксическом и семантическом анализе вопросов пользователя.

Слова второго типа отражают специфику конкретной проблемной области, отображаемой в файле базы данных. К ним относятся: *служащий, фамилия, город, зарплата* и т. п. По существу, они являются внешними именами (и, возможно, значениями) атрибутов файла базы данных, общение с которым обеспечивает система. Эти слова, а также их синонимы определяются администратором базы данных (исходя из логической структуры файла) и с помощью специального языка вводятся в систему. Поскольку администратор базы данных не может заранее предусмотреть всех слов, соответствующих значениям атрибутов, то они могут вводиться в словарь в процессе адаптации системы в результате уточняющих поддиалогов, инициируемых системой.

Примерами переносимых систем могут служить такие системы, как АИСТ, TEAM. В отличие от системы INTELLECT они обеспечивают ЕЯ-общение не с одним, а с несколькими связанными файлами базы данных. При этом настройка допускается не только на язык общения и сущности проблемной области, но и на логическую структуру базы данных и тип СУБД (в рамках заданного класса СУБД).

Переносимость налагает на разработчиков ЕЯ-системы достаточно жесткие ограничения, которые в полном объеме в существующих системах пока не выполняются. Во-первых, используемый синтаксис должен задаваться общей грамматикой ЕЯ, а не проблемно-ориентированным множеством правил. Это позволяет исключить из процесса

настройки модификацию грамматики языка общения. Во-вторых, поскольку семантика языка общения не может зависеть от некоторого проблемно-ориентированного синтаксиса, выбранного для упрощения семантического анализа, то должен быть разработан общий механизм приобретения и присоединения семантики к широкому множеству синтаксических конструкций. В-третьих, лексика языка общения (словарь) также должна пополняться в процессе настройки. При этом вся необходимая информация специфицируется в терминах, ориентированных на пользователя или прикладного специалиста. И, наконец, в-четвертых, процесс доступа к базе данных должен иметь возможность использовать приобретаемые в процессе настройки знания для перевода того, что было сказано в запросе пользователя, в запрос, соответствующий логической структуре базы данных. При этом от пользователя и от процесса анализа скрыто, присутствует ли требуемая пользователем информация явно или выводится из информации, имеющейся в базе данных.

Как правило, в переносимых ЕЯ-системах процесс настройки рассматривается как отдельная задача, которая может решаться либо средствами самой ЕЯ-системы (АИСТ, TEAM), либо с помощью специальной, дополнительной ЕЯ-системы. Последнее имеет место в системе IRUS, для настройки которой применяется специальная ЕЯ-система IRACQ. Блок настройки позволяет «переложить» инициативу и значительную часть ответственности за процесс настройки с лица, производящего настройку (пользователя или прикладного специалиста), на систему. При этом пользователь или прикладной специалист избавляются от заботы о полноте и непротиворечивости вводимой в систему информации, а также от необходимости знания особенностей используемых в настраиваемой ЕЯ-системе способов представления проблемно-ориентированных знаний.

Адаптация ЕЯ-систем. Основу процесса адаптации составляет оперативное приобретение знаний, недостающих ЕЯ-системе, для обеспечения эффективного удовлетворения информационных потребностей конечного пользователя в конкретных ситуациях общения. Необходимость адаптации ЕЯ-систем обусловлена тем, что при настройке системы, как правило, не удается предусмотреть все будущие изменения проблемной области.

По сравнению с настройкой адаптация является более сложным процессом, так как оперативный (т. е. в процессе решения задач пользователя) характер ее выполнения требует в общем случае многоаспектной интерпретации высказываний пользователя (т. е. интерпретации на несколько областей знаний системы). В настоящее время эта задача еще не имеет общего решения, которое может быть рекомендовано для практического применения. В связи с этим возможности адаптации действующих ЕЯ-систем, как правило, весьма ограниченны. В большинстве существующих ЕЯ-систем предусматривается адаптация к терминологии, используемой в высказываниях пользователя. Следует также отметить возросший в последнее время интерес к адаптации ЕЯ-систем к уровню квалификации пользователя и решаемым им задачам.

Рассмотрим адаптацию к терминологии пользователя. Обычно она выполняется в процессе уточняющего поддиалога, инициируемого системой при обнаружении в

высказывании пользователя неизвестных ей слов. При этом система использует заранее заготовленный сценарий уточняющего поддиалога. Приведем три типовых сценария, получившие наиболее широкое распространение в современных ЕЯ-системах.

Сценарий первый. Пользователь «приравнивает» новый термин (или фразу) к структурам, уже известным системе. Например:

П: Кто преподает дискретную математику?

С: Слово «преподает» неизвестно. Уточните.

П: Преподает означает то же, что и ведет занятия.

С: Дискретную математику преподает доцент Петров.

Сценарий второй. Система с помощью последовательности простых вопросов предлагает пользователю выработать необходимое определение, в результате чего она получает как синтаксическую, так и семантическую информацию о новом термине. Фактически в данном случае управление передается блоку приобретения знании, используемому при настройке ЕЯ-системы. Например:

П: Выдать список студентов отличников?

С: Слово «отличник» неизвестно. Отличник – это студент?

П: Да.

С: Слово «отличник» связано с личными данными, успеваемостью, развлечением.

П: С успеваемостью.

С: Слово «отличник» связано со средним баллом аттестата, со средним баллом зачетной книжки, с оценкой по дисциплине.

П: Со средним баллом зачетной книжки.

С: Средний балл зачетной книжки за текущий семестр, предыдущий, за все семестры.

П: За текущий.

С: Средний балл больше, меньше или равно скольки (указать операцию сравнения и интервал)?

П: Равен 5.

С: Т.о. отличник – студент, у которого средний балл зачетной книжки за текущий семестр равен 5?

П: Да.

С: Напечатать фамилии всех студентов с документа «зачетная книжка», средний балл документа = 5, сессия – зимняя, учебный год – 2003-2004.

Сценарий третий. Он основан на предположении, что неизвестный термин представляет собой значение, имеющееся в базе данных. Такое предположение оправдано, если ЕЯ-система сопряжена с большой, быстро изменяющейся базой данных. В этом случае нецелесообразно включать в словарь все значения базы данных, а смысл неизвестного слова может рассматриваться как имя поля (атрибута) базы данных, чьим значением служит данное слово. Примером системы, использующей этот сценарий (вместе с первым из рассмотренных сценариев), может служить ЕЯ-система INTELLECT. Приведем пример уточняющего диалога, использующего данную стратегию.

П: Сообщите имена бухгалтеров в ВСГТУ.

С: Слово «ВСГТУ» системе неизвестно.

Если вы предполагаете найти это слово в базе данных, нажмите клавишу «Возврат». В противном случае исправьте его написание или введите синоним.

П: <нажимает клавишу «Возврат»>

С: В каком поле оно должно появиться?

П: Название учебного заведения.

С: Напечатать фамилии всех служащих с названием учебного заведения - ВСГТУ и должность-бухгалтер.

Число записей в отчете =14: Бадмаева Д. Р., Захарова Г. М., ...

В заключение отметим, что для действующих ЕЯ-систем характерно использование не какого-либо одного, а сразу нескольких из рассмотренных сценариев. Поэтому при обнаружении незнакомого термина система предоставляет пользователю возможность выбора одного из имеющихся у нее сценариев.

В данном разделе были рассмотрены архитектура, состояние развития современных ЕЯ-систем, методы реализации основных компонентов ЕЯ-систем: диалогового, понимания и генерации высказываний. Выделены основные задачи, методы решения каждой задачи, а также перспективы исследований в данном направлении. В качестве примеров приводились методы, лежащие в основе уже существующих ЕЯ-систем.

Ядром любой естественно-языковой системы является лингвистический процессор, структура и задачи которого более подробно рассмотрены в следующем разделе.

РАЗДЕЛ 2. ЛИНГВИСТИЧЕСКИЙ ПРОЦЕССОР - ЯДРО ЕЯ-СИСТЕМЫ

Лекция 5. Лингвистический процессор

Назначение лингвистического процессора

Попытки формализовать интеллектуальную деятельность человека привели к постановке фундаментальной лингвистической задачи, состоящей в моделировании его языкового поведения, т.е. в построении функциональной модели естественного языка. Естественный язык служит человеку для выражения собственных мыслей и для понимания мыслей других людей. Первому виду языковой деятельности соответствует производство ЕЯ-текстов, а второму - понимание таких текстов. Если обозначить множество текстов через {Т}, а множество выражаемых ими смыслов через {С}, то модель естественного языка можно определить как транслятор, устанавливающий соответствие между этими двумя множествами: {Т} $\Leftrightarrow$ {С}.

Формальные модели языка рассматриваются как компоненты различных прикладных ЕЯ-систем. Компонента системы, реализующая формальную лингвистическую модель и способная работать с ЕЯ во всем его объеме, называется *лингвистическим процессором* (ЛП).

Две основные функции ЛП состоят в извлечении смысла из заданного текста и в выражении заданного смысла текстом на ЕЯ, иначе это функции:

- моделирования понимания (анализ);
- моделирования производства текстов (синтез).

Формальная модель, лежащая в основе ЛП, является наиболее полной моделью класса «Смысл $\Leftrightarrow$ Текст». Такая модель обеспечивает получение связных синтаксических структур для всех предложений обрабатываемых текстов, независимо от степени их сложности, и переработку текстов на естественном языке без смысловых потерь.

Структура и состав лингвистического процессора

Со стороны своего внутреннего устройства лингвистический процессор представляет собой многоуровневый преобразователь. В нем различаются три уровня пофразного представления текста - морфологический, синтаксический и семантический. Каждый из уровней обслуживается соответствующим компонентом модели - массивом правил и определенным словарем. На каждом из уровней предложение имеет формальный образ, именуемый в дальнейшем его структурой - морфологической (МорфС), синтаксической (СинтС) и семантической (СемС). Синтез представляет собой обратный переход от СемС предложения к его записи в обычном орфографическом виде. Структура лингвистического процессора представлена на рисунке 12.

Под *морфологической структурой* понимается последовательность входящих в анализируемое предложение слов с указанием части речи и морфологических характеристик (падежа, числа, рода, одушевленности, вида и т.п.).

Под *синтаксической структурой* понимается дерево зависимостей, в узлах которого стоят слова данного естественного языка с указанием части речи и грамматических характеристик, а дуги соответствуют специфичным для данного естественного языка

отношениям синтаксического подчинения.

Под *семантической структурой* понимается дерево зависимостей, в узлах которого стоят либо предметные имена, либо слова универсального семантического языка, а дуги соответствуют универсальным отношениям семантического подчинения, таким, как аргументное, атрибутивное, конъюнкция, дизъюнкция, равенство, неравенство, больше, меньше, принадлежит и т.п. Существенным компонентом СемС является информация о кореферентности узлов, т.е. информация о том, в каких случаях речь идет об одном и том же объекте, а в каких - о разных.

Рис. 12. Структура лингвистического процессора

Лингвистический процессор в целом должен обеспечивать выполнение следующих преобразований:

предложение на ЕЯ ⇒ МорфС ⇒ СинтС ⇒ СемС (при анализе)
СемС ⇒ СинтС ⇒ МорфС ⇒ предложение на ЕЯ (при синтезе)

Таким образом, чтобы построить ЛП, необходимо разработать:

- формальные языки для записи (образов) предложений на морфологическом, синтаксическом, семантическом уровнях представления;
- формальное понятие структуры предложения для каждого из этих уровней;
- массивы правил для преобразования структур смежных уровней друг в друга;
- морфологический, синтаксический и семантический словари, включив в них всю информацию о каждой лексеме, необходимую для осуществления

соответствующего преобразования.

Анализ ЕЯ-текстов в лингвистическом процессоре

Цель анализа предложения на естественном языке - перевод их на М-язык ВС. Функциями анализатора являются:

- распознавание правильно построенных предложений ЕЯ;
- фиксация, локализация и возможность исправления ошибок в ЕЯ-тексте;
- декомпозиция предложения на составляющие (фрагменты) и построение соответствующей синтаксической структуры предложения;
- семантическая интерпретация фрагментов ЕЯ-предложения во фрагменты М-языка;
- композиция фрагментов М-языка в структуру, описывающую ситуацию проблемной среды.

Реализация этих функций осуществляется на этапах морфологического и синтаксического анализов, семантической интерпретации и проблемного анализа. Во многих моделях ЛП два последних этапа объединяются в один этап семантического анализа.

Синтез фраз ЕЯ-текстов в лингвистическом процессоре

В большинстве случаев вместо полного синтеза используется синтез по шаблонам. Суть его состоит в том, чтобы для конкретной системы рассмотреть все типы сообщений, относящиеся как к процессу общения, так и к процессу выдачи результатов работы ВС, и для каждого типа разработать шаблон, который заполняется при обращении к пользователю.

Задача синтеза заключается в переводе «текста» М-языка в ЕЯ-текст и состоит из следующих этапов:

- определение информации, которую нужно сообщить пользователю;
- определение уровня общности синтезируемой информации;
- выделение обязательной и необязательной информации, выражаемой в синтезируемых фразах;
- разбиение текста М-языка на фрагменты, соответствующие будущим фразам;
- определение лексем для синтезируемой фразы;
- построение синтаксической структуры фразы;
- приписывание морфологической информации вершинам синтаксической структуры фразы;
- определение порядка слов;
- осуществление морфологического синтеза лексем.

Суть семантического синтеза заключается в таком преобразовании текста М-языка, при котором его части могли бы соответствовать будущим фразам и предложениям ЕЯ. При этом требуется учет как языкового, так и смыслового факторов. Фраза должна быть приемлемой по размерам, быть стилистически доступной и т.п. Иногда для этого достаточно использовать простые правила с учетом ограничений, например, на число существительных, на число определений, выражаемых придаточными предложениями, и т.п. Такие

преобразования осуществляются за счет правил фрагментирования текста М-языка. Результатом семантического синтеза будет структура М-языка, разбитая на фрагменты, соответствующие будущим фразам.

