

УДК 519.711.2

ОНТОЛИНГВИСТИЧЕСКИЕ СИСТЕМЫ: ТЕХНОЛОГИИ ВЗАИМОДЕЙСТВИЯ С ПРИКЛАДНОЙ ОНТОЛОГИЕЙ

O.A. Невзорова

Аннотация

Онтологические системы ориентированы на решение сложных задач обработки естественного языка, требующих семантических знаний. В основе проектирования онтологических систем лежат процессы скоординированного взаимодействия различных уровней, прежде всего онтологического (обеспечивающего системные модели знаний о мире) и различных языковых уровней. В статье рассматриваются некоторые программные технологии взаимодействия с прикладной онтологией, разработанные при проектировании специализированной онтологической системы «ЛоТА», предназначенный для анализа специальных технических текстов «Логика работы системы...».

Введение

Повсеместная компьютеризация общества, становление и развитие сетевых информационных технологий способствовали переходу общества в новое качественное состояние глобальной информатизации. Автоматическая обработка текстов на естественном языке (ЕЯ) стала одной из приоритетных задач в области новых информационных технологий. Актуальными и востребованными являются технологии информационного поиска, извлечения знаний из текстов, автоматического реферирования, машинного перевода и др. В настоящее время исследования и разработки в области систем IE (Information Extraction) активно ведутся во всем мире [1–4]. Для задач информационных систем под извлечением информации (IE) понимается идентификация и семантическая классификация знаний, извлеченных из неструктурированных источников, таких как текст на ЕЯ. В последние годы задача IE интегрируется в более крупные приложения, такие как выбор (поиск) информации для различных целевых задач, задача принятия решений. Широко развивается направление исследований, связанное с моделированием онтологической семантики. Особо следует отметить размеченные ролевыми дескрипторами онтологические ресурсы, такие как FrameNet, PropBank и VerbNet для английского языка, Salsa для немецкого языка, Spanish FrameNet для испанского языка. Все эти ресурсы являются необходимой базой для задач семантической классификации.

Важнейшая роль семантических знаний всегда подчеркивалась в когнитивных исследованиях, на основании которых можно утверждать, что семантический уровень является уровнем, связующим все языковые уровни, то есть интегральным системообразующим свойством языковой системы.

Онтологические системы ориентированы на решение сложных задач обработки естественного языка, требующих семантических знаний [5]. В основе проектирования онтологических систем лежат процессы скоординированного взаимодействия различных уровней, прежде всего онтологического (обеспечивающего системные модели знаний о мире) и различных языковых уровней.

Целью создания онтолингвистических систем является обеспечение решения сложных задач обработки естественного языка путем организации системы взаимодействий различных языковых уровней, включая построение адекватной модели исследуемой предметной области и изучения лингвистически описанных свойств объектов предметной области и зависимостей между объектами. Моделирование предметной области осуществляется на основе онтологического подхода, интегрирующего знания экспертов и лингвистические знания.

Класс онтолингвистических систем отличается объединением экстралингвистических (онтологических) и лингвистических знаний, эвристических и формальных методов обработки ЕЯ. При этом роль и значение эвристических методов возрастают по мере роста сложности рассматриваемых лингвистических моделей.

1. Модель решения прикладной задачи в онтолингвистической системе

При проектировании лингвистических приложений предлагается использовать новый подход, центральной идеей которого является построение решения прикладной задачи на основе организации взаимодействия полифункциональной онтологической системы: прикладной онтологии, онтологии свойств и онтологии задач. Основная идея нового подхода заключается в следующем. Объектами лингвистического анализа являются текстовые документы, обработка которых производится для определенной целевой задачи. Тексты описывают совокупность объектов прикладной области, обладающих определенным набором свойств, важных для конкретной целевой ситуации. Иные целевые ситуации могут потребовать задания объектов с другим набором свойств. Другими словами, в разных задачах (практических целевых ситуациях субъектов) *объекты обладают разным набором свойств*, не только в разных проблемных областях, но и в одной и той же проблемной области, в которой решаются разные задачи.

