И.М. Ножов

Морфологическая и синтаксическая обработка текста (модели и программы) [1]

Научный руководитель -
доктор технических наук,
профессор Д.Г. Лахути

Научный консультант -
Т.Ю. Кобзарева.

Москва - 2003

[1] Internet-публикация содержит исправления и сокращения оригинального текста диссертации, а также изменено первоначальное название *«Реализация автоматической синтаксической сегментации русского предложения»*.

ВВЕДЕНИЕ

Синтаксический анализ является одним из наиболее исследованных направлений в теории computer science. Синтаксические анализаторы широко применяются в таких областях как создание компиляторов, проектирование интерфейсов баз данных, искусственный интеллект (ИИ), автоматическая обработка текстов (АОТ), в том числе для автоматизированных информационно-поисковых систем (АИПС, или «поисковых машин»),

машинный перевод (МП), анализ химических формул и распознавание хромосом. Синтаксическим анализом (parsing) называется процесс структурирования линейной репрезентации в соответствии с заданной грамматикой [D.Grune, C.Jacobs, 1990]. Такое определение, являясь наиболее общим и абстрактным, позволяет охватить весь спектр приложений синтаксических методов. Техникой parsing называется вся совокупность существующих алгоритмов для решения задач синтаксического анализа. Техника parsing берет свое начало в формальных синтаксических теориях естественного языка (ЕЯ), моделирующих механизм распознавания человеком языковых структур. Несмотря на это, именно применение техники parsing в задачах автоматической обработки текста далеко не всегда бывает эффективным и дает положительный результат. Так, например, контекстно-свободные грамматики (context-free grammars) и аппарат конечных автоматов (finite-state automata) широко используются в системах морфологического анализа, снятия омонимии и выделения именных групп внутри предложения, но теряют свое прикладное значение в задачах сегментационного, полного синтаксического и семантического анализа, особенно для языков с относительно свободным порядком слов, каким является русский. Формальные математические модели и их программные динамические реализации не способны охватить всю сложность и многообразие языковой системы. Применение формализма для структурирования предложения естественного языка зачастую приводит к потере правильного синтаксического представления или комбинаторному взрыву, когда программа оказывается не в состоянии просчитать все возможные варианты структур. Лингвистически мотивированные причины такого "провала" - явление омонимии, длина связи между словами, сочинительные конструкции, нарушающие древесность графа, и сложность сегментной структуры предложения. Сфера действия методов распознавания и классификации объектов в лингвистических процессорах тоже сильно ограничена: скрытые модели Маркова удается применить только в узких контекстно-ограниченных задачах снятия морфологической омонимии [Xerox, 1999], нейронные сети используются в системах автоматического распознавания речи [С. Гладунов, О. Федяев, 2002], - такие модели, построенные на обучении и являющие собой альтернативный технике parsing подход, не имеют достаточной силы для отражения способности предложения

естественного языка к неограниченному усложнению. Все эти обстоятельства позволили прикладной (компьютерной) лингвистике выделиться в отдельную область исследования и стать самостоятельно развивающейся ветвью искусственного интеллекта.

Далее в работе мы будем использовать понятие синтаксического анализа только применительно к предложению естественного языка.

Взаимодействие между лингвистикой и computer science началось еще полвека назад с возникновением теории Н. Хомского, развитием генеративизма и появлением электронно-вычислительных машин. Многие лингвистические идеи и концепции на протяжении последних десятилетий были заимствованы и воплощены в программировании, теоретической информатике и информационных системах. Наиболее яркими примерами такого заимствования могут служить базисный компонент порождающей грамматики Н. Хомского, который стал прототипом первых компиляторов искусственных языков, или выдвинутая М. Мински, исследователем в области ИИ, теория фреймов для представления реальных объектов в системах распознавания образов и естественных языков [Г. Буч, 2000], которая сыграла свою роль как в становлении объектно-ориентированного подхода в программировании, так и в семантических исследованиях языка, а наследование и полиморфизм - фундаментальные принципы объектно-ориентированного программирования - стали применяться в проектировании лексиконов [I. Sag, T. Wasow, 1999].