Цель синтаксической интерпретации - проинтерпретировать выделенные на предыдущем этапе фрагменты синтаксическими структурами ЕЯ, т.е. определить порядок следования фраз, сформировать их синтаксические структуры и заполнить эти структуры соответствующими лексемами. Выбор лексем может зависеть от истории общения. Например, при работе в системе типа «вопрос - ответ» синтезатор может использовать те лексемы, которые применялись пользователем в вопросе. Не полностью определенные синтаксические структуры подаются на этап синтаксического синтеза.

Задача синтаксического синтеза - конкретизация синтаксических структур с учетом отношений между лексемами. Здесь выбираются форма фраз и морфологические характеристики лексем.

Задача морфологического синтеза - построение конкретных словоформ ЕЯ по словарю и заданной морфологической информации. Морфологический синтез завершает процесс синтезирования, после чего сообщение на естественном языке выдается пользователю.

В данной лекции была рассмотрена архитектура лингвистического процессора, который лежит в основе всех естественно-языковых систем, описаны этапы анализа и синтеза ЕЯ-текстов. Основными задачами анализа ЕЯ-текстов являются морфологический, синтаксический и семантический анализы, поэтому в последующих главах рассмотрены методы, подходы и алгоритмы, позволяющие их реализовывать в существующих лингвистических процессорах.

РАЗДЕЛ 3. МОРФОЛОГИЧЕСКИЙ АНАЛИЗ ЕЯ-ТЕКСТОВ

Лекция 6. Анализ методов и подходов морфологического анализа

Стадия морфологического анализа (МА) является наиболее проработанным лингвистическим этапом процесса обработки естественного текста. За последние два десятилетия создано, по крайней мере, несколько десятков алгоритмов для разных языков, в том числе 10-12 для русского (Г.Г. Белоногов, И.А. Мельчук и др.).

Цель морфологического анализа (МА) заключается в определении морфологической информации словоформ для использования ее на последующих этапах обработки ЕЯ текста. На рисунке 13 изображена классификация морфологических признаков слов русского языка.

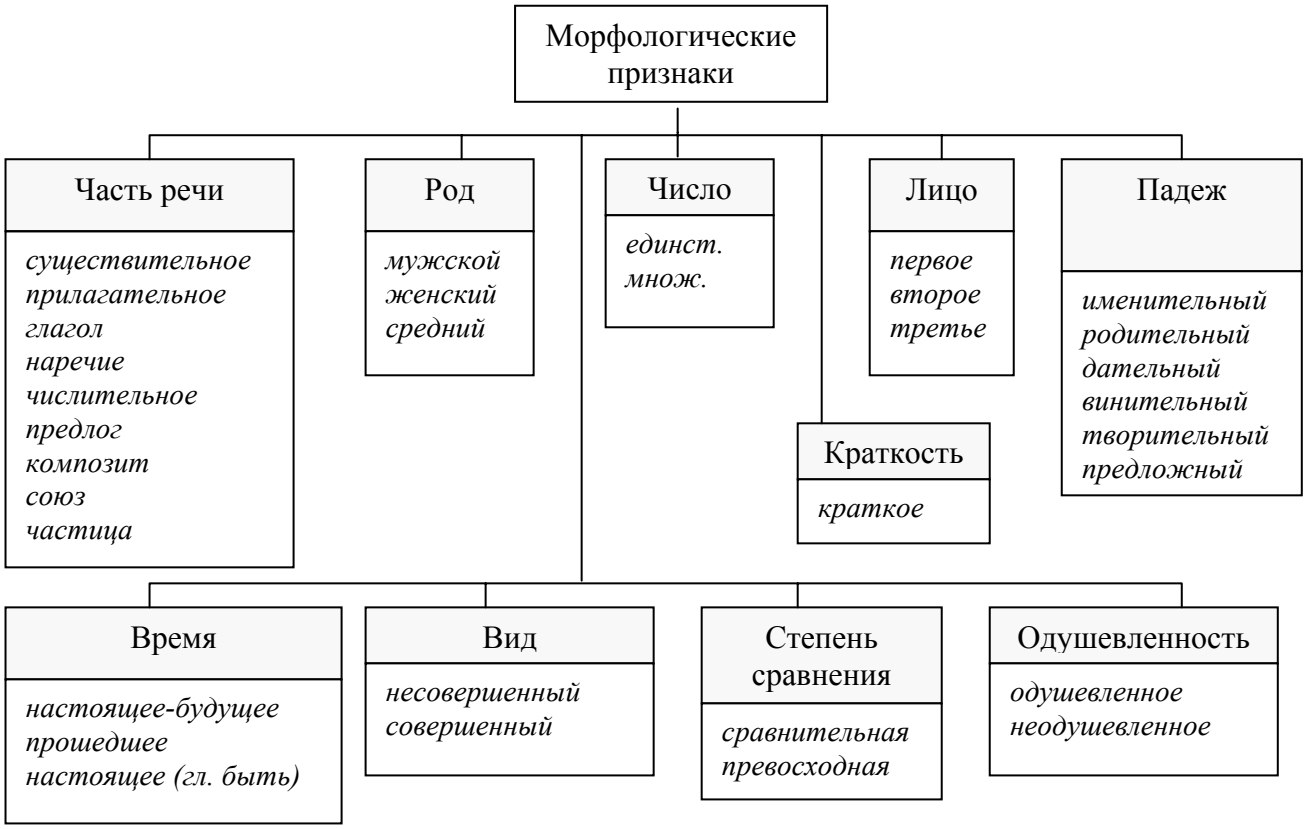


Рис.13. Морфологические признаки слов русского языка

Существуют три основных метода реализации МА: декларативный, процедурный и комбинированный.

При декларативном методе в словаре хранятся все возможные словоформы каждого слова с приписанной им морфологической информацией (МИ). В этом случае задача МА состоит просто в поиске словоформы в словаре и переписывании из словаря МИ [12, 14], поэтому можно считать, что в этом методе отсутствует как таковой морфологический анализ, а хранится только его результат. Так как количество различных словоформ у каждого слова довольно велико, декларативный метод требует больших затрат памяти ВС, что порождает ряд технических проблем, заключающихся в больших затратах труда на создание и поддержание словаря, в высокой избыточности информации. Достоинствами метода

является высокая скорость анализа, а также универсальность по отношению к множеству всех возможных словоформ русского языка.

Процедурный МА выполняет следующие функции: выделяет в текущей словоформе основу, идентифицирует ее и приписывает данной словоформе соответствующий комплекс МИ. Процедурный метод предполагает предварительную систематизацию морфологических знаний о ЕЯ и разработку алгоритмов присвоения МИ отдельной словоформе. Недостатком такого подхода является высокая трудоемкость составления словарей совместимости. При этом наличие в русском языке большого числа слов-исключений не позволяет сколько-нибудь автоматизировать этот процесс. Для проведения анализа словоформы необходимо наличие словарей «приставка-корень», «корень – суффикс - флективный класс», «флективный класс – окончание - морфологическая информация».

Работающая система, в которой реализован процедурный морфологический анализ, занимает значительно меньший объем памяти, но при этом увеличивается время поиска МИ за счет разбиения словоформы на составляющие и применения процедур совместимости. Исходя из этого, процедурный метод удобнее применять в системах с относительно небольшим количеством пользователей, в то время как декларативный – в системах с частым обращением к лингвистическому анализатору. Другим существенным недостатком процедурных методов является отсутствие универсальности, т.к. существует большое количество слов, которые нельзя представить в виде суммы неизменной основы и аффиксов.

В системах реальной степени сложности чаще используется комбинированный вариант морфологического анализа. При этом используется как словарь словоформ, так и словарь основ. На первом этапе проводится поиск по словарю словоформ, как при декларативном методе, и в случае успешного поиска анализ на этом завершается. В противном случае задействуется словарь основ и процедурный метод анализа.

В комбинированном методе, реализованном фирмой «Интелтек Плюс» словоформа разделяется на основу и аффикс (окончание и, возможно, суффикс), и словарь содержит только основы слов вместе со ссылками на соответствующие строки в таблице возможных аффиксов, причем основа должна оставаться неизменной во всех возможных словоформах данного слова. За счет использования словаря готовых словоформ обеспечивается достаточно высокая скорость определения МИ, а за счет справочников “основа+аффикс” – качество морфологического анализа.

Лекция 7. Анализ существующих моделей морфологического анализа

В настоящее время выделилось несколько направлений в разработке морфологического анализа.

Одно из них моделирует классическую схему анализа путем разделения словоформы на основу и аффиксы (приставку, суффикс, окончание) с последующей проверкой на совместность окончания с остающейся основой.

К данному направлению относится модель морфологического анализа Г.Г. Белоногова, в основе которой лежит флективный анализ слов, базирующийся на разбиении лексем (слов) русского языка на флективные классы (табл. 4).

Часть речи представляет собой классы слов языка, выделяемые на основании сходства их синтаксических, морфологических и логико-семантических свойств. Каждой части речи свойствен свой набор грамматических категорий, причём этим набором охватывается абсолютное большинство слов данной части речи. Многие слова, относящиеся к одной и той же части речи, могут быть сгруппированы в отдельный флективный класс (ФК), который описывает закон их словообразования, т.к. для характеристики системы окончаний слова-представителя ФК нет необходимости перечислять окончания всех его форм, а достаточно это сделать для нескольких типичных форм. По флективному классу при морфологическом анализе определяют постоянные параметры слова (для существительного - род и одушевленность, для других частей речи – часть речи).

Таблица 4

Группы и подгруппы флективных классов

Номер группы	Номер подгруппы	Наименование подгруппы
1	1	Существительные м.р., неодушевленные
	2	Существительные м.р., одушевленные
	3	Существительные ж.р., одушевленные
	4	Существительные ж.р., неодушевленные
	5	Существительные среднего рода
2	0	Прилагательные
3	0	Глаголы в личной форме
4	1	Глаголы прошедшего времени
	2	Краткие прилагательные
5	0	Количественные числительные
6	1	Неизменяемые слова, входящие в СГФ
	2	Неизменяемые слова, не входящие в СГФ

ФК изменяемых слов выделяется на основе анализа их синтаксических функций и систем падежных и родовых окончаний, а ФК неизменяемых слов – только по синтаксическому принципу.

Изменяемые слова с учетом их синтаксической функции объединены в следующие группы:

- существительные;
- прилагательные;
- глаголы в личной форме;
- глаголы прошедшего времени и краткие прилагательные;
- количественные числительные.

Группа «существительных» разбита на несколько подгрупп, выделенных по признакам рода и одушевленности. Причастия (полные и краткие) относятся к прилагательным (полным и кратким). К группе «прилагательные», наряду с полными прилагательными (полными причастиями), относятся порядковые числительные, субстантивированные прилагательные, а также количественное числительное «один».

В группе «неизменяемых слов» выделены две подгруппы:

- неизменяемые слова, включенные в словарь готовых словоформ (СГФ);
- неизменяемые слова, образующиеся на базе словаря корневых морфем.

ФК словоформы кодируется четырехзначным числом. В первом разряде кода ФК указывается номер группы, в которую входит данная словоформа, во-втором – номер группы. Два младших разряда кода используются для нумерации ФК внутри каждой подгруппы. Например, в таблице 5 приведены ФК существительных.

В словаре готовых словоформ хранятся слова, имеющие постоянную форму и относящиеся к подгруппе 6-1 (табл. 4) : предлоги, союзы, частицы, вводные слова и т.д. Эти слова не требуют морфологического анализа, так как вся информация о них (часть речи, синтаксическая функция) определены ФК, который указывается в СГФ для каждой словоформы (табл. 6). Изменяемые слова, а также неизменяемые слова, не содержащиеся в СГФ, разбиваются на составляющие их морфемы: префиксы, корни, суффиксы и окончания. Корневые морфемы собраны в словаре корней (табл. 7).