Задача как некоторый *тип* практической ситуации субъектов в большей степени определяется их способом существования (структурой мышления и пр.), чем конкретной проблемной областью. Таким образом, можно предположить, что под множеством языка, маркирующее структуру событий, связанных с задачами, квазинезависимо от конкретной проблемной ситуации и соответствующей ей структуры свойств объектов, то есть выделяемо в отдельную **онтологию задач**. Тем самым можно выделить некоторое универсальное (пополняемое) множество базовых задач (типовых элементарных ситуаций), на основе которых можно с помощью определенной логики последовательностей конструировать более сложные задачи. Следовательно, решение прикладной задачи может быть спроектировано как **система взаимодействий трех онтологий**: прикладной онтологии проблемной области, онтологии свойств и онтологии базовых задач. Для каждой онтологии формируются свои концепты, совокупность текстовых входов концептов и связи между концептами, базирующиеся на ключевых для данной онтологии отношениях. При этом взаимодействие онтологий реализуется в разметке концептов прикладной онтологии концептами-свойствами для конкретных концептов-задач.

Онтология задач на уровне файловых представлений должна быть унифицирована с онтологиями свойств и прикладной онтологии. Выделяются следующие типы концептов онтологии задач: *задачи, операции, данные* (входные/выходные). Метод построения спецификаций прикладной задачи должен быть реализован как процессор (интерпретатор) со всеми свойствами программируемой среды, который настраивается на конкретный концепт-задачу и последовательно реализует базовые операции этой задачи. Соответствующая инструментальная среда должна



Рис. 1. Типовые функциональные компоненты онтолингвистической системы

быть выстроена как набор специализированных и универсальных базовых операций, управляющих процессом решения. Таким образом, любая задача, решаемая процессором, представляет собой концепт онтологии задач, связанный с другими концептами связями «принадлежности-следования». Онтология задач может быть связана с онтологией свойств через механизмы конкретизации параметров концептов-данных и значений метрик отношений, определенных на онтологии.

При проектировании технологии взаимодействия полифункциональной системы онтологических моделей необходимо обеспечить решение следующих основных задач:

- реализацию операций разметки концептов прикладной онтологии концептами-свойствами для конкретных концептов-задач;
- разработку механизма взаимодействия компонентов онтологической системы;
- разработку механизмов контроля целостности онтологической системы.

2. Архитектура онтолингвистической системы

В структуре онтолингвистической системы выделяются две основные взаимодействующие компоненты: онтологическая и лингвистическая. Типовой набор функциональных компонентов онтолингвистической системы представлен на рис. 1. Онтологическая компонента поддерживает проектирование системы онтологических моделей и обеспечивает взаимодействие с лингвистической компонентой при разработке лингвистических приложений. Разработка прикладных онтологий может опираться на существующие стандарты разработки онтологий и тезаурусных систем, а также иметь специфические методы.

Онтологическая подсистема обеспечивает поддержку решения следующих задач:

- онтологическая разметка исследуемых текстов элементами (концептами, отношениями) прикладной онтологии;
- извлечение информации из текстов (распознавание и интерпретация прикладных задач);
- онтологическая поддержка задач лингвистического анализа;
- разрешение грамматической и лексической многозначности;
- сегментация внутри предложения (частичный синтаксический анализ);
- разрешение референции и восстановление эллипсиса;
- поддержка онтологических выводов.

Лингвистическая компонента обеспечивает решение следующих лингвистических задач:

- распознавание символов (графематический анализ);
- сегментация предложений;
- распознавание типов лингвистических объектов (словоформы, числа, дата, время, аббревиатура и т. п.);
- морфологический анализ словоформ;
- разрешение грамматической и лексической многозначности;
- синтаксический анализ и разрешение синтаксической многозначности;
- разрешение референции и восстановление эллипса.