Существует и удивительная связь между естественными и искусственными языками, которая заключается в закономерности эволюции языков. Первый опыт программирования в машинных кодах или на языках низкого уровня, к которым относится ассемблер, характеризуется скорее командным (императивным) стилем, где только упорядоченная последовательность операторов (команд) образует осмысленное действие, подобно тому как в языках с развитым словообразованием последовательная конкатенация грамматических аффиксов порождает слово, обладающее новым значением. С развитием таких языков как ALGOL-60 или COBOL усложняются синтаксические конструкции языка, появляется блочная структура программ. В следующем поколении языков, Pascal и C, текст программы становится похож на многопролетные лестницы, возможность описывать логику действий развернутыми синтаксическими конструкциями задает "ступенчатую" форму

текста. Последнее поколение объектно-ориентированных языков (CLOS, Object Pascal, C++ и Java) стремятся к описанию ключевых абстракций предметной области; абстракции объединяются в библиотеки классов, а программы оперируют объектами этих классов, вызывая методы классов и используя свойства классов, тем самым упрощая синтаксические конструкции, но усложняя структуру объектов и семантические зависимости между ними; текст современный программы напоминает набор коротких четверостиший или деклараций, где каждая строка - обращение к объекту со своим значением и сложной семантикой. Нечто подобное наблюдается и в процессе эволюции естественных языков, когда постепенное вырождение словоизменительной парадигмы в морфологии приводит к ужесточению порядка слов в предложении и фиксации жестких синтаксических конструкций, а последующее усложнение семантики, за счет насыщения языка идиомами и фраземами, за счет появления более абстрактных понятий или новых значений старых слов или за счет пополнения общеупотребительной лексики из научных метаязыков, приводит к упрощению синтаксиса. Конечно, такой сценарий развития не является обязательным и предопределенным для многих языковых групп и семей, но такой путь эволюции до некоторой степени справедлив для италийской группы индоевропейских языков - от латыни к современному итальянскому и французскому - и для группы германских языков.

Разумеется, что такое сравнение программных и естественных языков является во многом условным, но одно можно утверждать с полной уверенностью: "изменчивость - глубинное и универсальное свойство" [С. Бурлак, С. Старостин, 2001] как естественных, так и искусственных языков. Очевидно то, что направления векторов развития систем естественного и искусственного языков совпадают, как и то, что история человеческого языка насчитывает тысячелетия, а искусственных пять десятилетий. Возможно, именно глобальность задачи и разнообразие явлений синтаксиса предложения помноженное на число существующих на земле языков с развитой письменностью оправдывает разработку новых моделей и алгоритмов, отличных от общепризнанных техник parsing или математических моделей, успешно используемых в других областях человеческого знания.

Теоретическая лингвистика и типологический опыт исследования языков создали необходимый описательный аппарат для компьютерного

моделирования автоматического анализа текстов. Множество теоретических подходов можно разделить на два основных направления: формализм и функционализм. Формализм утверждает, что язык есть врожденная компонента человеческого мышления, которая может быть представлена в виде абстрактной модели на метаязыке формальной грамматики и не зависит от способов использования языка, а функционализм напротив полагает, что строение языка определяется его использованием [Я. Тестелец, 2001]. Исследования в формальной лингвистике можно тоже условно разделить на два подхода: построение универсальной грамматики, верной для всех существующих языков мира, и построение формальной модели, наиболее полно охватывающей все множество грамматических явлений конкретного языка. Н. Хомский стал родоначальником первого подхода и основателем школы генеративистов, самым ярким представителем второго подхода является И. Мельчук, автор модели "Смысл ⇔ Текст".

В задачах автоматической обработки текста (АОТ), как правило, используются концепции, разработанные в рамках формализма. Совмещая два подхода формальной лингвистики, программные модели являются лишь частичной реализацией теоретических исследований.

Работы по созданию синтаксического модуля велись еще в конце 60-ых годов, но вычислительная мощность компьютеров не позволяла реализовать сложные алгоритмы анализа в полном объеме. Упрощение алгоритмов и отказ от перебора омонимичных вариантов - компромисс, который приводил к низкой точности синтаксического анализа предложения. Сегодня, по-прежнему, задача автоматизированного анализа синтаксиса ЕЯ сводится к двум параметрам: качеству, определяемому парой «точность (уровень ошибок в построенных синтаксических структурах предложений), полнота (степень покрытия текста синтаксическими связями, или связность графа предложения)», и скорости, пока что недостаточной для ряда прикладных задач.

