МЕТОДЫ РЕШЕНИЯ ЛИНГВИСТИЧЕСКИХ ЗАДАЧ НА ОСНОВЕ ОНТОЛОГИЙ

Ольга Невзорова, Владимир Невзоров, Николай Пяткин

***Аннотация**: Онтолингвистические системы ориентированы на решение сложных задач обработки естественного языка, требующих семантических знаний. В основе проектирования онтолингвистических систем лежат процессы скоординированного взаимодействия онтологических и лингвистических моделей. В статье рассматриваются методы решения лингвистических задач на основе онтологий, разработанные при проектировании специализированной онтолингвистической системы «ЛоТА», предназначенной для анализа специальных технических текстов «Логика работы системы… ».*

***Ключевые слова**: онтолингвистические системы, онтологии, компьютерная лингвистика*

***ACM Classification Keywords**: I.2.7 Natural Language Processing*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

Повсеместная компьютеризация общества, становление и развитие сетевых информационных технологий способствовали переходу общества в новое качественное состояние глобальной информатизации. В этой связи активно стали развиваться информационные технологии обработки текстовой информации. 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 создания систем IE (Information Extraction) активно ведутся во всем мире [1-4]. Под извлечением информации (IE) понимается идентификация и семантическая классификация знаний, извлеченных из неструктурированных источников, таких как текст на ЕЯ, для задач информационных систем. 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, т.е. интегральным системообразующим свойством языковой системы.

Онтолингвистические системы ориентированы на решение сложных задач обработки естественного языка, требующих семантических знаний. В основе проектирования онтолингвистических систем лежат процессы скоординированного взаимодействия процессы скоординированного взаимодействия онтологических и лингвистических моделей. Целью создания онтолингвистических систем является обеспечение решения сложных задач обработки естественного языка путем организации системы взаимодействий различных языковых уровней, включая построение адекватной модели предметной области, исследование 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 возрастает по мере возрастания сложности рассматриваемых лингвистических моделей.

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 подмножество 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. Таким образом, 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 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.

Архитектура онтолингвистической системы

В структуре онтолингвистической системы выделяются две основные взаимодействующие компоненты: онтологическая и лингвистическая.

Онтологическая компонента позволяет поддерживать онтологическое моделирование предметной области. Разработка онтологической компоненты может опираться на существующие стандарты разработки онтологий и тезаурусных систем, а также иметь специфические методы.

Функционирование онтологической системы обеспечивает поддержку решения следующих задач:

- семантическая разметка исследуемых текстов элементами (концептами, отношениями) онтологической системы;
- извлечение информации из текстов (распознавание и интерпретация концептуальных структур);
- 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 поддержку решения следующих лингвистических задач:

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

Методы решения лингвистических задач

Рассматриваемый подход реализован в проектировании онтолингвистической системы «ЛоТА», предназначенной для анализа специализированных текстов типа "Логика …" [Невзорова&Федунов, 2001].

Основной задачей системы "ЛоТА" является извлечение из специализированного технического текста информационной модели схемы бортовых алгоритмов, решающих определенную задачу в определенной проблемной ситуации, и контроль структурной и информационной целостности выделенной алгоритмической схемы.

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 (авиаонтологии), обеспечивающей согласованное взаимодействие различных программных модулей. Авиаонтология концептуально представляет предметную область информационного (алгоритмического) обеспечения различных полетных режимов антропоцентрических систем [Добров и др., 2004].

Разрабатываемая программная система содержит типовой набор компонентов онтолингвистической системы. Основное внимание при данном подходе уделяется разработке механизмов совместного взаимодействия компонентов при решении конкретных задач обработки текста.

Программный комплекс состоит из двух взаимодействующих подсистем: подсистемы лингвистического анализа технических текстов "Анализатор", подсистемы управления и ведения онтологии "OntoIntegrator". Взаимодействие подсистем реализовано на базе технологии "клиент-сервер", причем в различных подзадачах подсистемы выступают в различных режимах (режим сервера или режим клиента) [Невзорова, 2006].

В рамках развиваемого подхода разработаны методы решения различных лингвистических задач:

- задача построения лингвистической оболочки онтологии;
- задача построения индексированной базы контекстов омонимов;
- задача разрешения многозначности;
- задача онтологической разметки текста;
- задача сегментации текста.

Метод построения лингвистической оболочки онтологии обеспечивает загрузку прикладной онтологии в специальную лингвистическую оболочку для последующего ее использования в задачах обработки текстов.

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 механизмы выполнения внешних запросов по разрешению (поиску) типовых омонимических контекстов в текстовом корпусе на основе индекса омонимов.

Интегрированная программная технология разрешения многозначности является комплексной технологией, объединяющей три разработанные программные технологии. Первая технология - технология разрешения функциональной омонимии на основе контекстных правил. Метод разрешения многозначности на основе контекстных правил позволяет разрешать функциональную (грамматическую омонимию) на основе контекстных правил, которые формулируются как результат тщательной лингвистической экспертизы поведения омонима в современных корпусах русского языка. В настоящее время разработано свыше 40 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, т.е. грамматическую информацию об онтологических концептах и их текстовых (синонимических) формах.