3. Проектирование экспериментальной онтолингвистической системы

Рассматриваемый подход реализован в проектировании экспериментальной онтолингвистической системы «ЛоТА», предназначенный для анализа специализированных технических текстов типа «Логика работы...» [6]. Основной задачей системы «ЛоТА» является извлечение из специализированного технического текста прикладной информационной модели схемы бортовых алгоритмов, решаящих определенную задачу в определенной проблемной ситуации, и контроль структурной и информационной целостности выделенной алгоритмической схемы [7].

Решение основной задачи обеспечивается комплексом технологий обработки текстов:

- технологии морфосинтаксического анализа;
- технологии семантико-синтаксического анализа;
- технологии взаимодействия с прикладной онтологией.

Указанная сумма технологий формируется на основе центрального ядра – прикладной онтологии (авиа-онтологии), обеспечивающей согласованное взаимодействие различных программных модулей. Авиа-онтология концептуально представляет предметную область информационного (алгоритмического) обеспечения различных полетных режимов антропоцентрических систем [8].

Разрабатываемая программная система содержит типовой набор компонентов онтолингвистической системы, а именно онтологическую и лингвистическую подсистемы. Основное внимание при данном подходе уделяется разработке механизмов совместного взаимодействия компонентов при решении конкретных задач обработки текста. Программный комплекс (рис. 2) состоит из трех взаимодействующих подсистем: подсистемы лингвистического анализа технических текстов «Анализатор», подсистемы ведения онтологии «OntoEditor+» и подсистемы «Интегратор». Взаимодействие подсистем реализовано на базе технологии «клиент-сервер», причем в различных подзадачах подсистемы выступают в различных режимах (режим сервера или режим клиента).

Инструментальная система визуального проектирования «OntoEditor+» [9] является специализированной СУБД. Система предназначена для ручного редактирования онтологий, хранящихся в реляционной базе данных в формате TPS, а также для обслуживания запросов пользователей и внешних программ. Новые возможности системы обеспечиваются функциональным набором «Лингвистический инструментарий», посредством которого реализуется встраивание прикладной онтологии в лингвистические приложения. Наиболее типичными задачами, решаемыми с помощью инструментария системы «OntoEditor+», являются изучение структурных свойств прикладной онтологии с помощью исследовательского инструментария системы «OntoEditor+», построение лингвистической оболочки прикладной онтологии, задача покрытия текста онтологическими входами, построение выводов по прикладной онтологии и др.