Таблица 5

Флективные классы существительных

ФК	Окончания	Пример	ФК	Окончания	Пример
неодушевленный, мужской род					
1101	_,_,ом,ы,ов,ы ?	телефон	1110	_,_,ем,и,ей,и ?	путь
1102	_,_,ом,и,ей,и ?	тираж	1111	й,й,ем,я,ев,я ?	край
1103	_,_,ем,и,ей,и ь	огонь	1112	_,_,ом,я,ев,я ?	брус
1104	й,й,ем,и,ев,и ?	перебой	1113	_,_,ом,а,_,а ?	глаз
1105	й,й,ем,и,ев,и и	санаторий	1114	_,_,ем,и,ей,ей ?	зародыш
1106	_,_,ом,и,ов,и ?	бланк	1115	_,_,ом,ы,_,ы ?	волос
1107	_,_,ом,и,_,и ?	сапог	1116	_,_,ем,я,ей,я ь	лагерь
1108	_,_,ом,а,ов,а ?	лес	1117	_,_,ю,и,ев,и ?	ложь
1109	_,_,ем,ы,ев,ы ?	колодец			
одушевленный, мужской род					
1118	а,_,ами,а,_,_ ?	ребята	1208	_,а,ом,и,ов,ов ?	сапожник
1200	_,а,ом,ы,ов,ов ?	кузнец	1209	_,а,ем,ы,ев,ев ?	испанец
1201	_,а,ом,ы,_,_ ?	солдат	1210	а,у,ей,и,ей,ей ?	юноша
1202	_,а,ом,и,ей,ей ?	сосед	1211	а,у,ой,ы,_,_ ?	мужчина
1203	_,а,ом,и,ов,ов ж	враг	1212	я,ю,ей,и,ей,ей ь	судья
1203	_,а,ом,и,ов,ов ч	враг	1213	_,а,ем,и,ей,ей ?	товарищ
1203	_,а,ом,и,ов,ов ш	враг	1214	_,а,ом,е,_,_ ?	гражданин
1203	_,а,ом,и,ов,ов щ	враг	1215	_,а,ом,а,ов,ов ?	профессор
1204	й,я,ем,и,ев,ев ?	пролетарий	1216	_,а,ем,я,ей,ей ?	муж
1205	ей,я,ем,и,ев,ев ?	воробей	1217	_,а,ым,ы,ых,ых ?	Иванов
1206	_,я,ем,и,ей,ей ь	конь	1218	_,а,ом,я,ей,ей ?	сын
1207	_,я,ем,я,ей,ей ь	учитель	1219	_,а,ом,а,_,_ ?	хозяин
			1220	_,а,ом,я,ев,ев ?	брат
одушевленный, женский род					
1300	а,у,ой,ы,_,_ ?	женщина	1305	а,у,ой,и,_,_ ?	санитарка
1301	а,у,ей,ы,_,_ ?	переводчица	1306	_,_,ю,и,ей,ей ь	мышь
1302	я,ю,ей,и,й,й и	нутрия	1307	а,у,ой,ы,ых,ых ?	Иванова
1303	я,ю,ей,и,й,й е	швея	1308	_,_,ью,и,ей,ей ь	дочь
1304	я,ю,ей,и,_,_ ?	цапля	1309	а,у,ей,и,_,_ ?	билетерша
неодушевленный, женский род					
1400	_,_,ю,и,ей,и ч	речь	1405	я,ю,ей,и,й,и н	линия
1400	_,_,ю,и,ей,и ж	ложь	1406	я,ю,ей,и,й,и е	галерея
1400	_,_,ю,и,ей,и ш	вошь	1407	я,ю,ей,и,ь,и ?	земля
1400	_,_,ю,и,ей,и щ	мощь	1408	я,ю,ей,и,ий,и ?	эскадрилья
1401	_,_,ю,и,ей,и ь	грань	1409	я,ю,ей,и,ей,и ь	статья
1402	а,у,ой,ы,_,ы ?	колба	1410	я,ю,ей,и,_,и ?	башня
1403	а,у,ей,и,_,и ?	задача	1411	а,у,ей,ы,_,ы ?	улица
1404	а,у,ой,и,_,и ?	заготовка	1412	и,и,ями,и,ей,и ?	бигуди
неодушевленный, средний род					

Продолжение таблицы 5

1500	о,о,ом,а, ,а ?	место	1507	е,е,ем,я,ий,я ?	побережье
1501	о,о,ом,а,ов,а ?	облако	1508	о,о,ом,и,ей,и ?	окно
1502	е,е,ем,я,ей,я ?	поле	1509	о,о,ом,и,ов,и ?	очко
1503	е,е,ем,я,й,я ?	сомнение	1510	е,е,ем,я,ей,я ?	ружье
1504	е,е,ем,а, ,а ?	жилище	1511	о,о,ом,и, ,и ?	колено
1505	о,о,ом,я,ев,я ?	перо	1512	е,е,ем,я,ев,я ?	платье
1506	я,я,ем,а, ,а ?	время	1513	е,е,ем,а,ев,а ?	блюдце
одушевленный, средний род					
1514	и,ей,и,и,ей,ей ?	дети	1603	ое,ое,ым,ые,ых,ых ?	животное
другие ФК					
1600	ый,ого,ым,ые,ых,ых ?	бездомный	1605	ая,ую,ой,ые,ых,ые ?	ванная
1601	ий,его,им,ие,их,их ?	нищий	1606	ая,ую,ой,ие,их,ие ?	мастерская
1602	ой,ого,ым,ые,ых,ых ?	больной	1607	ий,ий,им,ие,их,ие ?	английский
1604	ее,ее,им,ие,их,их ?	болеутоляющее			

Таблица 6

Словарь готовых словоформ

Номер п/п	Слово- представитель	ФК
1	в	6108
2	на	6111
. . .	. . .	. . .

Таблица 7

Словарь корней

№ п/п	Корень
. . .	. . .
56	ключ
. . .	. . .

Словообразующие аффиксы (префиксы и суффиксы) и словоизменительные аффиксы (окончания) собраны в списки.

Списки префиксов и суффиксов имеют одинаковую структуру. Они содержат порядковый номер аффикса, его буквенный код и отсылку, указывающую на номер вкладывающегося в него аффиксы, меньшего по количеству букв (табл. 8 и 9).

Таблица 8

Структура списка префиксов

№ п/п	Буквенный код префикса	Отсылка
. . .	. . .	. . .
27	пере	0
28	перео	27
29	перес	28
. . .	. . .	. . .

Таблица 9

Структура списка суффиксов

№ п/п	Буквенный код суффикса	Отсылка
. . .	. . .	. . .
5	а	0
6	н	0
7	нн	6
. . .	. . .	. . .

В списке окончаний приводятся все окончания, существующие в русской грамматике, и указывается их порядковый номер (табл. 10).

Все фактически возможные сочетания корневых морфем со словообразующими аффиксами отображаются в таблицах совместимости.

В таблице 11 каждому корню сопоставляется список номеров префиксов, с которыми допускается комбинировать этот корень. В таблице 12 для каждого возможного сочетания корня с суффиксами указывается ФК получаемого при этом слова.

Таблица 10

Список окончаний

№ п/п	Окончание
1	а
2	е
…	…
62	ыми

Таблица 11

Совместимость корня с префиксом

Номер корня	Список префиксов
. . .	. . .
56	0, 16, 27, . . .
. . .	. . .

Таблица 12

Совместимость корня с суффиксами

Номер суффикса	Номер корня		
	1	2	. . .
0	1306	1201	. . .
1	-	2003	. . .
. . .	. . .	. . .	. . .

Морфологическая информация отдельных форм слов, рассматриваемых вне контекста, обычно бывает многозначна. Поэтому отдельным словоформам сопоставляются наборы морфологической информации.

Морфологическая информация представлена в закодированном виде. Последовательность кодируемых характеристик для каждой синтаксической группы следующая:

- для существительных: число, падеж;
- для прилагательных: число, род, падеж;
- для глаголов в личной форме: число, лицо;
- для глаголов прошедшего времен: число, род;
- для количественных числительных: падеж.

Структура морфологической таблицы показана в таблице 13.

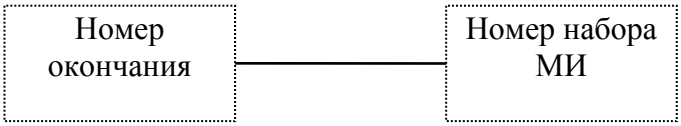
Таблица 13

Структура морфологической таблицы (а) и ее элемента (б)

а)

Флективный класс		
1100	1101	. . .
0 – 2	0 – 2	. . .
. . .	. . .	. . .

(б)



Анализ словоформ

В морфологическом анализе минимально формально выделяемой единицей связного текста считается лексема (слово). В составе лексемы различают корневые морфемы, префиксы, суффиксы и окончания. Известно, что корень несет основную смысловую нагрузку. Однако замена этих префиксов на другие приводит к изменению смысла, а замена аффиксов - к изменению синтаксической функции. Поэтому в методе осуществляется отождествление не только корней, но и словообразующих морфем.

Морфологический анализ словоформ начинается с декларативного способа. При успешном завершении поиска из словаря готовых словоформ извлекается код флективного класса, соответствующий данной лексеме и указывающий на часть речи и синтаксическую функцию словоформы. На этом морфологический анализ рассматриваемого слова заканчивается, и осуществляется переход к обработке следующей лексемы. В случае процедурного способа словоформа подвергается флективному анализу. Флективный анализ включает в себя три этапа:

- идентификация морфем словоформы (последовательная проверка возможностей вложения в анализируемую словоформу корня, суффикса, окончания и приставки);
- определение флективного класса словоформы (извлечение кода ФК из таблиц совместимости корня со словообразующими аффиксами);
- присвоение словоформе морфологической информации.

Все эти этапы тесно взаимосвязаны между собой, так как неудачное завершение второго и третьего этапов свидетельствуют о некорректном разбиении анализируемой словоформы на морфемы. На первом этапе идентифицируются следующие морфемы слова: корень; словообразующие аффиксы.

С целью выделения морфем выполняется последовательная проверка возможностей вложения в анализируемую словоформу справа налево суффиксов и окончаний, и слева направо префиксов.

Другое направление использует информацию, содержащуюся в конечных буквосочетаниях (эта информация получается в результате предварительной статистической обработки словаря). Этот путь также дает достаточно хорошие результаты для практических целей.

Третье направление развивается в последние годы. Оно вызвано стремлением преодолеть ограниченность существующих алгоритмов морфологического анализа. Известно, что они ориентировались на тексты определенной тематики и поэтому не полностью учитывали все особенности морфологии. Это направление пытается построить более адекватные морфологические модели. Создаются универсальные математические модели в форме открытой системы уравнений, позволяющих путем вычисления осуществлять нормализацию словоформ, получение грамматической информации и синтез словоформ. Одной из таких моделей является модель Ю. П. Шабанова-Кушнаренко [15], моделирующая процессы русского языка посредством языка алгебры конечных предикатов,

с помощью которого может быть математически описан любой аспект морфологии русского языка. В данной модели текст рассматривается как многоуровневая конструкция: из букв слагаются морфы, из морф – словоформы, из словоформ – предложения, из предложений – абзацы и т.д. Отдельные части этой конструкции – буквы, морфы, словоформы, предложения и т.д. называют фрагментами текста, а фрагментное отношение L(X, Y), у которого в роли переменной Y выступает часть слова (например, буква, морфема) или целое слово, называется морфологическим отношением. Описание морфологических отношений производится на языке алгебры конечных предикатов. В силу принципа однозначности любое морфологическое отношение L(X, Y) есть функция зависимости фрагмента Y от его смысла X, поэтому иногда морфологическое отношение называют морфологической функцией.

Морфологическая функция представляется в виде функции Y = F(X). В качестве переменной Y используются такие понятия, как основа словоформы, окончание, суффикс и т.д., а в качестве структуры переменной X- часть речи, род, число, падеж и т.д.

Однако данная модель распространяется лишь на небольшую часть механизма склонения имен существительных и прилагательных. Многие явления, непосредственно относящиеся к процессу склонения, не описываются моделью, что может привести к некорректному разбиению на фрагменты словоформы и, как следствие, неверному морфологическому анализу.

Другим подходом при создании универсальных математических моделей МА является построение адекватных формальных моделей с учетом всех фактов языка. Рассмотрим модель морфологии системы «Смысл – Текст», представляющую интерес с точки зрения реализации данного подхода.

Эта модель в отличие от предыдущих ориентирована на синтез словоформ. В общем виде правила морфологического синтеза выглядят следующим образом:

$$(\lambda, \chi) \rightarrow v,$$

где λ - символ лексемы; χ - морфологические характеристики, v – словоформа.

В модели используются семь промежуточных уровней:

- глубинно-морфологическое представление;
- укрупненная морфологическая схема;
- морфемная схема;
- поверхностно-морфологическое представление;
- цепочка не чередованных морфем;
- цепочка чередованных морфем;
- орфографическая словоформа.

Преобразованию при переходе с одного уровня на другой подвергается тройка (λ_i, χ_i ,ε_i), где λ_i – некоторая часть синтактики (точнее, морфологического описания) лексемы, отображающая и заменяющая лексему на i-том уровне, χ_i – релевантная для i-го уровня часть характеристик, ε_i – соответствующая i-му уровню формируемая цепочка символов.

На первом этапе происходит обращение к словарной статье лексемы и переработка в

ней информации. При этом правила выбора основной морфы присоединяются ко всем остальным правилам преобразования элементов цепочек ε_i, а из морф, характеристики χ и синтактики ξ компонуется новый вектор f, называемый грамматической характеристикой. На этом же этапе подготавливаются векторы Φ_i, состоящие из значений признаков вектора f, релевантных для этого этапа преобразования цепочки ε_i. По сути дела, Φ_i объединяет в себе λ_i и χ_i (как правило, в разные Φ_i входят значения разных признаков вектора f).

Однотипный алгоритм синтеза переводит двойку (Φ_i, ε_i) (i = 0, 1, …) в ε_{i+1}, затем к последней присоединяется заранее сформированный вектор Φ_{i+1}, после чего цикл формирования цепочки следующего уровня повторяется вплоть до формирования словоформы.

Морфологические правила при этом делятся на три группы:

1 – описание недопустимых вариантов характеристик;

2 – осуществляют компоновку признаков морфологической характеристики и синтактики в грамматическую характеристику f;

3 – основная часть правил, осуществляющих преобразование элементов цепочек ε_i.

В рассмотренной модели предлагается единая форма таких правил:

$$Q \mid\!- \; A\sigma_i B \rightarrow A\xi_{i+1} B,$$

где |- - разделительный знак;

Q – условие применимости правила в виде ДНФ, элементами конъюнкций в которой служат утверждения относительно значений признаков упомянутой грамматической характеристики;

σ_i – заменяющий символ;

ξ_{i+1} – возникающая подцепочка символов, иногда пустая;

A, B – релевантный внутрицепочечный контекст, т.е. другие подцепочки, которые в частном случае могут содержать и символы, возникающие на рассмотренном этапе синтеза.

Морфологическое описание лексемы состоит из правил выбора основной морфы и лексемной синтактики ξ, сжато характеризующей правила выбора аффиксальных морф для всех включенных в парадигму данной лексемы словоформ. Правила выбора основы имеют общий вид:

$$Q \mid\!- \; \{\text{основа}\} \rightarrow \alpha,$$

где Q – условие в виде ДНФ из значений признаков грамматической характеристики f_1;

{основа} – символ основной морфемы;

α - цепочка символов более низкого уровня, чем {основа}.

Обычно правило выбора основы является безусловным. При построении данной модели учитывались все возможные факты русской морфологии. Поэтому здесь можно описать любую русскую лексему, а с помощью соответствующего множества правил подстановок – синтезировать любую словоформу этой лексемы.