Ниже будут введены несколько определений понятий, связанных с синтаксическим анализом естественного языка, которые позже получат более точные формулировки. Линейной репрезентацией предложения естественного языка называется цепочка элементов, где каждый элемент является минимальной синтаксической единицей. Минимальная синтаксическая единица может быть словоформой или оператором с определенным набором

характеристик. Оператором называется знак препинания или сочинительный союз. Обязательной составляющей такого набора у словоформы является ее морфологическая репрезентация, обычно состоящая из значения части речи и граммем, а у знака препинания или сочинительного союза - тип оператора (значение, выполняемой им грамматической функции). Таким образом, можно представить линейную репрезентацию предложения в виде цепочки морфологических репрезентаций словоформ и типов операторов.

Процессом структурирования линейной репрезентации предложения называется построение ориентированного графа синтагм и ориентированного графа сегментов.

Синтагма определяет бинарное синтаксическое отношение вида R(A, B), где A и B - словоформы, а R - тип синтаксического отношения, который соответствует имени синтагмы; A является хозяином, B - слугой, т. е. A управляет B. Таким образом, узлами графа синтагм являются терминальные единицы. Связанность не является обязательным условием такого графа, так как синтагмы опираются только на морфологические репрезентации словоформы, линейный порядок предложения и, в некоторых случаях, на примитивную модель управления. На этом уровне анализа связи, для построения которых необходимо использовать сложную модель управления (предикатно-аргументную структуру) или семантическую информацию, могут не фиксироваться в графе синтагм.

Интуитивно сегмент можно определить как часть предложения (в частном случае целиком простое предложение), выделенную на письме знаками пунктуации и описывающую отдельную ситуацию; каждый такой сегмент имеет в качестве вершины явный предикат, выраженный в большинстве случаев финитной формой глагола, или «скрытый» предикат, который может быть выражен либо деепричастием, либо причастием, либо именем с семантической характеристикой действия; каждый такой предикат и задает ситуацию. Близкие по значению понятия в теоретической лингвистике - "предикация" и "элементарное предложение". В западной лингвистической традиции понятие сегмент эквивалентно термину клауза: "клаузой называется любая группа , в том числе и не предикативная, вершиной которой является глагол, а при отсутствии полнозначного глагола - связка или грамматический элемент, играющий роль связки" [Тестелец, 2001]. Например, любое придаточное

предложение (или причастный и деепричастный обороты) в составе сложного является сегментом, равно как и простое предложение в составе сложного образует отдельный сегмент. Сегмент, в терминах системы составляющих, является фразовой категорией (подобно NP, VP, PP, etc. [I. Sag, T. Wasow, 1999]) или нетерминальной единицей. Таким образом, узлами графа сегментов являются нетерминальные единицы.

Морфология, лексема, основа, окончание - понятия и термины, в последние годы ставшие общеупотребительными. Любой грамотный пользователь глобальной сети сможет "на пальцах" объяснить значение этих слов и преимущества поиска информации с использованием морфологии. На сегодняшний день только для русского языка существует несколько десятков известных систем морфологического анализа, число же программ английской морфологии в несколько раз больше. Следующим этапом в развитии направления искусственного интеллекта, занимающегося автоматической обработкой текста, является создание промышленной системы синтаксического анализа естественного языка.

Задача сегментации предложения является первой и, возможно, самой сложной компонентой полного синтаксического анализа. Целью сегментации является выделение и классификация сегментов в составе сложного предложения. Вторая компонента - построение внутрисегментных связей (графа синтагм) - исследована намного глубже и имеет успешные решения, экспериментально подтвержденные на анализе простых (односегментных) предложений. Основной упор в представляемой работе делается на разработку стратегий и методов автоматической системы сегментационного анализа предложения, хотя и предлагается ряд решений, связанных с внутрисегментным анализом терминальных единиц, а также методы моделирования морфологического анализа и снятия омонимии.