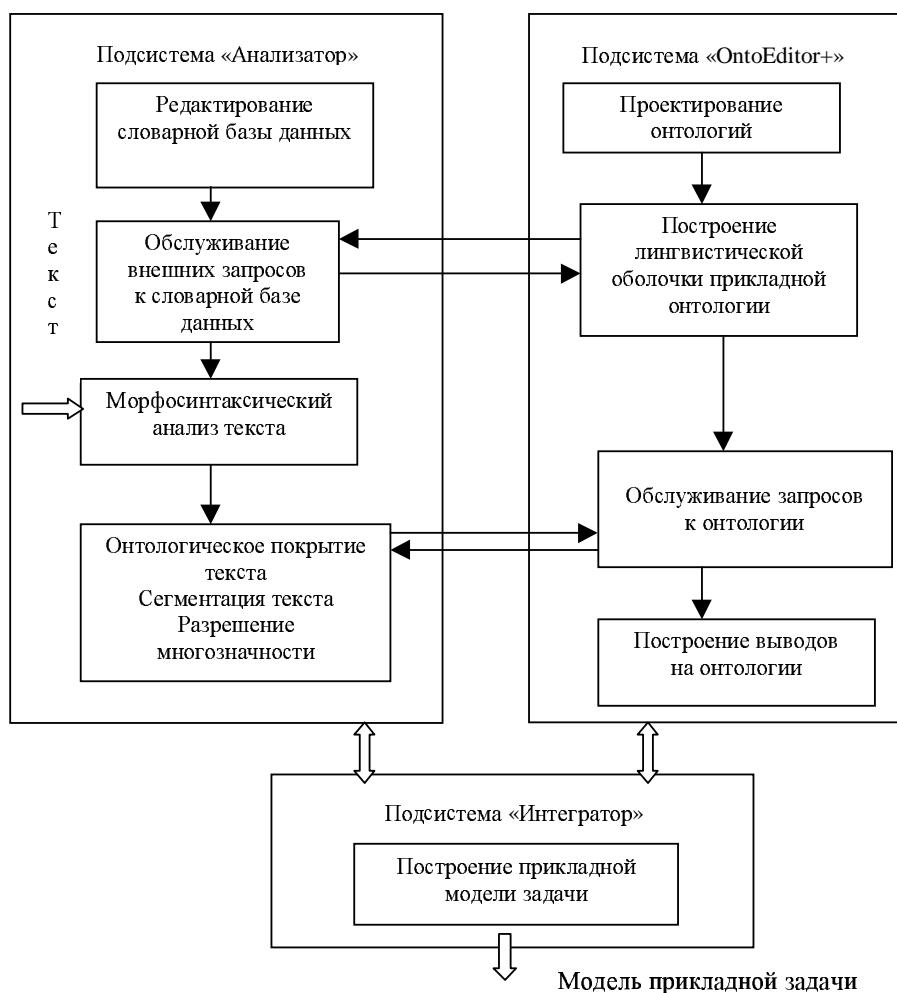


Рис. 2. Архитектура программного комплекса «ЛоТА»

Основные функциональные блоки системы «OntoEditor+» (блок настройки, блок проектирования онтологий, блок визуализации, лингвистический инструментарий, интерпретатор команд) имеют соответствующие специализированные наборы функций. Блок проектирования онтологий поддерживает основные табличные функции работы с онтологией (добавление, изменение, удаление записей; автоматическая коррекция записей; ведение нескольких онтологий, в том числе смешанных, то есть с общими списками типов отношений, классов, синонимов и др.; импорт онтологий различных форматов данных; фильтрация онтологий; ведение автоматической статистики по объектам онтологий; поиск цепочек отношений и др.). Блок визуализации обеспечивает графический режим проектирования онтологий. Система использует развитые механизмы фильтрации при работе в графическом режиме. Блок настройки позволяет задавать различные цветовые и графические образы, используемые для формирования графического образа онтологий, а также другие функциональные настройки. Лингвистический инструментарий реализует функции загрузки корпуса текстов; автоматическое ведение статистики по различным объектам корпуса; функции предсинтаксической обработки текста (сегментация предложений, распознавание аббревиатур, разрешение омонимии на

основе специальных протоколов взаимодействия с внешними словарными ресурсами); построение лингвистической оболочки онтологии; распознавание терминов прикладной онтологии во входном тексте (задача покрытия). Интерпретатор команд поддерживает различные типы запросов к онтологии, используя развитые механизмы фильтрации запроса. Система имеет многооконный интерфейс и снабжена развитой системой подсказок, а также механизмами поиска конкретной записи.

Подсистема «Анализатор» реализует основные этапы лингвистической обработки текста (графематический, морфосинтаксический). В подсистеме реализован частичный синтаксический анализ. Результат синтаксического анализа – дерево сегментов определенных типов, причем в основе сегментации лежит процедура распознавания в тексте онтологических входов концептов прикладной онтологии. Задача покрытия текста онтологическими входами решается на основе взаимодействия с подсистемой «OntoEditor+». Лингвистический инструментарий системы «OntoEditor+» позволяет эффективно распознавать в тексте различные лексические формы линейных онтологических входов. Усложненные синтаксические конструкции, в частности конструкции с однородными членами, распознаются механизмами синтаксического анализа подсистемы «Анализатор», и информация о структуре однородной группы передается на вход подсистемы «OntoEditor+».