В данном разделе описаны методы и подходы к проведению морфологического анализа словоформ. Указаны недостатки и достоинства каждого из методов. Рассмотрены модели

МА, каждая из которых предлагает свою, уникальную модель МА. Анализ показал, что наиболее распространенным методом МА является декларативный, что объясняется простотой его алгоритма и удобством кодирования. После морфологического анализа лексеме приписывается кортеж с совокупностью морфологической информации, которая поступает на вход синтаксического анализатора, рассмотренного в следующем разделе.

РАЗДЕЛ 4. СИНТАКСИЧЕСКИЙ АНАЛИЗ ЕЯ-ТЕКСТОВ

Лекция 8. Методы, алгоритмы и подходы синтаксического анализа ЕЯ-текстов

В отличие от морфологического анализа текста синтаксический анализ (СА) - развивающаяся область прикладной лингвистики. Цель синтаксического анализа - автоматическое построение функционального дерева фразы, т.е. нахождение взаимозависимостей между разноуровневыми элементами предложения. Существует достаточно много различных способов синтаксического анализа естественно-языковых текстов, которые можно проанализировать с различных точек зрения. Общая структура классификации способов синтаксического анализа приведена в таблице 14.

Таблица 14

Классификация способов синтаксического анализа

№ п/п	Основание классификации	Группа методов
1	Тип цели	Одноцелевые Многоцелевые
2	Синтаксическая структура	Построение графа зависимостей Построение дерева непосредственных составляющих
3	Формальные теории описания естественного языка	Формально-грамматические методы Вероятностно-статистические методы

С точки зрения цели синтаксического анализа можно выделить два основных подхода: одноцелевой и многоцелевой. При первом подходе для фразы требуется построить одно синтаксическое представление, этот подход характерен для первых алгоритмов синтаксического анализа, когда считалось, что синтаксических средств достаточно для того, чтобы обеспечить правильный анализ фразы, хотя бы для большинства фраз. При втором подходе для фразы требуется получить все те синтаксические представления, которые удовлетворяют определенным соглашениям (все «правильно построенные» представления). Вопрос о том, какое из этих представлений является не только правильно построенным, но и правильным, т.е. соответствующим смыслу анализируемой фразы, в рамках синтаксического анализа не решается.

Одним из основных компонентов лингвистической базы знаний, осуществляющей автоматический синтаксический анализ, является описательная модель синтаксической структуры предложения. Такая модель в значительной степени передает концепцию разработчиков относительно синтаксического уровня анализа: какая именно информация об элементах предложения и их взаимосвязях должна выявляться в процессе анализа, присутствовать в его результатах и какие формы представления ей адекватны. Наиболее общим для разработчиков синтаксических анализаторов является взгляд, что синтаксическое строение предложения можно представить некоторым частично упорядоченным множеством бинарных связей между элементами. Виды и свойства элементов, связей и отношения порядка варьируют в разных моделях.

Представления о бинарных синтаксических связях используются в двух известных моделях синтаксической структуры: графах зависимостей и графах непосредственных

составляющих. В настоящее время эти две формы представления синтаксической структуры остаются основными. Они используются в чистом виде или – очень часто – в смешанных формах, сочетающих в себе свойства обоих графов.

Описание структур в форме классического графа зависимостей хорошо соответствует русской грамматической традиции: оно основывается на понятии бинарного словосочетания в предложении с выделенными главными и зависимыми элементами. Элементы изображаются узлами графа, подчинение одного узла другому – направленными дугами, вследствие чего граф зависимостей является ориентированным графом. Обычно ровно один узел графа в подавляющем большинстве моделей, соответствующий сказуемому, не имеет подчиняющего узла и называется вершиной. Иногда двумя вершинами представляют подлежащее и сказуемое.

Отношение подчинения задает частичный порядок на множестве узлов. Если одному узлу подчиняется сразу несколько узлов, то среди последних порядок не определен: граф зависимостей не передает информацию об относительной степени близости подчиненного слова к главному. В некоторых случаях недостаток этой информации вполне очевиден – сравним, например, граф зависимостей для фразы «*программное обеспечение вычислительной техники и автоматизированных систем»* (рис. 14).

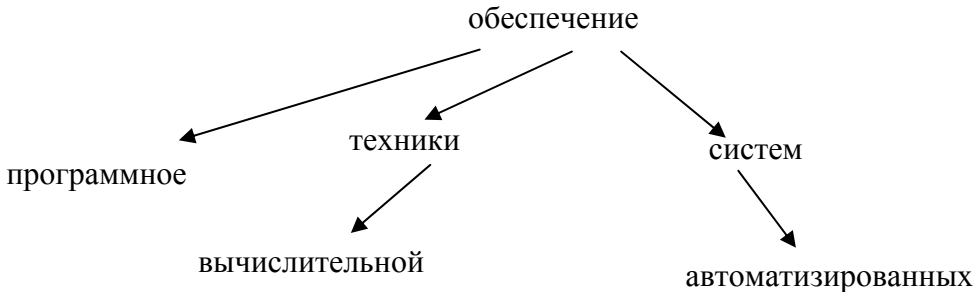


Рис. 14. Граф зависимостей

Как правило, отношение подчинения подразделяется на ряд типов, и дуги графа помечаются индексами синтаксических отношений. К числу редких исключений, когда синтаксическое отношение в графе зависимостей не дифференцируется, относятся системы группы Г.Г. Белоногова.

Иногда граф зависимостей одновременно с отношением подчинения задает и отношение линейного порядка следования узлов. Такой граф называется расположенным. Один из способов изображения такого графа представлен на рисунке 15.

В большинстве случаев отношение подчинения и отношение линейного порядка слов в предложении связаны законом проективности, который при данном способе изображения формулируется так: никакая дуга, исходящая из некоторого узла, не пересекает других дуг или перпендикуляров, опущенных из более верхних узлов.

Особая сложность связана с представлением в древесной структуре явлений однородности. Изображение всех связей однородных членов между собой, с подчиняющими

и подчиненными элементами приводит к возникновению замкнутых контуров в графах зависимостей. Чтобы избежать этого, часто используют представление, при котором сочинительная связь включается в граф зависимостей наравне с другими синтаксическими отношениями (СинО), а подчинительные связи, общие для группы однородных членов, изображаются лишь для одного члена группы (рис. 16).

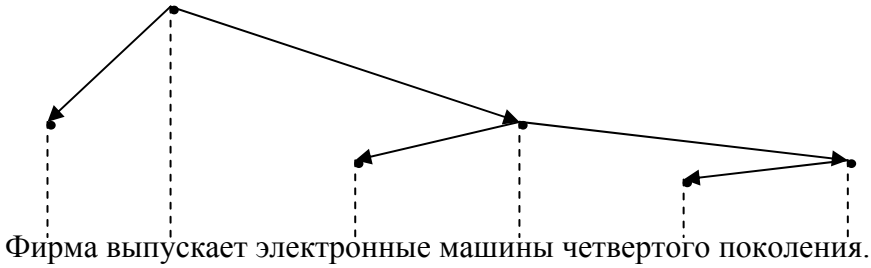


Рис.15. Расположенный граф зависимостей

Так сделано в системе ЭТАП-2 и ряде других систем [10, 11, 13].

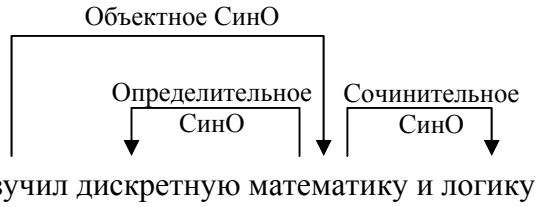


Рис. 16. Представление однородности

Вопрос о допущении недревесности синтаксических графов зависимостей возникает еще и в связи с представлением неоднозначностей синтаксической структуры. Как известно, процедура синтаксического анализа может приводить к построению нескольких вариантов синтаксической структуры предложения. Разные варианты синтаксической структуры могут описываться разными синтаксическими представлениями, в том числе в виде дерева зависимостей. Однако существует и другой подход: например, принципом синтаксического анализа является неразделение на варианты и в результате анализа может получиться недревесный граф зависимостей, в котором сохраняются все виды неоднозначности. В графе зависимостей системы ПОЭТ допускаются неоднозначные зависимости, но только внутри именных групп.

Вторая классическая модель синтаксической структуры – дерево непосредственных составляющих. Основные идеи по этой модели принадлежат Блумфилду, которые он высказал в начале 30-х годов. Конструктивное развитие модель получила в работах Уэллса, Хэрриса, Хомского. В основе модели дерева составляющих лежит представление об устройстве предложения как о последовательном попарном синтагматическом сцеплении составляющих от минимальных отдельных слов до максимальной - предложения, составляющими которого в случае полного личного предложения являются группа

подлежащего и группа сказуемого.

Представление синтаксической структуры в терминах дерева составляющих хорошо согласуется с традиционным «разбором» предложения, при котором подлежащее, сказуемое и их элементы описываются категориальными характеристиками – именами частей речи или групп. Например, классическая фраза Блумфилда «*Бедный Джон убежал прочь»* будет представлена так, как показано на рисунке 17.

Отличительной особенностью модели дерева составляющих является то, что она задает порядок (степень близости между словами) во множестве слов, которые в предыдущей модели подчинялись бы одному и тому же узлу.

Дерево составляющих передает также соответствие между синтагматикой и линейной упорядоченностью слов в предложении. Нарушение прямого соответствия выражается в форме прерывных (или разрывных) составляющих, которые особенно распространены в языках со свободным порядком слов. Как и в графе зависимостей, в дереве составляющих могут использоваться условные узлы и связи.

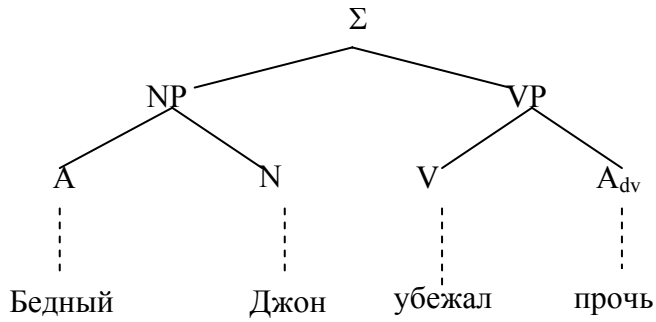


Рис. 17. Синтаксическая структура в терминах дерева

Здесь Σ - символ предложения, A – прилагательное, N – существительное, V – глагол, A_{dv} – наречие, NP – именная группа, VP - глагольная группа.

Следует подчеркнуть, что системы составляющих и деревья зависимостей характеризуют синтаксическую структуру предложения в разных аспектах. С помощью первых описываются в явном виде словосочетания, но игнорируется ориентация связей (т.е. не различаются “хозяин” и “слуга”); вторые дают возможность рассматривать направленные связи, но только между отдельными словами.

В настоящее время распространенным способом описания синтаксической структуры является комбинирование приемов двух классических моделей: обозначение порядка замыкания связей в дереве составляющих систем ЛГУ (2 версия), использование нетерминальных узлов в графах зависимостей системы ПОЭТ.

Выбор того или иного способа представления синтаксической структуры в значительной степени связан с устройством алгоритма синтаксического анализа. Для жестко заданных процедур, вычисляющих синтаксическую структуру предложения по «формуле» правильной структуры, в качестве такой формулы плохо подходит модель типа граф зависимостей: она либо не доопределяет процедуру построения синтаксической структуры, и

тогда появляется слишком много вариантов анализа, либо - если используются сильные ограничения - как формула становится слишком сложной для вычисления. Формальные грамматики работают, как правило, с синтаксическим представлением в виде дерева составляющих. Привлекательными свойствами графа зависимостей является их экономичность, удобство использования в преобразованиях, возможность представления частичных результатов анализа в виде множества подграфов. Модель данного типа используют системы групп Г.Г. Белоногова, АРТ, РЕЗОН, ЭТАП-2, ПОЭТ, АДАМАНТ, САГА, большинство японских систем анализа текста и ряд других.

С точки зрения описания естественного языка формальными теориями различают формально-грамматический и вероятностно-статистический подходы. Формально-грамматический подход направлен на создание сложных систем правил, которые позволяли бы в каждом конкретном случае принимать решение в пользу той или иной синтаксической структуры, а статистические – на сбор статистики встречаемости различных структур в похожем контексте, на основе которого и принимается решение о выборе варианта структуры.

Формально-грамматические подходы заложены классификацией формальных языков и грамматик, предложенной Хомским. Для компьютерной лингвистики среди них наиболее важны грамматики конечных автоматов, контекстно-свободные и контекстно-зависимые грамматики. Для описания естественно-языковых феноменов в основном применяются КС-грамматики с некоторыми расширениями.

Грамматика конечных автоматов (Finite-State Transition Network) формально соответствует простой по возможностям грамматике третьего типа. Конечный автомат содержит набор состояний (нетерминальных символов), среди которых выделяют одно или несколько начальных и конечных, и условий перехода между состояниями. Информацией для перехода по условиям служат символы, поступающие с ленты, которую читает автомат. Иногда конечный автомат может писать символы на другую ленту, в англоязычной традиции такой автомат называют *transducer*. Часто для лингвистических приложений условия перехода не задаются непосредственно, а вычисляются словарным компонентом, ставящим в соответствие символам или цепочкам символов ленты-символы их обобщенных классов.

Конечные автоматы являются декларативным средством представления, что означает возможность их обратимости, т.е. применения и для анализа, и для синтеза. Они также весьма эффективны с точки зрения скорости работы, но ограничены в возможности описания многих структур, встречающихся в естественном языке, таких как вложенные конструкции, например, из вложенных друг в друга придаточных предложений.

Более высокий уровень грамматик составляют контекстно-свободные грамматики, которые описываются в виде продукций (правил), ставящих в соответствие нетерминальным символам в своих левых частях (до знака «=») набор терминальных и нетерминальных символов в правых частях. Пример контекстно-свободных правил (КС-правил) для простой грамматики русского языка дан на рисунке 18. КС-правила в первой колонке описывают структуру нетерминальных символов, во второй – словарь, т.е. соответствие между

нетерминальными и терминальными символами.