В последние десятилетия в странах Западной Европы, США и России проводятся чрезвычайно интересные и перспективные исследования по созданию систем автоматического синтаксического анализа для многих индоевропейских языков. Все попытки моделирования таких систем, как правило, происходят без предварительной сегментации предложения, что приводит к порождению в ходе анализа большого числа ложных синтаксических связей внутри сложного предложения и значительному

снижению скорости анализа. Отсутствие в моделях отдельного сегментационного компонента можно считать одной из основных причин того, что до сих пор не создано эффективных систем синтаксического анализа для русского языка (РЯ) [Т. Кобзарева и др., 2000]. Сегментационный компонент может быть использован и в качестве самостоятельной системы при решении многих прикладных задач автоматической обработки текстов (ИПС, автоматическое реферирование, машинный перевод, etc.). Сегментация предложения, наряду с морфологическим анализом, должна стать базисной составляющей любой полной системы АОТ. Таким образом, создание компонента сегментации русского предложения является чрезвычайно актуальной задачей.

Синтаксический анализ - задача приближения. Любая синтаксическая теория должна обладать описательной и объяснительной силой. Это утверждение с некоторыми оговорками и дополнениями остается справедливым и для прикладных моделей. Описательная сила модели формулируется как максимально возможное покрытие грамматических явлений рассматриваемого языка. Объяснение в теоретической лингвистике заключается в рассмотрении вопроса о существовании в языке именно данных наблюдаемых фактов, а не других [Я. Тестелец, 2001]. В данной работе объяснение понимается в контексте ИИ: любая интеллектуальная система должна уметь обосновать каждый шаг принятых ею в ходе анализа решений [M. Boden, 1990]. Такой критерий подразумевает, что количество эвристик и вероятностно-статистических распределений в системе синтаксического анализа должно быть сведено к минимуму. Существует и третий, не менее важный критерий прикладной модели - эмулирующий принцип построения алгоритмов, - который заключается в способности лингвистического процессора к воспроизведению интуиции и схемы рассуждений человека в процессе изучения и восприятия языка.

Идеальная модель лингвистического процессора состоит из четырех основных анализаторов: графематического (внешнее представление текста), морфологического, синтаксического и семантического. В данном случае мы ограничимся рассмотрением трехсоставного процессора без семантического анализатора.

Целью настоящей работы было создать экспериментальную систему автоматической сегментации русского предложения, демонстрирующую возможность эффективного – с точки зрения качества и скорости анализа – решения этой задачи как ключевого этапа полного автоматического синтаксического анализа русского текста. Основной решаемой проблемой была при этом разработка методов автоматической сегментации предложения и способов программирования, позволяющих минимальными силами решить поставленную задачу применительно к текстам произвольной (или почти произвольной) синтаксической сложности, а также построение прикладной модели лингвистического процессора, удовлетворяющего описательному, объяснительному и эмулирующему принципам.

Предметом исследования является структура сложного предложения русского языка и законы ее построения.

Работа построена на описании и сравнении решений и результатов двух систем синтаксического анализа, использующих компонент сегментации русского предложения.

Синтаксический процессор группы ДИАЛИНГ был создан в рамках проекта русско-английского машинного перевода (1999-2001). Фундаментом для исследований группы ДИАЛИНГ послужила система французско-русского автоматического перевода (ФРАП), разработанная в ВЦП совместно с МГПИИЯ им. М. Тореза в 1976-86 гг., и система анализа политических текстов (ПОЛИТЕКСТ), разработанная в Центре информационных исследований совместно с ВЦ ИСК РАН в 1991-97 гг [Н. Леонтьева, 1995].

Синтаксический анализатор научный группы Отделения интеллектуальных систем (ОИС) Института Лингвистики РГГУ (Д.Г. Лахути, Т.Ю. Кобзарева, И.М. Ножов) был создан в 1999-2003 гг. Предлагаемый проект продолжает развиваться и содержит наиболее полную реализацию идей сегментации русского предложения. Базисом для проводимых исследований послужила модель автоматического поверхностно-синтаксического анализа русского предложения, разработка которой была начата еще в 1971 г. в Информэлектро в секторе (затем отделе) Д.Г.Лахути группой лингвистов под руководством Г.А.Лесскиса .