В модуле «Построение прикладной модели задачи» подсистемы «Интегратор» формируется ответ на внешний запрос, в структуре которого определяется структура прикладной модели пользователя. Текущая версия системы связана с анализом структуры прикладной информационной модели алгоритма. Внешняя форма запроса интерпретируется при взаимодействии с подсистемой «OntoEditor+» как структура, привязанная к прикладной онтологии. Выделение компонент информационной модели происходит на основе механизмов отождествления элементов дерева сегментов входного текста (взаимодействие с подсистемой «Анализатор») и компонентов модели прикладной задачи (взаимодействие с подсистемой «OntoEditor+»). Этот процесс поддерживается различными видами онтологического поиска по запросам определенного типа (поиск концепта верхнего уровня, поиск цепочки определенного вида между двумя концептами и др.). Контроль информационной целостности текста по внешним критериям, в роли которых выступает внешний запрос (прикладная задача), заключается в распознавании и интерпретации ключевых элементов текста (лингвистических моделей) и установлении их соответствия прикладным моделям (соответствующим компонентам прикладной задачи).

4. Технологии взаимодействия с прикладной онтологией

В рамках развивающегося подхода разработан ряд специальных технологий взаимодействия с прикладной онтологией:

- технология построения лингвистической оболочки онтологии;
- специальная программная технология построения индексированной базы контекстов омонимов;
- интегрированная программная технология разрешения многозначности;
- технология разметки текста онтологическими входами;
- технология контроля информационной целостности данных.

Ниже в статье приводится концептуальное описание указанных технологий и пример программной реализации технологии разрешения функциональной (грамматической) омонимии.

Метод построения лингвистической оболочки онтологии [10] обеспечивает загрузку прикладной онтологии в лингвистическую оболочку для последу-

ющего ее использования в задачах обработки текстов. Создание лингвистической оболочки представляет собой многоэтапный процесс, в котором выделяются следующие шаги:

- определяется тип словоформ, составляющих онтологические входы. Данный этап реализуется с помощью запросов к подсистеме «Анализатор». Возможные типы словоформ (не опознано; не слово; слово; омоним (при этом передаются допустимые части речи); аббревиатура; фамилия; число; метка; стандартное сокращение);
- с помощью лингвистов-экспертов устраняется неоднозначность функциональной омонимии для словоформ-омонимов;
- в автоматическом режиме с помощью подсистемы «Анализатор» каждой словоформе онтологических входов приписывается номер словарной парадигмы (или несколько номеров, если словоформа обладает лексической омонимией);
- с помощью лингвистов-экспертов устраняется неоднозначность лексической омонимии для словоформ – лексических омонимов;
- в автоматическом режиме с помощью подсистемы «Анализатор» каждой словоформе онтологических входов присваивается набор грамматических характеристик (или несколько наборов, если словоформа обладает морфологической омонимией);
- с помощью лингвистов-экспертов устраняется неоднозначность морфологической омонимии для словоформ – морфологических омонимов;
- выполняется автоматическая проверка согласованности грамматических характеристик словоформ онтологических входов со словарем (с использованием подсистемы «Анализатор»);
- выполняется автоматическая проверка согласованности грамматических характеристик словоформ онтологического входа;
- задаются правила согласования словоформ онтологических входов при их применении в тексте сначала в автоматическом режиме, а потом с помощью лингвистов-экспертов.

Учитывая сложность и многоступенчатость перечисленных выше процедур, в подсистеме «Ontoeditor+» разработан мастер построения лингвистической оболочки, который вызывается командой основного меню.

Интегрированная программная технология построения индекса базы контекстов омонимов различных типов (функциональных, лексических) включает модули создания и ведения индекса омонимов, модуль согласования индексной базы с основным лингвистическим ресурсом – грамматическим словарем, а также механизмы выполнения внешних запросов по разрешению (поиску) типовых омонимических контекстов в текстовом корпусе на основе индекса омонимов.

Технология построения базы контекстов омонимов является автоматизированной, состоящей из нескольких этапов.

На первом этапе с помощью лингвиста-эксперта разрабатываются текстовые модели, разрешающие контексты омонимов (функциональных и лексических), которые описываются на определенном формальном языке. При наличии специальных меток на основе модели разрешающих контекстов может быть автоматически сгенерировано подмножество контекстных экземпляров.