Интегральный метод разрешения омонимии реализует весь комплекс перечисленный выше технологий. Первоначально осуществляется поиск в базе контекстов омонимов, при отсутствии необходимой информации о разрешении омонимии запускаются процедуры разрешения на основе контекстных правил.

Метод онтологической разметки текста распознает в тексте онтологические концепты. Реализация метода базируется на специальных протоколах обмена между подсистемой "Analyzer" (клиент) и подсистемой "OntoIntegrator" (сервер). Распознавание линейных онтологических входов в тексте осуществляется на основе грамматических описаний, заданных в лингвистической оболочке онтологии. Новые результаты получены при разработке методов распознавания онтологических единиц, подвергшихся сочинительному сокращению в тексте. При анализе сочиненных синтаксических конструкций определенных типов решается обратная задача выделения потенциальных составляющих конструкции и их распознавание как самостоятельных онтологических единиц. На основе разработанных

механизмов в тексте распознаются синтаксические конструкции с однородными членами, а также некоторые типы симметричных конструкций. Например, в синтаксической конструкции "*атаки пар и звеньев истребителей"* выделяются составляющие "*атаки пар истребителей*" и "*атаки звеньев истребителей*", которые распознаются как отдельные онтологические единицы.

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 районе 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, т.е. члены однородных конструкций должны относится к одному семантическому классу. На этапе построения правил выделяются два основных семантических класса: класс предметных и 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.

Задача сегментации текста на составляющие – сегменты является одной из ключевых в процессе анализа текста. Результатом сегментации предложения является иерархическая совокупность семантико-синтаксических сегментов. Выделенные сегменты являются "блоками", из которых собираются по тексту информационные модели, определяемые типом решаемой задачи. Сегмент имеет синтаксический и семантический тип. Выделяются два главных семантических типа сегментов: группа субъекта и группа предиката предложения. С главными семантическими типами связаны подмножества синтаксических типов. Синтаксический тип определяет синтаксическую модель сегмента. Выделенное подмножество синтаксических моделей исчисляет синтаксические шаблоны для главных семантических типов. Распределение главных семантических типов по синтаксическим моделям фактически задает различные синтаксические структуры предложений русского языка. Текущая версия модуля сегментации поддерживает сегментацию базовых классов моделей русского предложения, а именно полных (двусоставных) предложений с группой субъекта в форме N*им/Abb (сущ./местоименное сущ. в именит. падеже или аббревиатура) и глагольным предикатом. В этих ограничениях количественные оценки синтаксических моделей для группы субъекта – 4 подтипа, для глагольной группы – 11 подтипов (простой глагольный, осложненный частицей, составной глагольный, именной составной). Помимо главных семантических типов выделяются семантические сегменты с дополнительной семантической характеристикой – атрибутивные сегменты. Атрибутивные сегменты имеют собственное подмножество синтаксических моделей. Алгоритмы сегментации на основе синтаксических моделей выделяют главные семантические сегменты и их расширения в виде атрибутивных сегментов. Сегменты, не вошедшие в

расширенные модели главные сегментов, интерпретируются как атрибуция предложения в целом (например, сегменты - локативы или сегменты - темпоративы).

Задача сегментации решается совместно с задачей онтологической разметки. Построение сегментов осуществляется в границах распознанных онтологических входов. Процесс организации взаимодействия подсистем при решении задач сегментации и онтологической разметки приведен на рис. 2.

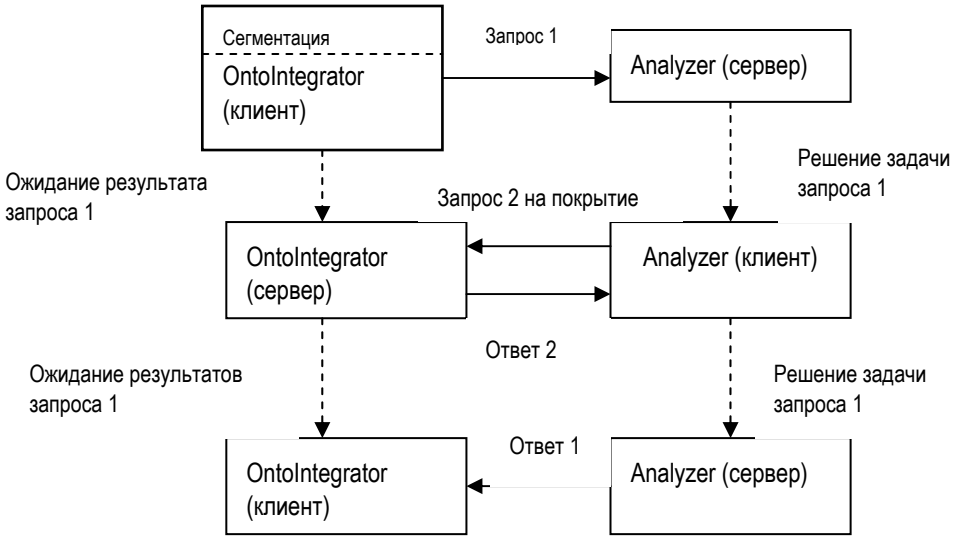

Рис.2. Взаимодействие подсистем при решении задачи сегментации

Запрос на решение задачи сегментации передается от подсистемы OntoIntegrator к системе Analyzer. Для решения этой задачи подсистема Analyzer запрашивает у подсистемы OntoIntegrator информацию об онтологической разметке текста. Полученная информация используется для уточнения границ сегментов.