Также в работе предложены альтернативные подходы к проектированию некоторых составляющих лингвистического процессора, разработанные

автором диссертации в НТЦ "Система" (1997-1998 гг.) и в исследовательском отделе компании Inxight, Software Inc. (2002-2003 гг.).

Методы исследования:

- Создание и пополнение лексиконов, содержащих необходимую для анализа морфологическую и грамматическую информацию;
- Разработка лингвистических стратегий и правил, отвечающих синтаксическим законам языка; изучение множества грамматических явлений, характерных для русского языка; поиск (с использованием конкорданса) случаев применения описываемых грамматических конструкций в корпусе текстов;
- Проектирование общей схемы лингвистического процессора и прикладной модели синтаксического анализа;
- Разработка алгоритмов порождения и перебора структурных вариантов предложения, связанных с явлением морфологической и синтаксической омонимии естественного языка;
- Создание динамических структур данных для представления и хранения синтаксической информации и программное моделирование процесса анализа на ЭВМ;
- Создание отладочного массива предложений, охватывающего все множество отраженных в модели грамматических явлений, и тестирование системы на пространстве реальных текстов.
- Оценка эффективности применения предложенных методов в системах АОТ или МТ.

Научная новизна работы состоит в том, что предложенные алгоритмы порождения структурных вариантов предложения позволили создать успешную модель лингвистического процессора и отказаться от декартова произведения омонимов, проверить работоспособность оригинальных грамматических стратегий анализа и реализовать методы автоматической сегментации без искусственного ограничения на перебор структурных вариантов, обусловленных морфологической и синтаксической омонимией, и без ограничения на глубину рекурсии сегментов и длину предложения.

Практическая значимость работы определяется программными реализациями анализаторов, созданных на базе разработанных методов и

стратегий и получивших практическое применение в различных системах автоматической обработки информации. В диссертации приведены примеры внедрения программ.

В процессе работы над диссертацией автором были получены следующие научные результаты:

1. Разработаны два метода автоматического синтаксического анализа предложения: метод активизации омонимов и рекурсивный метод монтажа разрывных сегментов.
2. Построена прикладная модель синтаксического анализатора, удовлетворяющего описательному, объяснительному и эмулирующему принципам, и позволяющая вести анализ параллельно: "снизу вверх" и "сверху вниз".
3. Отлажены грамматические стратегии сегментации и доказана их работоспособность.
4. Программно реализованы, совместно с другими разработчиками, две системы: промышленный синтаксический процессор группы "Диалинг" и экспериментальная сегментационный анализатор группы ОИС под руководством Д.Г. Лахути.
5. В процессе проводимых исследований и изучения существующих подходов к проектированию лингвистических процессоров автором, совместно с другими исполнителями, были разработаны и внедрены следующие прикладные модули: бессловарный морфологический анализ (НТЦ "Система") и Russian LinguistX Platform 3.5 (Inxight, Software Inc.), включающая в себя tokenizer, stemmer, tagger и np-grouper русского языка.

Апробация работы. Основные выводы и научные результаты диссертационной работы докладывались на международных конференциях Диалог в 2000 и 2001 гг., на национальных конференциях по искусственному интеллекту КИИ в 2000 и 2002 гг. и на научно-технической конференции ВИНИТИ в 2000 г. По теме диссертации автором опубликовано 6 работ. Сдана в печать одна статья.

Структура и объем работы: Диссертация состоит из введения, четырех глав, заключения, списка литературы из 53 наименований и двух приложений. Общий объем работы - 148 страниц, основной текст – 131 страница.

В первой главе приводятся аналогии с химическим строением сложного вещества, шахматной игрой и монтажом фильма, существенные для понимания изложенного в работе подхода к построению модели синтаксической сегментации; рассматриваются современные представления об искусственном интеллекте и его взаимосвязях с естественным языком в аналитической философии; вводятся определения лингвистических понятий релевантных для прикладных моделей; содержится изложение фундаментальных концепций синтаксической теории Head-driven Phrase Structure Grammar (HPSG) и описание ее приложений; рассматриваются синтаксические процессоры английского (LinkParser) и немецкого (STP) языков.