На втором этапе происходит импорт подготовленных для генерации текстовых файлов, генерация подмножества контекстных экземпляров, а также контроль хода генерации (проверки согласованности грамматических характеристик, формата модели контекста и др.). Все изложенные выше механизмы осуществляются при

взаимодействии двух подсистем: подсистемы «OntoEditor+» (онтологическая компонента) и подсистемы «Анализатор» (лингвистическая компонента). Взаимодействие подсистем реализовано на основе клиент-серверной технологии и поддерживается различными специализированными протоколами обмена.

Интегрированная программная технология разрешения многозначности является комплексной технологией, объединяющей три разработанные программные технологии. Первая технология – технология разрешения функциональной омонимии на основе контекстных правил [11]. Основу этой технологии составляет разработка обобщенных контекстных правил разрешения омонимии (каждое обобщенное правило разрешает определенный функциональный тип омонима). В настоящее время разработано 15 обобщенных правил наиболее частотных типов функциональных омонимов, в том числе правила для сложных случаев типа разрешения (например, для омонимов это, все/всё и др.).

Вторая технология разрешения омонимии базируется на использовании индексируемой базы контекстов омонимов. Этот метод позволяет эффективно разрешать как функциональную, так и лексическую омонимию. Механизмы разрешения основаны на распознавании контекстов омонимов во входных предложениях. Модель контекста омонима имеет ряд распознаваемых параметров (грамматические характеристики компонентов коллокации, расстояние до разрешающей словоформы), при обнаружении которых выдается информация о типе омонима и его грамматических характеристиках.

Третья технология разрешения омонимии использует лингвистическую оболочку онтологии, то есть грамматическую информацию об онтологических концептах и их текстовых (синонимических) формах. Интегральный метод разрешения омонимии реализует весь комплекс перечисленных выше технологий. Первоначально осуществляется поиск в базе контекстов омонимов, при отсутствии необходимой информации о разрешении омонимии запускаются процедуры разрешения на основе контекстных правил.

Интегральный метод позволяет динамически фиксировать новые контексты разрешения и добавлять их в индексируемую базу контекстов омонимов.

На рис. 3 приведена графическая форма результата трассировки решения прикладной задачи «Анализ функциональной омонимии в тексте» в виде сформированной последовательности решаемых подзадач. Выделенные подзадачи представляют собой некоторое подмножество базовых задач онтологии задач, связанных отношением последовательности при конструировании процесса решения прикладной задачи.

Метод онтологической разметки текста на основе прикладной онтологии реализуется на основе специальных протоколов обмена между подсистемой «Анализатор» (клиент) и подсистемой «OntoEditor+» (сервер). Распознавание линейных онтологических входов в тексте осуществляется на основе грамматических описаний, заданных в лингвистической оболочке онтологии. Нелинейные синтаксические структуры, в частности конструкции с однородными членами определенных типов, распознаются специальными механизмами подсистемы «Анализатор», и соответствующие составляющие (члены однородной группы) передаются для распознавания подсистеме «OntoEditor+».

Особую сложность представляет выделение составляющих внутри сочинительных конструкций. При образовании сочинительных конструкций используются различные механизмы синтаксического сокращения и решение обратной задачи (выделение составляющих) не всегда является однозначным. Например, в сочинительной конструкции «прикрытие бомбардировщиков и штурмовиков в районе боевых действий» выделяются составляющие «прикрытие бомбардировщиков в рай-