S	= NPVP	ADJECTIVE	=	молодой
VP	= VERB	ADJECTIVE	=	старого
VP	= VERBNP	ADJECTIVE	=	лежащего
NP	= NOUN	NOUN	=	лис
NP	= ADJECTIVENP	NOUN	=	волка
PP	= PREPOSITION NP	VERB	=	видит
		VERB	=	лежит

Рис. 18. Пример КС-правил (S - предложение, NP - именная группа, VP - глагольная группа, PP - предложная группа)

Подобная грамматика описывает такие предложения, как "лис видит волка"; "молодой лис видит старого волка"; "молодой лис видит старого лежачего волка"; "лис лежит" и т.д. Достаточно просто расширить эту грамматику, чтобы представить в словаре русскую морфологию в более полном виде. Заметим, что в данной грамматике выбор конкретного правила для построения глагольных групп (VP-правила) или именных групп (NP-правила) задан вариантами, гарантированный выбор между которыми сделать в рамках данного правила невозможно. Подобная грамматика относится к так называемым недетерминированным грамматикам.

Синтаксис КС-правил очень прост, однако для описания многих феноменов естественного языка простого аппарата КС-грамматики оказывается недостаточно. В частности, контекстно-свободными правилами неудобно описывать согласование (например, в лице и числе между подлежащим и сказуемым). КС-аппарат неудобен также для отображения разорванных зависимостей (long-distance dependencies), вызванных передвижением слов по фразе, или для описания отсутствия составляющих (deletion).

В традиции трансформационных грамматик для представления подобных феноменов вводятся трансформации, переводящие синтаксическую структуру таких фраз в стандартную. Одним из способов отражения изменений синтаксической структуры без использования трансформаций является *Node raising*. В такой методологии то место, которое должно быть занято некоторой именной группой в стандартной синтаксической структуре дерева составляющих, обозначается пустым узлом и дополняется признаком *slash* (NP/). Такой узел располагается, как правило, справа от реальной позиции соответствующей составляющей и в более глубокой, составляющей дерева (например, Wh-группа зависит от корня дерева, а NP/ - от глагольной группы). В таком описании Wh-группа как бы поднимается относительно своей стандартной позиции (отсюда понятие raising).

В классических КС-грамматиках так же неестественно представляется такой феномен, как субкатегоризация, т.е. специфические свойства подкласса какой-либо категории. Например, КС-грамматика, изображенная на рисунке 18, не отличает переходные и непереходные глаголы, поэтому она принимает предложения, содержащие прямые дополнения у непереходных глаголов. Если же ввести два нетерминальных символа, TV и IV для переходных и непереходных глаголов соответственно, то в этом формализме невозможно будет отразить свойства, общие для обеих групп глаголов. Все эти проблемы

приводят к тому, что грамматические формализмы расширяют аппарат КС-правил с целью представления подобных феноменов.

Кроме отмеченных выше проблем КС-грамматики добавляют еще одну. В правиле, выражающем отношения между составляющими, не отражается естественная особенность естественных языков - поглощение одной категории другой, так что новая составляющая выступает заменителем управляющей категории. В частности, есть очевидное сходство в образовании именных групп из существительных и глагольных групп из глаголов. По этой причине синтаксис КС-правила, использующего составляющие, ограничивается описанием отношения категории X и категории, которой она управляет. Например, если существительное управляет определением, тогда именная группа может быть записана как $\overline{N}$ (иногда записывается как N').

С другой стороны, построение именной группы завершается указанием спецификатора (артикля или местоимения), для отражения такого факта вводится комбинация надчерков: $\overline{\overline{N}}$. Число надчерков при этом означает уровень проекции данной составляющей. Существенно ограничение максимального количества штрихов двумя: первый соответствует частично построенной группе, например, глагольной группе вместе со своими актантами, введение подлежащего (максимальная проекция второго уровня) превращает глагольную группу в законченную пропозицию. Таким образом, вся синтаксическая структура состоит из комбинации поддеревьев.

Многие теории (примерно с начала 80-х годов) перешли от описания грамматики в терминах правил к описанию ограничений (licensing rules), накладываемых на сформированность (well-formedness) частей выражения. При таком способе описания языка синтаксис языка не задается, различные ограничения в явном виде друг с другом не связаны. Анализ (или синтез) при этом является попыткой найти представление, одновременно удовлетворяющее всем ограничениям, причем возможные варианты конструкций строятся параллельно (или псевдо-параллельно). Представители этого направления связывают популярность таких грамматик с тем, что правила (КС или КЗ) описывают структурные свойства лингвистических конструкций, в то время как ограничения на сформированность являются более общими принципами, определяющими эти конструкции. В частности, это приводит к большей независимости правил от конкретных конструкций (нужно написать меньше правил для описания сравнимых элементов грамматики языка) и возможности описания в грамматике свойств лексических единиц.

Существует два способа применения синтаксических правил: снизу вверх и сверху вниз. В первом случае применяются правила, заменяющие структуру, описанную в правой части, символом, представленным в левой части. Во втором случае доказывается выводимость данного предложения из начального символа S. Часто оказывается возможным применить правила несколькими способами при анализе снизу вверх.

В синтаксическом анализе существуют две стандартные стратегии применения правил при возможности альтернативного выбора: поиск "в ширину" и поиск "в глубину". В первом случае запоминаются все возможные варианты, и каждый из них разворачивается

параллельно (или по очереди в случае последовательного анализа), при неудаче какого-либо варианта разбора соответствующий вариант удаляется из набора возможностей. Во втором случае, при анализе "в глубину", выбирается одна из альтернатив, а при неудаче построения разбора происходит возврат на точку последней альтернативы и выбор другого варианта. Использование анализа с проходом сверху-вниз не позволяет создавать неграмматичные варианты. С другой стороны, анализ снизу-вверх не позволяет генерировать гипотезы разбора, невозможные для данного предложения.

Комбинацию достоинств этих вариантов представляет анализ с помощью таблиц, содержимое которых является результатом частичного разбора. В случае, если разбор по какому-то пути зашел в тупик, происходит возврат на точку выбора последнего правила и делается попытка использовать другое правило. Однако заполнение таблицы, порожденное предыдущим способом разбора, сохраняется в таблице и может быть использовано в разборе по текущей ветке. Эта информация не запрещает проход анализа по тем веткам, которые уже были опробованы, но неудачно. Для этой цели применяется запоминание также и гипотез, выдвигаемых при разборе, и результатов их проверки. Такой подход называется анализом с помощью схем (chart-parsing). Впервые его предложил Мартин Кэй в системе Powerful Parser.

Грамматики конечных автоматов достаточно эффективны в реализации, но обладают слишком ограниченными возможностями для анализа, по этой причине одним из широко используемых механизмов анализа является формализм расширенных сетей переходов (augmented transition networks, ATN). Формализм ATN расширяет грамматику конечных автоматов, вводя аппарат рекурсивного вызова новой подсети переходов (операция PUSH) и набор регистров, в которых хранятся текущие результаты разбора фразы, а также средства работы с этими регистрами. Значения регистров могут выступать условиями для переходов по веткам, что обеспечивает частичную зависимость от контекста и выход за пределы КС-грамматик. Благодаря регистрам и операциям над значениями, которые там хранятся, ATN-формализм эквивалентен процедурному языку программирования, в котором можно описать анализ языка произвольной сложности.

Логико-алгоритмические подходы синтаксического анализа. В настоящее время можно говорить о существовании трех основных логико-алгоритмических подходов СА: недетерминированный «сначала в ширину», недетерминированный «сначала в глубину» и детерминированный.

Недетерминированный «сначала в ширину». Недетерминированный подход «сначала в ширину» характеризуется тем, что процедура синтаксического анализа на первом этапе порождает заведомо избыточный набор синтаксических связей, из числа которых на втором этапе с помощью серии фильтров отбираются такие, которые в совокупности давали бы правильную синтаксическую структуру входного предложения (или несколько правильных синтаксических структур). Этот подход был впервые теоретически обоснован и экспериментально проверен О.С.Кулагиной. В настоящее время эта стратегия имеет варианты, которые различаются:

1) степенью ослабления контекстуальных условий на этапе порождения связей;

2) статусом синтаксических структур, подвергающихся фильтрации (синтаксическая структура входного предложения, синтаксическая структура фрагмента входного предложения) и другими чертами.

Примерами систем, использующих этот подход, могут служить АРТ, ЭТАП-2, алгоритм Л.Г. Митюшина,1 и 2 версии ЛГУ, алгоритм Е.И. Анно и т.д.

Недетерминированный «сначала в глубину». Во многих системах, ориентированных на промышленную эксплуатацию, используется другая стратегия СА. Она встречается под разными названиями: стратегия, опирающаяся на механизм возвратов backtracking, стратегия depth-first («сначала в глубину»); в некоторых работах эта стратегия объединяется с предыдущей под одним названием «недетерминированный анализ». Этот подход принят, например, в системе АДАМАНТ. Отличие его от концепции псевдопараллелизма состоит в том, что алгоритм на каждом шаге выбирает одну из возможных интерпретаций, но при этом сохраняется принципиальная возможность порождения альтернативных интерпретаций в случае той или иной неудачи с первой (например, если полученная синтаксическая структура входного предложения или его фрагмент не удовлетворяет требованиям проективности, связности, не проходит семантический фильтр и т.п.). Если первый вариант разбора признается неудовлетворительным, нет необходимости начинать анализ сначала. Процедуре анализа достаточно вернуться в ближайшее из состояний, при котором возможен был альтернативный путь, и попытаться довести до конца этот вариант. Если же и он окажется неприемлемым, процедура снова использует механизм возвратов и перейдет к следующему варианту и так далее, пока не будет порожден первый приемлемый вариант разбора входного предложения. Поиск других вариантов после этого прекращается.

Скорость работы системы с механизмом возвратов зависит от того, удается ли ей в подавляющем большинстве случаев получать приемлемый вариант синтаксического анализа с минимальным количеством возвратов – в идеале без них. Если алгоритм не позволяет этого, он не является оптимальным с точки зрения быстродействия, так как прежде, чем приемлемый вариант будет найден, алгоритм затратит время на порождение и фильтрацию неверных вариантов анализа входного предложения или его фрагментов. Такими являются, к примеру, алгоритмы в системах LUNAR Вудса и SHRDLU Т. Винограда.

Это общий недостаток двух рассмотренных стратегий. Однако скорость анализатора, опирающегося на механизм возвратов, по-видимому, выше. Алгоритм, опирающийся на механизм возвратов, может располагать эффективным способом обработки простых и стандартных по структуре предложений, практически не порождая избыточных синтаксических структур. В то же время использование механизма возвратов позволит ему найти приемлемую интерпретацию для менее стандартного по структуре предложения.

Чтобы избежать указанного недостатка, в ряде систем (например, АДАМАНТ) упор делается на развитые эвристические методы, управляющие процессом анализа, которые позволили бы получать предпочтительный вариант разбора первым.

Детерминированный подход. Третья стратегия – стратегия детерминированного анализа – базируется на следующем принципе: ни одна синтаксическая связь, установленная

в процессе анализа предложения, не может быть отвергнута, иными словами, если связь порождена, она должна присутствовать в синтаксической структуре, являющейся результатом работы синтаксического анализатора.

Эта стратегия используется в системах ЯИП, САГА и другие. Синтаксический анализатор, разработанный в группе Г.Г. Белоногова, в целом также может быть отнесен к этому типу, хотя локально он допускает пересмотр уже установленных связей.

Стратегии, о которых шла речь выше, на этапе порождения связей используют лишь часть информации, к которой имеет доступ синтаксический анализатор. Неполнота касается, прежде всего, сведений о контексте, которые учитываются в полной мере после того, как связи порождены: при фильтрации связей или оценке приемлемости построенной синтаксической структуры. Стратегия детерминированного анализа не использует подобного деления на этапы: вся информация, которая в построенном синтаксическом анализаторе может повлиять на установление связи между конкретными текстовыми единицами, привлекается одномоментно. Укажем еще одну отличительную характеристику стратегии детерминизма: при установлении каждой связи должны соблюдаться такие условия, которые гарантировали бы получение связной синтаксической структуры предложения на выходе.

Для окончательного вывода о наличии связи определенного вида между двумя текстовыми единицами (ТЕ) необходимо проверить, помимо условий на сочетаемость, соблюдение некоторого количества контекстуальных условий (наличие или отсутствие в фиксированной позиции других ТЕ с заданными характеристиками, наличие или отсутствие в фиксированной позиции тех или иных знаков препинания и т.п.). Такие условия могут быть сформулированы не для конкретной пары ТЕ, а для большого класса таких пар. В этом случае очевидно, что набор таких условий, заданный в обобщенном виде, описывает синтаксическую ситуацию, диагностичную для расстановки связей. В основе стратегии детерминированного анализа лежит инвентарь синтаксических ситуаций, которые учитываются данной моделью синтаксического анализа. Описание ситуации может быть задано в декларативном или процедурном виде – это зависит от языка программирования.

Синтаксические ситуации привязаны к тому или иному грамматическому явлению: поиск и установление связей однородных членов, поиск подлежащего, выявление определительного номинатива и поиск его хозяина и прочее. Каждому грамматическому явлению сопоставлен набор синтаксических ситуаций. Алгоритм проверяет, какая из предусмотренных ситуаций реализована в анализируемом предложении, и в соответствии с этим устанавливает синтаксические связи. Так как стратегия в принципе ориентирована на построение одного варианта грамматического разбора, описание синтаксической ситуации задано с той степенью подробности, которая позволяет разработчикам принимать решение об однозначной расстановке связей.

Однако не исключены ситуации, в которых синтаксический анализатор не имеет достаточной информации для однозначного выбора, а статистические наблюдения не позволяют уверенно предпочесть одно решение другому. Система ЯИП и система группы Г.Г. Белоногова в таких ситуациях все равно делает однозначный выбор на основе

достаточно грубых вероятностных соображений. Система САГА допускает в строго ограниченном количестве ситуаций подобного рода построение альтернативных вариантов, которые останутся в результирующем представлении входного предложения.