Во второй главе дается описание составляющих лингвистического процессора, которые предшествуют синтаксическому анализатору; рассматриваются различные решения и подходы к проектированию системы морфологического анализа, модуля снятия омонимии и задачи выделения из текста именных групп (NP).

В третьей главе дается описание синтаксического процессора ДИАЛИНГ: системы сегментационных и синтаксических правил, вершины сегментов и синтаксические группы, тезаурусы, элементарные аналитические формы и группы с разрывными союзами; содержится описание сегментационного анализатора группы ОИС: грамматические стратегии сегментации Т.Ю. Кобзаревой и модульность анализа, два типа омонимии (морфологическая и синтаксическая), граф синтагм и граф сегментов, общая схема и прикладная модель сегментации, рекурсивный метод монтажа разрывных сегментов и метод активизации омонимов; приводится сравнительный анализ двух систем.

В четвертой главе диссертации обсуждаются примеры использования и внедрения синтаксических процессоров ЕЯ и их составляющих: бессловарный морфологический анализ в системах автоматического построения словарей и поиска в правовой базе данных НТЦ "Система", технологии Inxight LinguistX Platform в системах АОТ (Murax, Categorizer и Smart Discovery), синтаксический процессор в системе машинного переводчика ДИАЛИНГ, экспериментальные и обучающие возможности сегментационного анализатора группы ОИС.

Создание сегментационного анализатора группы ОИС стало возможным в первую очередь благодаря лингвистико-алгоритмическому аппарату,

разработанному Т.Ю. Кобзаревой, и руководителю проекта д.т.н., профессору Д.Г. Лахути.

Разработка синтаксического процессора группы ДИАЛИНГ - результат коллективного творчества. В разное время в проекте принимали участие следующие специалисты:

1. А. Сокирко (руководитель проекта);
2. Д. Панкратов (русский синтаксис и сегментация, программная реализация);
3. Л. Гершензон (система синтаксических и сегментационных правил);
4. Т. Кобзарева (русский синтаксис и сегментация);
5. И. Ножов (русский синтаксис и сегментация, программная реализация).

Всем участникам проекта ДИАЛИНГ автор выражает свою благодарность.

За техническую поддержку в реализации проекта бессловарного морфологического анализа автор благодарит А.Н. Кудрина (руководителя отдела разработки НТЦ "Система").

Также автор выражает благодарность исследователям компании Inxight, Software Inc. за оказанную техническую поддержку, научные консультации и обсуждения, проводившиеся при создании русской версии LinguistX Platform (tokenizer, stemmer, tagger и np-grouper):

1. Masayo Iida (руководитель отдела лингвистических исследований Inxight, Санта Клара, Калифорния, США);
2. David van den Akker (руководитель департамента разработки Inxight, Антверпен, Бельгия);
3. Carolina Rubio de Hita (ведущий специалист Inxight, Антверпен, Бельгия).

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ И ПРИКЛАДНЫЕ СИСТЕМЫ

I. Синтаксические аналогии

В современной теоретической лингвистике часто используются аналогии, связанные с другими научными дисциплинами и областями человеческого знания, которые помогают наглядно представить и продемонстрировать структурные задачи и подходы к моделированию языковых процессов.

Так, для задачи реконструкции праязыков в компаративистике распространено сопоставление понятия "генетического дрейфа" в биологии и законов распределения фонетических соответствий в языках. Самая популярная и распространенная аналогия в синтаксических теориях связана с химией: строение молекулы и явление изомерии [И. Мельчук, 1999].

В этом разделе будут приведены три аналогии, которые могут быть полезны для понимания задачи сегментации сложного предложения, - химическое строение сложного вещества, шахматная игра и монтаж фильма.