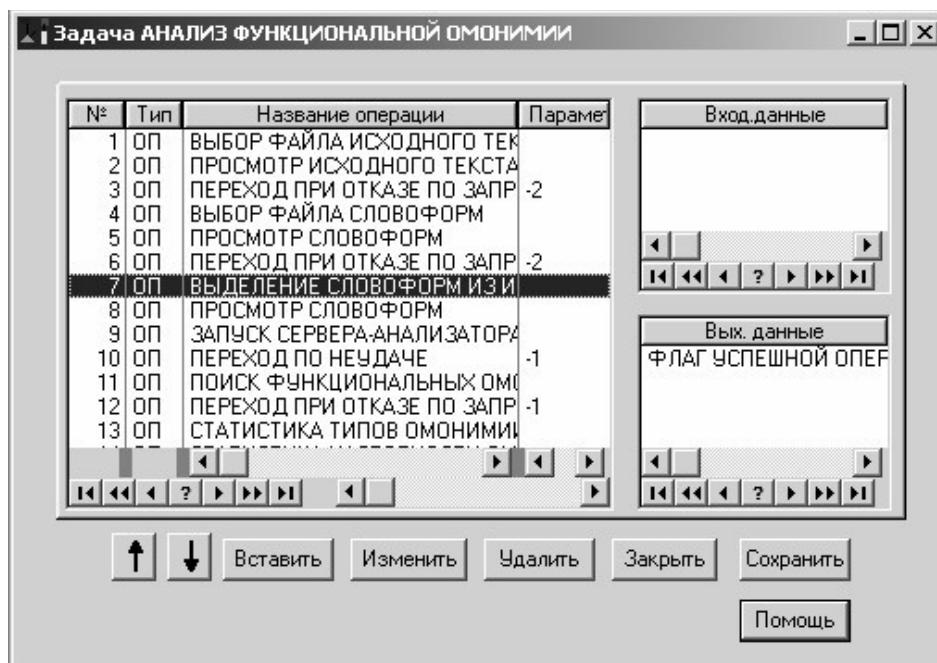


Рис. 3. Графическая форма результата трассировки решения прикладной задачи

оне боевых действий» и «прикрытие штурмовиков в районе боевых действий», однако в других случаях предложно-падежная группа (типа «в районе боевых действий») может не являться общим элементом составляющих. Выделение составляющих из сочинительных конструкций производится на основе специальных правил, которые учитывают явление «семантической однородности». Семантическая однородность предполагает построение синтаксических конструкций с семантически однородными членами, то есть члены однородных конструкций должны относиться к одному семантическому классу. На этапе построения правил выделяются два основных семантических класса: класс предметных имен и класс непредметных имен. Семантическая однородность допускает построение синтаксических конструкций либо для предметных, либо для непредметных сущностей. Например, допустимыми являются конструкции типа «самолеты и ракеты противника» (предметная однородность), либо «перехват и уничтожение противника» (непредметная однородность).

Синтаксические конструкции с однородными определениями составляют другой тип синтаксического сокращения. В этом случае выделяется группа составляющих с одиночными определениями. Так, например, однородная синтаксическая группа типа «естественные и искусственные помехи» распознается как состоящая из элементов «естественные помехи» и «искусственные помехи».

Все составляющие сложных синтаксических конструкций затем отождествляются как онтологические входы. С каждым распознанным в тексте онтологическим входом передается информация об онтологическом концепте и его семантическом классе (концепте верхнего уровня по иерархии). Метод распознает различные ситуации распределения онтологических входов в предложении. При вложении сегментов как результат передается сегмент максимальной длины, при перекрытии сегментов передаются все перекрывающиеся составляющие.

Заключение

Класс онтолингвистических систем отличается объединением экстравелингвистических (онтологических) и лингвистических знаний, эвристических и формальных методов обработки ЕЯ. Ядром онтолингвистических систем являются знания различной природы, в том числе различные онтологии, представляющие прикладные знания, метазнания, а также знания о прикладных задачах и их свойствах.

Основные технологические задачи онтолингвистических систем решаются через взаимодействие онтологической и лингвистической компонент системы. Соответствующие технологии взаимодействия позволяют решать общие и специальные задачи анализа текстов, при этом общая структура решаемой задачи может динамически меняться через специальные механизмы настройки типа решаемой задачи. В статье приведены концептуальные решения по проектированию технологий взаимодействия с прикладной онтологией в онтолингвистической системе «ЛоТА». Эксперименты по построению указанных технологий показали перспективность предложенного подхода и направления его дальнейшего усовершенствования. В частности, требуется проведение тщательного системного анализа при выделении множества свойств объектов прикладных задач и классификации структуры прикладных задач в лингвистических приложениях.