Качество анализа в системах, основанных на концепции детерминизма, может быть разным. Чем более дифференцированным описанием ситуаций оперирует алгоритм, тем точнее он работает: возможности совершенствования качества анализа здесь достаточно богатые. Но с повышением точности скорость анализа уменьшается. Чем более грубо заданы синтаксические ситуации, тем быстрее работает алгоритм, но тем больше вероятность ошибки.

По сравнению с другими стратегиями стратегия детерминированного анализа оказывается более экономной в том смысле, что она не затрачивает время на порождение и фильтрацию избыточных связей.

Лекция 9. Алгоритмы и база знаний синтаксического анализа

В результате синтаксического анализа должны быть однозначно определены все синтаксические единицы естественно-языкового предложения. *Синтаксическими единицами* будем называть конструкции, в которых их элементы (компоненты) объединены синтаксическими связями и отношениями. Синтаксическая связь является выражением взаимосвязи элементов в синтаксической единице, то есть служит для выражения синтаксических отношений между словами, создает синтаксическую структуру предложения и словосочетания, а также условия для реализации лексического значения слова. Исходными данными для проведения синтаксического анализа являются результаты морфологического анализа, представленные в виде множества пар $<x_i, V_i>$, где x_i – ЕЯ-лексема, V_i – вектор морфологической информации x_i лексемы.

Синтаксический анализ проходит три этапа. На первом этапе осуществляется нормализация лексем естественно-языкового предложения для выделения синтаксических групп, к которым относятся группы ФИО, ДАТА, ПС (существительное с предлогом) и другие, описание и правила выделения которых более подробно рассмотрены в следующем разделе данной главы. На этом же этапе осуществляется удаление несущественных лексем из исходного множества, таких как служебные части речи (предлоги, союзы, частицы и т.п.). В результате будут сформированы два множества: новое исходное множество лексем X и L - множество синтаксических групп в виде векторов связанных лексем.

Синтаксическая связь, относящаяся к типу подчинение, передает сочетание двух слов, в котором одно выступает как главное, а другое – как зависимое. Поэтому задачей второго этапа является выявление синтаксической связи между двумя лексемами множества X и множества векторов L, разбиение лексем на множество главных слов L_1 и множество зависимых слов L_2, причем $L_1 \cap L_2 \neq \varnothing$, и формирование множества сочетаемых пар лексем $D = \{(x_i, x_j) | \, x_i \in L_1, x_j \in L_2\}$. Для нахождения корневой вершины необходимо:

1) объединить множества L_1 и L_2: $L_3 = L_1 \cup L_2$;

2) найти разности множеств L_3 и L_2: $L_4 = L_3 \setminus L_2$, где L_4 – одноэлементное множество

корневых вершин.

Таким образом, формируется один или несколько графов зависимостей $G=<X, D>$, где X – множество вершин графа G, которое составляет множество лексем $X=\{x_i| \ i=1,n\}$, а D – множество дуг.

Итоговый граф зависимостей G будет удовлетворять следующим требованиям:

- граф G является неполным графом, т.е. не содержит петель и циклов;
- граф G является связным.

База знаний синтаксического анализатора. Синтаксический анализ осуществляется на основе использования следующих видов информации:

- знания о морфологических характеристиках словоформ;
- знания о синтаксических отношениях (отношения зависимости) словоформ;
- знания о порядке слов в предложении;
- знания о пунктуации.

Знания о морфологических характеристиках словоформ представлены в виде результатов морфологического анализа, которые подаются на вход синтаксического анализа [6, 9].

Знания о синтаксических отношениях (отношения зависимости) словоформ определяются на основе правил соответствия их морфологических характеристик. Правила соответствия описываются в виде условий применимости и основываются на теории синтаксиса русского языка. В зависимости от принадлежности главного слова к той или иной части речи различаются лексико-грамматические типы словосочетаний: глагольные, именные, наречные. Глагольные словосочетания имеют следующие модели:

1) глагол + существительное или местоимение с предлогом или без предлога;

2) глагол + инфинитив или деепричастие;

3) глагол + наречие.

Именные словосочетания делятся на субстантивные, адъективные, с главным словом числительным и с главным словом местоимением.

Основными моделями субстантивных словосочетаний являются:

1) согласуемое слово + существительное;

2) существительное + существительное;

3) существительное + наречие;

4) существительное + инфинитив.

К основным моделям адъективных словосочетаний относят:

1) прилагательное + наречие;

2) прилагательное + существительное (местоимение);

3) прилагательное + инфинитив.

Последние типы словосочетаний с главным словом числительным и с главным словом местоимением являются синтаксически не свободными и разнообразием моделей не отличаются (например, двое друзей, два товарища, некто в белом, что-нибудь особенное).

Словосочетания наречного типа (с предикативными и непредикативными наречиями)

имеют 2 модели:

1) наречие + наречие;

2) наречие + существительное.

Связь между частями речи, представленных в данных моделях словосочетаний, определяется на основе морфологической информации лексем. Эти модели и составляют базу правил синтаксического подчинения.

Знания о порядке слов в предложении также влияют на результат анализа. Порядок слов в русском языке, вопреки устойчивому заблуждению, не вполне свободный, гибкий [7, 8, 9]. В каждом отдельном случае порядок слов зависит как от грамматики предложения, так и от смысла высказывания. Самое существенное - то, ради чего и создается предложение, должно располагаться в конце его. В случае запросов к базе данных и согласно разработанным ограничениям, наоборот, самое существенное должно располагаться в начале предложения и будет являть собой объект запроса, а все остальное – относиться к условию запроса. В данной работе будут учитываться правила порядка предлогов и существительных, союзов и теория примыкания падежей с предлогами, что должно способствовать правильному распознаванию моделей сочетания существительных с другими сопряженными частями речи.

Существительное принимает в качестве определителя разнообразные примыкающие падежные формы с предлогами. Эти формы или, подобно согласуемым формам, определяют имя, или, очень часто, определяя имя, одновременно тяготеют к глаголу, который управляет этим именем. На основе переразложения глагольных связей в современном языке активно пополняется состав примыкающих к существительному падежных форм с предлогами (например, написать письмо в деревню - письмо в деревню, провести вечер у костра - вечер у костра и т.д.). В таких предложениях для более корректного анализа необходимо знать, какую падежную форму и какое обстоятельственное значение несет существительное с примыкающим предлогом.

Всего в русском языке насчитывается порядка 300 предлогов. При этом один и тот же предлог может использоваться в разных падежных формах существительного, т.е. может участвовать в формировании различных обстоятельственных значений существительного.

Формы, примыкающие к существительному и определяющие его, могут нести в себе более или менее ярко выраженные обстоятельственные значения: места, времени, количества или меры, причины, назначения, источника или происхождения, условия, состояния, совместности или несовместности, возместительности, сферы действия. Тогда по формам предлогов, примыкающим к существительному и определяющим его, можно будет определить обстоятельственные значения существительного, что в дальнейшем понадобится при семантическом анализе естественно-языкового запроса.

Знания о пунктуации необходимы для определения однородных членов предложения, определения причастных и деепричастных оборотов. Вся языковая информация представляется в виде формального описания, согласованного с выбранным методом и используемого в качестве данных для переработки входного предложения.

Основываясь на приведенных в данном разделе видах языковой информации, выделяются три группы правил, связанных с анализом различных ситуаций: 1 группа правил позволяет выявить синтаксические группы; 2 группа - синтаксическую связь между парой лексем; 3 группа - синтаксическую связь между парой лексем и синтаксической группой.

РАЗДЕЛ 5. СЕМАНТИЧЕСКИЙ АНАЛИЗ ЕЯ-ТЕКСТОВ

Лекция 10. Анализ лингвистических моделей

На данный момент разработано множество моделей лингвистического анализатора, которые способны в той или иной степени выполнять анализ естественно-языкового текста, определять смысл и генерировать высказывания. При этом подходы к моделированию процесса общения весьма разнообразны. Основные отличия этих подходов заключаются в методах реализации компонента понимания смысла, используемых средствах анализа, а также в объеме и способах представления знаний, поскольку именно знания, представленные в различной форме, являются базой, от которой зависит процесс общения, глубина проникновения в смысл и, соответственно, качество самой модели лингвистического анализатора. От выполнения отдельных функциональных компонент зависит практическая реализация моделей в различных системах общения (системы общения с базами данных, системы машинного перевода и др.). Некоторые из них легли в основу конкретных систем формирования семантического представления на основе обработки текстов (например, модель Смысл-текст в системе «Поэт»).

Проанализируем наиболее проработанные модели лингвистического процессора с точки зрения реализации анализа и интерпретации входного высказывания и синтеза выходного высказывания.

В задачу анализа входит выделение смысла входного текста (под смыслом будем понимать семантику – информацию, которую пользователь хотел передать системе) и выражения этого смысла на внутреннем языке системы. Интерпретация заключается в отображении входного текста на знания системы. Одним из основных параметров анализа текста является понимание смысла входного предложения, включающее в себя описание сущностей входного текста, определение их свойств и отношений между ними. От этого параметра часто зависит глубина проникновения в смысл входного текста.

В существующих моделях лингвистического анализатора можно выделить следующие способы выделения и представления смысла: компонентный анализ; сеть концептуализаций; идентификация смысла по образцу; интегральный подход.

Одна из первых попыток формализации входного текста принадлежит компонентному анализу, который исходит из предпосылки, что семантика естественных языков может быть выражена в терминах конечного неструктурированного набора семантических множителей (атомов смысла). В процессе рассмотрения слов выделяются признаки (одушевленность, неодушевленность и т.п.), которые разбивают слова на отдельные группы. При кажущейся естественности данный метод связан с существенными трудностями при реализации и не лишен слабостей. Он становится сложным при выражении смысла целого предложения и громоздким при анализе многозначных слов, при этом нет достаточного объяснения слова, что может привести к неправильному его употреблению.

В дальнейшем идея описания входного текста с помощью компонентного анализа нашла свое продолжение в модели «Семантические падежи (роли)» Ч. Филмора. Но в отличие от предыдущей модели в предикатах указывается не только аргументная структура и

количество, но и их семантическое содержание (роли). Филмор выделяет следующие семантические роли: агент, контрагент, объект, адресат, пациенс, результат, инструмент, источник. В модели предложена более детальная концепция смысла высказывания. Каждое понятие расщепляется на две сущности: значение и пресуппозицию. Различия между пресуппозицией и значением в собственном смысле слова проявляются, например, в различном влиянии на них отрицания. В область действия отрицания попадает только значение, а не пресуппозиция. В результате исследований была разработана классификация семантических элементов, что привело к пересмотру обычной схемы словарной статьи в толковом словаре (словарь стал основным средством задания семантических структур и правил их перевода в поверхностные структуры).

Продолжением данной теории явился метод падежной грамматики (Филмор). При этом для записи содержания входного высказывания используются специальный синтаксический язык, словари и правила, устанавливающие соответствие между естественно-языковыми выражениями и их семантическим представлением.

Ко второму классу относятся модели, в которых смысл текста представляется в виде сети концептуализаций. В таких моделях явления рассматриваются только на одном уровне детальности, что не позволяет как описывать сложные события в терминах более простых подсобытий, так и дробить при необходимости примитивные действия (атомы). Чаще всего эти модели являются моделью языка, а не моделью общения, что приводит к нечеткому выделению языковых средств и средств для описания моделируемого окружения. Среди моделей данного класса наибольший интерес представляет модель «Концептуальной зависимости».

Основой семантического представления модели «Концептуальной зависимости» (Р. Шенк) является сеть концептуализаций. Сеть концептуализаций есть квазиграф, подобный размеченному ориентированному графу, в котором, кроме бинарных отношений, есть тернарные и кварнарные, а дуги связывают не только вершины, но и другие дуги.

Концептуализация в модели концептуальной зависимости определяется как основная единица семантического уровня, из таких единиц конструируются мысли. Концептуализация включает в себя действие, множество его концептуальных падежей и участников действия (их состояний).

Будучи моделью языка, она не учитывает модели пользователя, что приводит к полному перебору при построении умозаключений. Наличие модели пользователя позволило бы определить его цели (намерения) в диалоге и использовать их для направления процедуры построения умозаключений.

Другая модель - «Семантик предпочтения» относится к классу моделей, идентификация смысла в которых осуществляется по образцам. Отличительной чертой таких моделей является то, что в них отсутствуют блоки морфологического и синтаксического анализов, что является принципиальным их недостатком, так как не обеспечивается глубина анализа значений слов, необходимая для точного установления семантической связности текста.

В этой модели (Уилкс) текст характеризуется следующими сущностями: смыслами

слов, сообщениями, фрагментами текста и семантической совместимостью. Сообщение рассматривается как теоретический конструкт, посредством которого для каждого слова, входящего во фрагмент текста, может быть выбран один из смыслов слова, посредством чего снимается многозначность. Слову назначается тот из его многих смыслов, который образует «сообщение», согласующееся, в конце концов, с рассматриваемым фрагментом текста. Если слово может подойти к нескольким сообщениям, то выбирается такое, которое согласуется с рассматриваемым текстом.

Анализ фрагмента текста протекает по следующей схеме. С помощью специальных слов-маркеров выполняется фрагментация текста, затем словам приписывают из словаря все их значения. Далее на анализируемый фрагмент текста поочередно накладываются простые шаблоны, известные системе. С помощью специальных правил расширения простой образец преобразуется в полный образец путем добавления слов из текста, которые не вошли в образец. Указанная процедура осложнена тем, что может подойти не один простой образец. Используя процедуры установления семантической близости полученных образцов, формируется окончательное представление обрабатываемого текста. К недостаткам анализа следует отнести то, что анализ текста осуществляется с помощью словаря шаблонов, которые способны различать только класс событий, а не сами конкретные события.