Следуя аналогии Мельчука, попытаемся представить "объемное" предложение с включенными в него придаточными молекулой сложного вещества в органике, состоящей из атомов двух и более видов, где каждый отдельный сегмент играет роль такого атома. Сегмент, в свою очередь, также состоит из конечного множества иерархически организованных элементов, т.е. имеет свою внутреннюю независимую от общей структуру. Как соединения атомов в молекуле образует разные вещества, так и по-разному связанные сегменты образуют сложные предложения, отличающиеся по смыслу. Рассмотрим для наглядности следующий пример: 'Иван, который оставался в городе, сказал, что видел Петра'. Это сложное предложение состоит из трех разнородных простых сегментов: (1) 'Иван сказал', (2) 'который оставался в городе' и (3) 'что видел Петра'. Соединение сегментов '2←1→3' задает смысл приведенного примера, в то время как тип соединения '1→3→2' соответствует предложению с другим общим смыслом: ' Иван сказал, что видел Петра, который оставался в городе', а тип соединения '2→3→1' порождает бессмыслицу. Разумеется, разные типы соединения обусловлены не только

внешними условиями, но и составляющими внутри каждого сегмента, равно как и устойчивые связи между атомами в молекуле зависят не только от физических условий, но и от самих химических элементов. Конечно, аналогия с химическим строением сложного вещества весьма субъективна, но позволяет продемонстрировать тот факт, что в предложении существует некоторая макроструктура, живущая по своим законам и отличающаяся от принятой (состоящей из слов).

Первым ученым, который заметил аналогию между шахматной партией и системой языка, был швейцарский лингвист Фердинанд де Соссюр. Для него шахматы служили удачной метафорой для противопоставления диахронии и синхронии в языке: каждое передвижение фигуры в течение партии изменяет позицию и дальнейшее развитие на доске, причем последствия одного хода могут быть незначительными, а могут иметь необратимые последствия; передвижение фигур во время игры аналогично языковым изменениям в диахронии, а каждая позиция на доске между ходами игроков сравнима с синхронным срезом языка во времени [Ф. де Соссюр, 1999]. Существуют другие, придуманные после Соссюра и не менее интересные шахматные аналогии для естественного языка. Тот факт, что на одной клетке шахматной доски ни в какой момент игры не могут одновременно стоять две фигуры сравнивается с гипотезой единственности заполнения грамматической позиции в предложении, когда, например, не может быть двух подлежащих или двух сказуемых в одном простом предложении [Я. Тестелец, 2001]. Но для представления процесса сегментации нас будет интересовать совсем другое свойство шахматной игры, а точнее способность шахматиста. Способность шахматиста заключается в его интуиции, которая позволяет даже человеку с минимальным опытом игры выбирать фокусное пространство на шахматной доске, т.е. из миллиарда возможных ходов и комбинаций в каждой позиции безошибочно выбирать десяток единственно правильных и осмысленных, не просчитывая остальные. Таких фокусных пространств или ключевых узлов в шахматной позиции может быть несколько, и человек сосредоточивается на выборе одной, самой выгодной на его взгляд, комбинации из десятка возможных, пытаясь просчитать изменение позиции на несколько шагов вперед и предсказать ответы противника. Выбор такого фокусного пространства стал ключевой задачей для программистов и специалистов в области ИИ,

создававших шахматные программы. Оперативной памяти даже самых мощных на сегодняшний день компьютеров не хватает, чтобы просчитать все комбинаторно возможные комбинации на несколько шагов вперед и выбрать из них лучшую. Задача эмуляции такой способности шахматиста была решена через обучающие алгоритмы, но даже сейчас можно легко "повесить" среднюю шахматную программу, сделав в начале партии непредсказуемый, бессмысленный ход, не заложенный в память (схему) программы, который заставит ее потерять фокусное пространство и приступить к перебору всего множества вариантов. Способность шахматиста и существование фокусного пространства на шахматной доске подводит нас к гипотезе, что процесс восприятия и понимания человеком сложного предложения состоит отнюдь не в попытке построения всех связей между словами в этом предложении. Очевидно, что человек запоминает и строит только самые необходимые, базовые связи, вырывая нужные куски-ситуации и забывая остальное, а потом, при необходимости, достраивая и предсказывая подробности. Возможно, поэтому человек всегда пересказывает полученную им информацию "своими словами", добавляя иногда не существующие в реальности подробности. Такой принцип выбора фокусного пространства, безусловно, связан и с явлением языковой избыточности, позволяющей человеку вычислять смысл незнакомого слова в тексте из контекста. Теперь попробуем применить этот принцип к построению сегментов. Рассмотрим следующий пример: 'Девочка, решив уже, когда ее позвали, задачу, засмеялась'. Чтобы собрать сегменты: (1) ' девочка засмеялась', (2) 'решив уже задачу' и (3) 'когда ее позвали', - абсолютно необязательно знать, что 'решив' управляет 'задачу' или 'позвали' управляет 'ее', или 'девочка' зависит от 'засмеялась'. Достаточно знать структурные законы и последовательность действий, которые позволят собрать разрозненные блоки-отрезки ('решив уже' и 'задачу') в один сегмент ('решив уже задачу'), и структурные ограничения, которые не позволят объединить два не относящихся друг к другу отрезка ('девочка' и 'задачу' или 'когда ее позвали' и 'засмеялась') в одно целое ('девочка задачу' или 'когда ее позвали засмеялась'). Таким образом, подобная шахматная аналогия подводит нас к гипотезе о том, что, имея правильную грамматическую стратегию, анализ существующей макроструктуры предложения и сборку разрывных сегментов возможно осуществить без предварительного построения большинства синтаксических связей между словами, т.е. провести анализ