Работа выполнена при финансовой поддержке РФФИ (проект № 05-07-90257).

Summary

O.A. Nevzorova. Ontolinguistic system: some technologies of interaction with applied ontology.

Ontolinguistic systems are focused on the decision of challenges of the natural language processing demanding semantic knowledge. The processes of coordinated interaction of various levels of text processing, first of all ontological (providing system models of knowledge of the world) and various language levels form the basis of the designing of ontolinguistic systems. Some program technologies of interaction with applied ontology developed for designing specialized ontolinguistic system “LoTA” are discussed in the paper.

Литература

1. Хорошевский В.Ф. OntosMiner: семейство систем извлечения информации из мультиязычных коллекций документов // Тр. IX нац. конф. по искусственному интеллекту с международным участием КИИ-2004. – М.: Физматлит, 2004. – Т. 2. – С. 573–581.
2. Александровский Д.А., Кормалев Д.А., Кормалева М.С., Куршев Е.П., Сулейманова Е.А., Трофимов И.В. Развитие средств аналитической обработки текста в системе ИСИДА-Т // Тр. десятой нац. конф. по искусственному интеллекту КИИ-2006. – М.: Физматлит, 2006. – Т. 2. – С. 555–563.
3. Hobbs Jerry R. Information extraction from biomedical text // J. Biomedical Informatics. – 2002. – V. 35, No 4. – P. 260–264.
4. Enrigue A., Conzalo J., Peinado V., Penas A., Verdejo F. An empirical study of information synthesis tasks // Proc. of the 42th Annual Meeting of Association for Computational Linguistics. – East Stroudsburg, PA: ACL, 2004. – P. 208–215.
5. Невзорова О.А. Онтолингвистические системы: методологические основы построения // Науч. сессия МИФИ-2007: Сб. науч. тр. Т. 3. Интеллектуальные системы и технологии. – М., 2007. – С. 84–85.

6. *Невзорова О.А., Федунов Б.Е.* Система анализа технических текстов «ЛоТА»: основные концепции и проектные решения // Изв. РАН. Теория и системы управления. – 2001. – № 3. – С. 138–149.
7. *Невзорова О.А.* Подход к разработке методов автоматизированного контроля информационной целостности технических текстов // Тр. десятой нац. конф. по искусственному интеллекту КИИ-2006. – М.: Физматлит, 2006. – Т. 2. – С. 564–571.
8. *Добров Б.В., Лукашевич Н.В., Невзорова О.А., Федунов Б.Е.* Методы и средства автоматизированного проектирования прикладной онтологии // Изв. РАН. Теория и системы управления. – 2004. – № 2. – С. 58–68.
9. *Невзоров В.Н., Невзорова О.А.* Система визуального проектирования онтологий «OntoEditor»: функциональные возможности и применение // Тр. IX нац. конф. по профессиональному интеллекту с международным участием КИИ-2004. – М.: Физматлит, 2004. – Т. 3. – С. 937–945.
10. *Невзорова О.А.* Инструментальная система визуального проектирования онтологий «OntoEditor+» в лингвистических приложениях // Вестн. КГТУ им. А.Н. Туполева. – 2006. – № 3. – С. 56–60.
11. *Невзорова О.А., Зинькина Ю.В., Пяткин Н.В.* Метод контекстного разрешения функциональной омонимии: анализ применимости // Тр. междунар. конф. «Диалог'2006», – М.: Наука, 2006. – С. 399–402.

Поступила в редакцию
10.09.07

Невзорова Ольга Авенировна – кандидат технических наук, доцент, ведущий научный сотрудник НИИ математики и механики им. Н.Г. Чеботарева Казанского государственного университета, доцент кафедры математики, информатики и методики преподавания информатики Татарского государственного гуманитарно-педагогического университета, г. Казань.

E-mail: *olga.nevzorova@ksu.ru*