Другой подход к способу анализа по образцу представлен в моделях, использующих табличный метод. Он основан на анализе ключевых слов, встречающихся в предложениях. Суть табличного метода состоит в идентификации смысла всего предложения на основании нескольких ключевых слов или их групп. После процесса идентификации слова предложения заменяются на их каноническую форму - коды. Замена осуществляется с помощью словаря словоформ. При этом также выделяются некоторые группы слов, несущие тематическую нагрузку. Далее производится распознавание и замена стандартных словосочетаний. Данный метод обладает рядом недостатков, преимуществом является его простота для однозначных естественно-языковых предложений, в которых не требуется полного понимания смысла предложения (например, запросы к базе данных).

Модели, в которых достаточно глубоко продуманы процедуры морфологического, синтаксического и проблемного анализов, можно отнести к моделям, основанным на интегральном подходе описания языка. Это модель «Смысл-текст» и модель контекстного фрагментирования.

Модель «Смысл-текст» (И.А. Мельчук) представляет собой многоуровневый транслятор текстов в смыслы и наоборот. Выделяются четыре основных уровня – фонетический, морфологический, синтаксический и проблемный. Каждый из них, за исключением проблемного, подразделяется на два других уровня – поверхностный и глубинный.

Данная модель может быть применима в системах, где необходимо понимание текста в полном смысле (например, вопросно-ответные системы, системы принятия решений). Но для реализации полной схемы анализа и синтеза модели «Смысл-текст» придется учесть индивидуальные свойства сотен тысяч словарных, морфологических и лексических единиц и

индивидуальные свойства громадного числа пар единиц. Их полное формальное описание представляет собой громадную и объемную теоретическую работу, поставленную в лингвистике в последнее время и еще далекую от решения.

Модель контекстного фрагментирования разрабатывалась для анализа и синтеза естественно-языкового предложения, но ее проработка касается в основном анализа. Задача лингвистической трансляции естественно-языкового текста рассматривается отдельно от других задач общения на естественном языке и от задач самой вычислительной системы. Анализ и трансляция текста осуществляются при наличии достаточно мощных средств описания и фрагментации лингвистических знаний. Основу модели контекстного фрагментирования составляет трехуровневая система: лингвистическая модель, базовые механизмы обработки предложений и ассоциированные процедуры. Лингвистическая модель содержит информацию о морфологии, синтаксисе и семантике подмножества естественного языка. В модели выполняется очень глубокий синтаксический анализ с одновременным преобразованием распознаваемых синтаксических отношений в семантические. Достоинством данного метода является то, что существует возможность динамически изменять стратегию обработки естественно-языкового текста в зависимости от необходимой глубины и последовательности этапов трансляции и расширять метод при включении новых конструкций естественного языка и редуцировать его для упрощенных подмножеств естественного языка и проблемных областей.

В заключение обзора различных подходов и направлений реализации моделей лингвистического процессора можно сделать вывод о том, что к настоящему времени модели способны: извлекать знания из заданного текста и строить правильные предложения естественного языка по заданным значениям смысла; перефразировать эти предложения; оценивать их с точки зрения связности и выполнять ряд других задач.

Лекция 11. Анализ средств формального описания понятий

Локальная модель мира представляет собой формализованное описание некоторого контекста, отражающего объекты и их отношения. Разделение лексем или групп лексем на объекты и отношения является достаточно условной процедурой и зависит от семантических ролей, исполняемых лексемами или группами лексем, отражающими некие значения в определенном контексте.

Как известно, в лингвистике разделяются такие понятия, как значение высказывания (или сущности) и его пресуппозиция. Пресуппозиция определяется как предшествующий контекст, предшествующее знание, или как контекст, в котором происходит определение значения сущности. Очевидно, что понимание сущности, прежде всего, обеспечивается именно пресуппозицией.

Известно, что первая попытка категоризации, т.е. выделения ролей элементов реального мира была осуществлена еще Аристотелем, который определил 10 категорий: сущность, действие, количество, качество, место, время, положение, претерпеваемость, обладаемость, соотнесенность. Понятия структурированы по двум типам отношений: «род-вид», «часть-целое». Первый тип структур определяет факт понятия в родовидовом дереве

(РВД) понятий, второй - уровень понятия в иерархии понятий. В модели мира один и тот же элемент, в зависимости от ситуации, может выступать в различных ролях и, наоборот, разные элементы могут выступать в одной и той же роли. Идея семантических ролей или семантических падежей достаточно активно исследовалась Ч. Филлмором, П. Уинстоном, Д. Апресяном и другими известными учеными в этой области. Проявление объектов и отношений в тексте можно рассматривать в трех аспектах:

1 - синтаксическом (КАК формируется?);

2 - семантическом (ЧТО означает?);

3 - прагматическом (ДЛЯ чего? В каких целях?).

Классификация элементов и назначение определенных ролей элементам или классам элементов и у Ч. Филлмора, и у Д. Апресяна происходит не на семантической, а на прагматической, целевой основе, т.е. по их назначению, а не по семантическому содержательному признаку [16]. Поэтому, вместо принятой в литературе понятия "семантическая роль", здесь предлагается новое понятие - "прагматическая роль" элемента, означающее целевую функцию объекта в заданном контексте. В данной лекции приведен анализ тех объектно-предикатных систем, которые, практически, покрывают все пространство объектов и их отношений, описанных во всех упомянутых работах, тем не менее, оставаясь лишь одним из вариантов выделения и описания прагматических ролей, не претендующим на завершенность и даже на достаточность. Ч. Филлмор в своих работах выделил 9 ролей элементов (рис. 19).

Данная система ролей, неоднократно модифицированная, стала основой для многих ролевых моделей и дала импульс для развития соответствующей теории. Объекты (или понятия), как правило, передаются на поверхностном уровне (в предложении, в дискурсе) в виде группы существительного. Поэтому вместо термина "роль" стало общепринятым использовать название соответствующей категории имени существительного - "падеж". Число падежей, используемых разными авторами, естественно, различается, так как это связано с моделью мира, которую они описывают, используя падежи. Описание модели мира вплотную связано с менталитетом разработчика, с его представлениями об объектах и их ролях, с его компетентностью, а также может раскрываться и дополняться возможностями, которые также априори, независимо от разработчика, заложены в естественном языке, единицами которого на поверхностном уровне кодируются элементы модели мира.

П. Уинстон в своих работах приводит 12 возможных падежей (см. рис. 19). Семантические падежи (роли) выделены и в модели концептуальной зависимости (КЗ-модель), разработанной группой Р. Шенка [16]. В КЗ-модели выделяется 9 семантических ролей (см. рис. 19).

Концептуализация Шенка включает в себя действие, множество его концептуальных актантов и участников действия (то есть ролей). В отличие от предыдущих систем, полная КЗ-модель является достаточно сложной конструкцией, так как преследует цель не только получить концептуальное представление текста на ЕЯ, но и "понимание" текста на его основе. Система ролей в КЗ-модели, тем не менее, весьма ограничена и может быть

применена лишь для очень ограниченного контекста. Очевидно, расширение системы ролей в целях универсализации КЗ-модели приведет к нелинейному увеличению ее сложности и, соответственно, к неэффективности системы. Вместе с тем, КЗ-модели можно отнести к одним из первых прагматически-ориентированных лингвистических моделей, использующих анализ, основанный на ожиданиях. Анализ при этом строится на том предположении, что наличие некоторого семантического представления текста, который уже начал анализироваться, задает набор возможных типов ожидаемых концептуальных структур. В рамках этого подхода используется прагматически-ориентированная технология, т.е. весь комплекс средств лингвистического и нелингвистического обеспечения рассматривается как единое знание, привлекаемое для обработки текста, определяющее «ожидание», следующее за текущей точкой разбора. Рассмотренные системы Ч. Филлмора, П. Уинстона и Р. Шенка отличаются друг от друга полнотой, терминологией, однако они схожи тем, что предназначены для описания пресуппозиции, т.е. более широкого контекста, нежели непосредственно объектно-предикатной ситуации, представленной в лексиколизованном (текстовом) виде. В отличие от описанных выше ролевых систем, Р.Г. Бухараевым и Д.Ш. Сулеймановым [16] разработана система ролей в вопросно-ответной ситуации, продиктованная необходимостью построения языко-зависимого лингвистического процессора. В основе системы Бухараева-Сулейманова лежит следующий Постулат 1.

Постулат 1. Множество ожидаемых значений вопроса определяет систему отношений и ролей, а также смысловых конструкций, формируемых в виде последовательности словоформ, т.е. текстов на естественном языке.

Бухараевым-Сулеймановым определено следующее множество ролей, необходимое для разработки формальной базы семантической интерпретации ответных текстов. Множество прагматических ролей (концептул), отражающих различные типы понятий:

Ks = {*SS*(*i*), *SO*, *Sc*, *SA*, *SP*}.

Здесь *SS(i)* - роль, отражающая *i*-ое главное понятие в тексте. *Главное понятие* - это понятие (понятия), относительно которого (которых) задан вопрос.

SO - роль, отражающая понятие, состоящее в некоторой определенной связи с *SS*(*i*).

Sc - роль, отражающая обобщенное понятие. *Обобщенное понятие* - это понятие, находящееся по отношению к *SS*(*i*) на более высоком уровне в иерархии понятий проблемной области (например, понятие "человек" по отношению к понятию "студент").

SA - роль, отражающая понятие-аргумент.

SP - роль, отражающая понятие-результат.

Множество прагматических ролей, отражающих различные типы отношений обозначены:

Kr = {Rso, Ros, Ra, Rr}, где

Rso, Ros = {Rc, Rсост, Rвкл, Rд, Rвро, Rпро, Rкло, Rкчо}.

Здесь *Rso* - роль, отражающая отношение *SS*(*i*) к другому понятию (понятиям); *Ros* - роль, отражающая отношение другого понятия (понятий) к *SS*(*i*); *Ra*, *Rr* - роли, отражающие отношение *SS*(*i*) к *SA* и *SR*, соответственно; *Rc* - роль, отражающая отношение

СОСТОЯНИЕ.

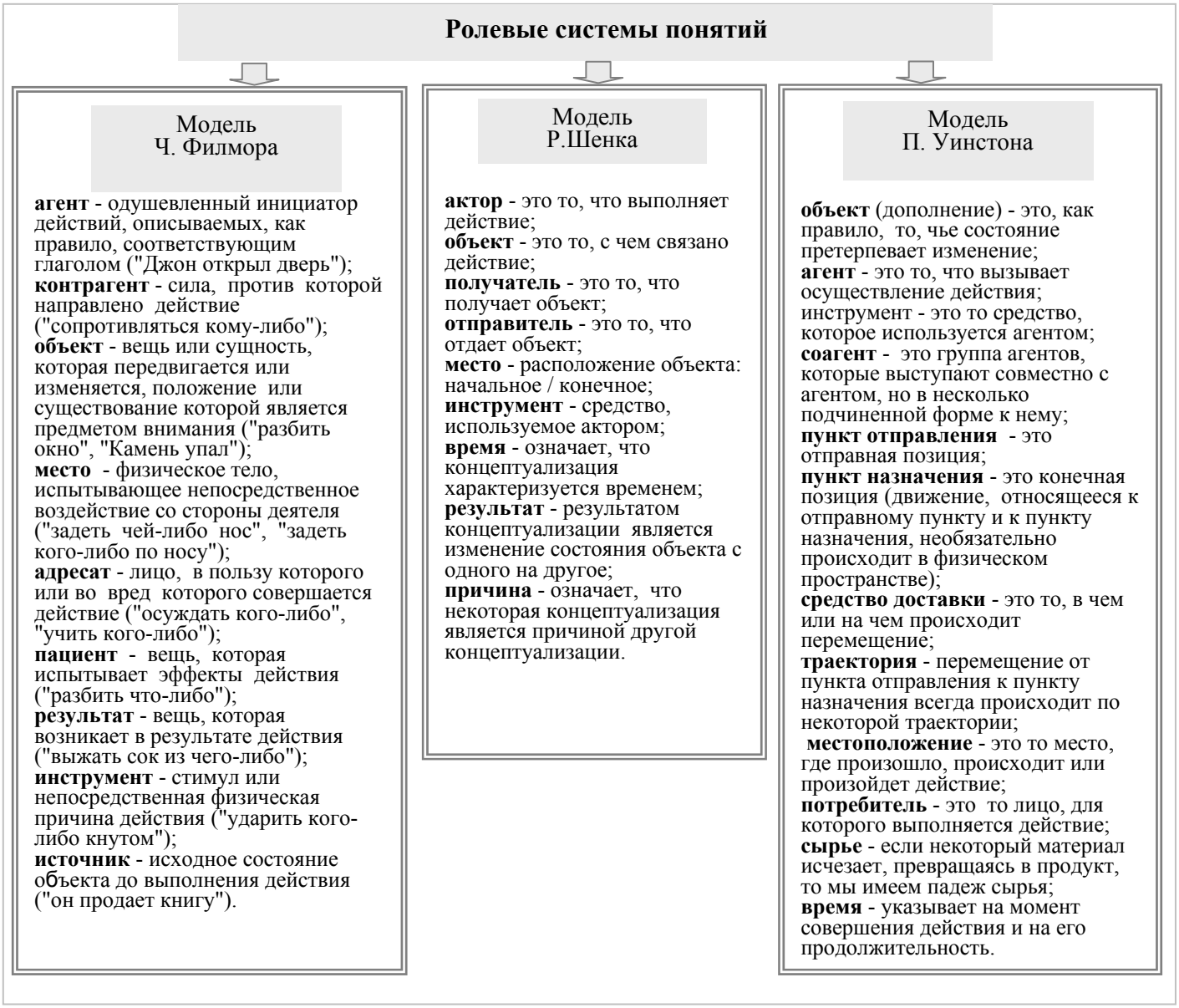


Рис. 19. Ролевые системы понятий ПО

Rсост, *Rвкл*, *Rд*, *Rвро*, *Rпро*, *Rкло*, *Rкчо* - отражают отношения Состав, Включение, Действие, Временное отношение, Пространственное отношение, Количественное отношение, Качественное отношение - соответственно. Кроме того, выделяются роли, отражающие грамматические признаки лексем, необходимые для сокращения пространства ожидаемых ответов (падежные окончания, предлоги и др.) и роли, отражающие специальные признаки (возможно, некие ограничения, четкие и нечеткие).