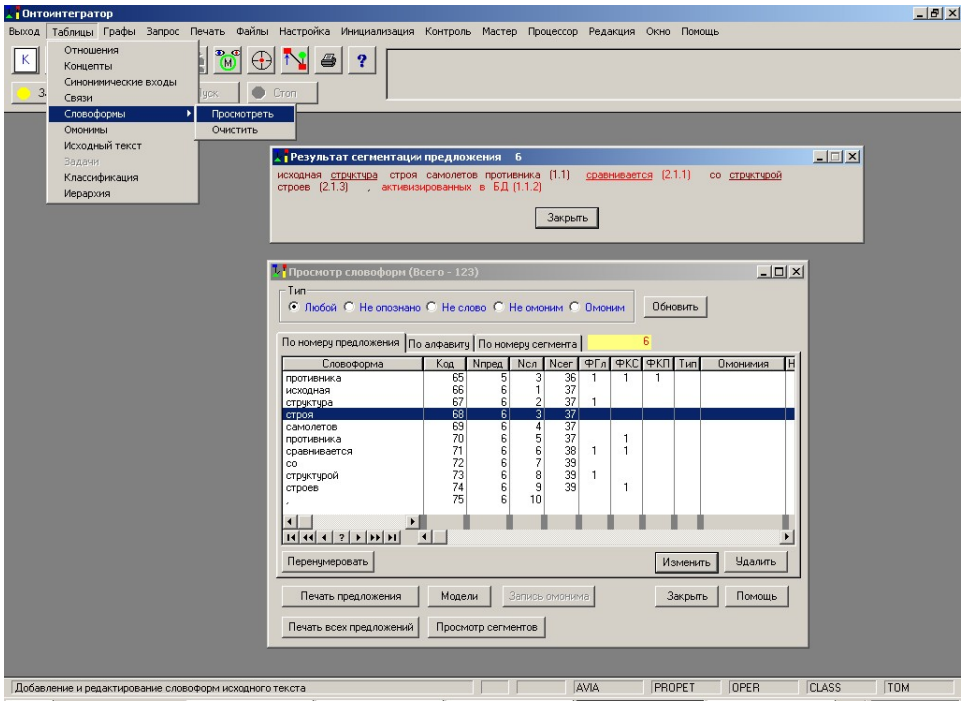

Рис. 3. Вывод результатов сегментации предложения

На рис. 3. показан результат сегментации предложения "(*Исходная структура строя самолетов противника) (сравнивается) (со структурой строев),(активизированных в БД.)".* Построенные сегменты заключены в круглые скобки, каждому сегменту приписан семантический тип. Если сегмент содержит в своих границах онтологический концепт, то информация о семантическом классе онтологического концепта учитывается при определении семантического типа сегмента.

Заключение

Класс онтолингвистических систем отличается объединением экстралингвистических (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, в том числе знания о прикладных задачах и их свойствах.

Основные лингвистические задачи решаются через взаимодействия онтологической и лингвистической компонент онтолингвистической системы, при этом общая структура решаемой задачи может динамически меняться через специальные механизмы настройки типа решаемой задачи.

Благодарности

Данная работа выполнена при поддержке Российского Фонда Фундаментальных исследований, грант № 08-07-90407.

Литература

[Невзорова&Федунов, 2001] 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.

[Добров и др., 2004] Добров Б.В., Лукашевич Н.В., Невзорова О.А., Федунов Б.Е. Методы и средства автоматизированного проектирования прикладной онтологии // Известия РАН. Теория и системы управления. М.: 2004. № 2. С. 58-68.

[Невзорова, 2006] Невзорова О.А. Подход к разработке методов автоматизированного контроля информационной целостности технических текстов //Труды десятой национальной конференции по искусственному интеллекту КИИ-2006. Том 2. М.,Физматлит, 2006. С. 564-571.

Authors' Information

Ольга Невзорова *– НИИММ им. Н.Г. Чеботарева, Татарский государственный гуманитарно-педагогический университет, Казань, Россия; e-mail: olga.nevzorova@ksu.ru*

Владимир Невзоров *– Казанский государственный технический университет им. А.Н. Туполева, Россия; e-mail: nevzorov@mi.ru*

Николай Пяткин *– НИИММ им. Н.Г. Чеботарева, Казань. Россия; e-mail: nikolaip@mail.ru*