Ролевые системы Уилкса, Апресяна, Филлмора и др., главным образом, строятся как система ролей понятий. Причем, в этих системах не учитываются в необходимой мере грамматические (синтаксические и морфологические) признаки, в работе же Шенка, как уже упоминалось, они полностью игнорируются. Такой подход отражения контекста, очевидно, оправдан и удобен при описании значений корневых морфем, более того, имен существительных, или лексем, выступающих в роли имени или именной группы. Однако для

описания значений аффиксальных морфем наиболее удачным оказалось описание их через классы отношений. Рассмотрим крупноблочное описание объектно-предикатной системы М.З. Закиева.

Предикаты - отношения, связи (действия или состояния):

- предикат действия (я играю в футбол);
- предикат движения (мяч залетел в ворота);
- предикат чувственного восприятия (я обрадовался солнцу);
- предикат речи (ты расскажи стих);
- предикат состояния (ребенок спит);
- предикат долженствования (тебе надо платить);
- предикат предположения (похоже, он ушел);
- предикат позволения (тебе можно смотреть);
- предикат квалификации (моя сестра – певица);
- предикат-материал (у них мост из камня);
- предикат детерминации (парень очень умелый);
- предикат принадлежности (эта книга твоя);
- предикат обладания\отсутствия (он остался без коня);
- предикат наличия\отсутствия (на лице видна улыбка);
- предикат предназначения (эта книга дана тебе);
- предикат цели (хочу, чтобы ты учился);
- предикат времени (с театра вернулись рано);
- предикат места (наша деревня у реки);
- предикат сравнения (ты похожа на свою маму);
- предикат порядковый (он пришел первым);
- предикат количества (многие луга исчезли).

Субъект - предмет суждения, это то, о чем говорится, о чем сообщается (утверждается, что это спорная категория). *Объект* - это то, на что направлено действие или состояние:

- объект воздействия (дождь испортил настроение);
- объект активного воздействия, или контрагент (мы спрятались от грозы);
- объект совместного действия, или коагент (дрова пилили вместе с отцом);
- объект пассивного действия (сруб поднят);
- объект содержания речи (диктор передает сообщения);
- объект-место (живет в деревне);
- объект неожиданности (разделся до рубашки’);
- объект попутный (вместе с лошадью убежали и овцы);
- объект, не ставший им (взял вместо кнута);
- объект опережающий (после дождя выглянуло солнце);
- объект в качестве исполнителя (он выступал в качестве мельницы);
- объект-исполнитель (заставил делать брата домашние дела).

В работе Г.С. Осипова [17] выделено и описано 17 видов семантических связей:

1. Генеративная связь, один компонент которой обозначает лицо или предмет, принадлежащий некоторой совокупности, категории, обозначаемой вторым компонентом (корова - домашнее животное).

2. Дестинативная связь, один компонент которой обозначает назначение для другого компонента (этот овес для лошади).

3. Директивная связь, в которой один компонент обозначает путь, направление другого

компонента (идет в лес).

4. Инструментальная связь, один компонент которой обозначает орудие действия, обозначаемого другим компонентом (топор плотника).

5. Каузальная связь, один компонент которой обозначает причину проявления другого компонента спустя какое-то время (проросло посаженное зерно).

6. Комитативная связь, один компонент которой обозначает сопровождающее другой компонент действие, сопутствующий предмет, сопровождающее лицо (за самолетом потянулся след).

7. Коррелятивная связь, один компонент которой выражает возможность наблюдения другого компонента или соответствия предмета другому предмету, компоненту (в очках могу прочитать).

8. Негативная связь, один компонент которой отрицает, исключает возможность появления другого компонента (урожая не будет).

9. Лимитативная связь, один компонент которой обозначает сферу применения, назначения другого компонента (морковь - чтобы грызть).

10. Медиативная связь, один компонент которой имеет значение способа, средства действия другого (плывет на спине).

11. Поссесивная связь, один компонент которой выражает отношение владения другим компонентом (карандаш папы).

12. Потенсивная связь, в которой один компонент приводит к увеличению возможности появления другого спустя некоторое время (с удобрением растет быстрее).

13. Результативная связь, в которой один компонент выражает следствие действия второго (я посадил дерево).

14. Репродуктивная связь, в которой один компонент обозначает исходную точку для воспроизведения или превращения для другого компонента (пирог испекли в печи).

15. Ситуативная связь, в которой один компонент обозначает ситуацию, определяющую состояние или область действия второго компонента (свадьба состоится в деревне).

16. Трансагрессивная связь, в которой один компонент обозначает результат превращения второго (дрова превратились в золу).

17. Финитивная связь, в которой один компонент имеет значение цели, назначения другого (я поступил учиться).

Под семантической связью в [17] в общем случае также понимается отношение понятий в понятийной системе предметной области, употребляющийся в качестве синонима понятия предикат. Работа группы исследователей из Минска [18] содержит подробную классификацию отношений между понятиями. Ниже проведена систематизация и описание следующих 14 классов отношений, практически, охватывающих все отношения, касающиеся работ по объектно-предикатным системам. Названия классов и подклассов и примеры доступно отражают суть этих классов, и нет необходимости «накручивать» их дополнительными, строгими определениями.

1. Отношения классификации.

- Иметь имя. («Собаку звали Джек»).
- Класс-подкласс. («Органическое соединение - спирт»).
- Часть-целое. («Колесо трактора»).
- Элемент-класс. («Домашнее животное - корова»).
- Род-вид. («Млекопитающие - парнокопытные»).
- Вышестоящее-нижестоящее. («Ректор - декан»).
- Быть эталоном. («Победитель олимпиады»).

2. Признаковые отношения.

- Иметь признак. («Цвет объекта»).
- Иметь значение признака. («Синий»).

3. Количественные отношения.

- Иметь меру. («Вес объекта»).
- Иметь значение меры. («5 кг»).

4. Отношения сравнения.

- Равно. («Все стороны равностороннего треугольника равны»).
- Сравнимо. («Вес объекта и вес части объекта»).
- Больше. («Индюк больше курицы»).
- Больше или равно. («Количество дней в одном месяце больше или равно 28 »).
- Меньше. («Плотность льда меньше плотности воды»).
- Меньше или равно. («Количество листьев на дереве меньше или равно количеству почек»).
- Несравнимо. («Вес объекта и цвет объекта несравнимы»)

5. Отношения принадлежности. («Егор студент ВСГТУ»).

6. Временные отношения. («скорый поезд пришел после товарного»).

- Быть одновременно. («Марат и Азат пришли к началу занятий»).
- Быть раньше. («До яйца была курица»).
- Быть позже. («Яйцо появилось после курицы»).
- Совпадать во времени. («Время отлета самолета и отхода поезда в Москву - 19=00»).
- Пересекаться во времени. («В три часа обе машины будут проезжать Казань»).
- Быть внутри по времени. («В течение твоего пребывания в Казани мы сходим в театр»).
- Начинаться одновременно. («Свисток судьи оповестил о начале бега на 5 и 10 километров»).
- Кончаться одновременно. («Мое терпение лопнуло в тот момент, когда заглох мотор»).

7. Пространственные отношения.

- Совпадать в пространстве. («И шайба и клюшка оказались в воротах»).
- Быть слева. («Слева от дерева стояла машина»).
- Быть справа. («Справа от машины зеленело дерево»).
- Быть спереди. («Перед преподавателем сидели два студента»).
- Быть сзади. («Далеко за горами виднелись облака»).
- Наискосок. («Чуть сбоку от дороги вдали светились огни»).
- Пересекаться в пространстве. («Над деревом сошлись два облака»).
- Касаться. («Облака плыли касаясь крыши домов»).
- Находиться на. («Стол стоит на полу»).
- Быть сверху. («Перьевые облака плывут выше дождевых»).
- Быть снизу. («Подо льдом мирно текла река»).
- Находиться в. («В кабине сидело пять человек).

8. Каузальные отношения.

- Быть целью. («Мы хотим покорить вершину»).
- Быть мотивом. («Он нарушил клятву»).
- Причина-следствие. («Горячий уголь прожег материал»).

9. Инструментальные отношения.

- Служить для. («Бревно подпирает ворота»).
- Быть средством для. («Он доехал в лес на машине»).
- Способствовать. («Он предоставил ему свое ружье»).
- Быть инструментом. («Обезьяна палкой сшибла банан»).
- Быть вспомогательным средством. («У него на поясе висела веревка на случай сильного течения реки»).

10. Информационные отношения.

- Быть отправителем. («Он передал письмо для любимой).
- Быть получателем. («Мне сегодня пришло письмо»).
- Быть источником информации. («Он сообщил мне, что заказ готов»).

11. Порядковые отношения.

- Быть следующим. («После Сидоровых пришли Ивановы»).
- Быть очередным. («За весной настала очередь лета»).
- Быть ближайшим. («Зеленодольск - ближайший к Казани город»).

12. Модальные отношения.

- Возможность. («Самолет, который стоит на поляне полетит к вечеру»).
- Действительность. («На фоне заката летит самолет»).
- Необходимость. («Для вывоза урожая требуется пять бортовых машин»).

13. Модификаторы. («Желательно, чтобы Вы не опоздали к началу сеанса»).

14. Квантификаторы.

- Квантор общности. («Все студенты первого курса сдали экзамен по ЭВМ и программированию»).
- Квантор существования. («Нашелся студент, который не смог решить квадратное уравнение»).

Как видно по классификации М.З. Закиева, по работе минских исследователей выделение классов предикатов и объектов есть процесс перманентный, требующий глубокой лингвистической интуиции от автора. Очевидно, ни одна из рассмотренных классификаций не является полной и завершенной и навряд ли вызовет сомнение у исследователей утверждение, что вопросы полноты и достаточности объектно-предикатной системы могут решаться лишь в ходе практического ее использования, причем, лишь для какой-то фиксированной ситуации. Следовательно, весьма актуально иметь некий инструментарий для фиксирования выделенных объектов и отношений, а также автоматизированного поиска и установления их в огромных массивах машиночитаемых ЕЯ-текстов.

БИБЛИОГРАФИЯ

1. Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л. и др. Лингвистический процессор для сложных информационных систем. - М.: Наука, 1992. - 256 с.

2. Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л. и др. Лингвистическое обеспечение системы Этап-2. -М.: Наука, 1989.

3. Богуславский И.М., Цинман Л.Л. Семантический компонент лингвистического процессора // Семиотика и информатика - 1990. №32. С. 5-30.

4. Виноград Т. К процессуальному пониманию семантики // Новое в зарубежной лингвистике. Вып. XII.- М., 1983.

5. Жигалов В.А. Естественное общение с приложением // Открытые системы – 2001– №12.

6. Искусственный интеллект: В 3 кн. Кн.1. Системы общения и экспертные системы: Справочник / Под ред. Э.В. Попова. – М.: Радио и связь, 1990. – 464 с.

7. Искусственный интеллект: В 3 кн. Кн.2. Модели и методы: Справочник /Под ред. Д.А. Поспелова. - М.: Наука, 1990. – 472 с.

8. Искусственный интеллект: Применение в интегрированных производственных системах / Под ред. А.И. Дашенко, Е.В. Левнера. – М.: Машиностроение, 1991. – 539 с.

9. Мельчук И.А. Опыт теории лингвистических моделей “смысл-текст”. – М.: Наука , 1982.- 314 с.

10. Моделирование языковой деятельности в интеллектуальных системах / Под ред. А.Е. Кибрика и А.С. Нариньяни. - М.: Наука, 1987.

11. Нариньяни А.С. Лингвистические процессоры ЗАБСИБ (1-я и 2-я части). Препринт ВЦ СО АН СССР, №199. 1979.

12. Попов Э.В. Естественно-языковые системы: прошлое, настоящее и будущее. VI национальная конференция ИИ-2000, Переславль-Залесский, 24-27 октября 2000. - М.: ИФМЛ, 2000.- С. 17.

13. Попов Э.В. Общение с ЭВМ на естественном языке. – М.: Наука, 1982.

14. Хорошевский В. Ф. Обработка естественно-языковых текстов: от моделей понимания к технологиям извлечения знаний // Новости искусственного интеллекта – 2002 - №6 - С. 19-26.

15. Шабанов-Кушнаренко Ю.П. Теория интеллекта. Математические средства. - Харьков: Вища школа, 1984. - 143 с.

16. Бухараев Р.Г., Сулейманов Д.Ш. Об одном подходе к разработке интеллектуальных АОС // Кибернетика – 1986 - № 3. - С.42-49.

17. Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. -М.: Наука. Физматлит, 1997. -112 с. - (Проблемы искусственного интеллекта).

18. Экспертные системы для персональных компьютеров: методы, средства, реализации: Справочное пособие / Крисевич В.С., Кузьмич Л.А., Шиф А.М.и др. - Минск: Выш.шк., 1990. -197 с.

Инга Сергеевна Евдокимова

ЕСТЕСТВЕННО-ЯЗЫКОВЫЕ СИСТЕМЫ

Курс лекций

Редактор Е.В. Белоплотова

Ключевые слова: естественный язык, естественно-языковая система, морфологический анализ, семантический анализ, синтаксический анализ.

Подписано в печать 17.01.2005 г. Формат бумаги 60×84 1/16.
Усл. печ. л. 5,35, уч.-изд. л. 4,85. Печать операт., бум.писч.
Тираж 30 экз. Заказ № _____

Издательство ВСГТУ. 670013, г.Улан-Удэ, ул.Ключевская 40, в.