"сверху вниз". Важен и другой вывод: такой способ восприятия предложения является более естественным для человека.

Фильм состоит из эпизодов, а каждый эпизод есть последовательность кадров, выбранная и определенная режиссером из порой огромного объема отснятого материала. Такая смонтированная последовательность кадров должна наилучшим образом выражать все грани и оттенки смысла, чувств и переживаний, столь неуловимых категорий, которыми оперирует художник, и которые он пытается донести до зрителя в этом эпизоде. Основная цель монтажа попытаться расположить кадры фильма в таком линейном порядке, чтобы их конечная непрерывная цепочка возымела максимальное воздействие на сознание зрителя. Сергей Эйзенштейн считается родоначальником русской школы кинематографии и одним из первых теоретиков монтажа. Эйзенштейн определяет монтаж как столкновение и конфликт кусков [С. Эйзенштейн, 2000], оппонируя Пудовкину, для которого монтаж - последовательное сцепление кадров, отражающее логическую последовательность действия в эпизоде. Хороший монтаж различает чередование общего и крупного планов, где в батальных сценах можно противопоставить трагедию общественную (поле боя) и трагедию личную (лицо солдата), или ряд сделанных в эпизод вставок кадров из других временных и пространственных сцен и т.д. Именно такого рода чередования и вставки определяют Эйзенштейновские принципы столкновения и конфликта, делают монтаж интеллектуальным. Техника монтажа, опирающаяся на эти два принципа, повышает экспрессивную (выразительную) силу кинематографа (особенно, это было актуально для немого кино). Эйзенштейн напрочь отметает возможность сцепления - "кирпичики", рядами излагающие мысль. Многие лингвисты склонны объяснять избыточность грамматики естественного языка выразительной силой последнего. Одну и ту же мысль одними и теми же словами можно выразить в пределах одного предложения множеством разных синтаксических конструкций и грамматических построений. Зачем понадобилось в приведенном выше примере ("Девочка, решив уже, когда ее позвали, задачу, засмеялась.") разрывать два цельных сегмента, и почему грамматика русского языка допускает подобные построения, когда ту же "историю" можно было разложить на три простых предложения ("Девочка уже решила задачу. Ее позвали. В этот момент она засмеялась.") или расположить последовательно ситуации ("Решив уже задачу,

девочка засмеялась, когда ее позвали"). Синтаксис языка позволяет вкладывать ситуации друг в друга и чередовать их, разбивая тем самым цельные фрагменты и создавая "матрешечную" структуру предложения. Свойство рекурсивности сегмента, т.е. возможность включать в себя теоретически бесконечное число других сегментов, наглядно демонстрирует экспрессивную (выразительную) силу языка. Необходимо признать и то, что оригинальное построение примера отличается от его двух вариантов неуловимым оттенком или "интонацией" смысла. Человек оперирует сегментами: разбивает, склеивает, чередует их, - выбирая, в конечном счете, оптимальную форму для выражения своей мысли. С точки зрения человека работа автоматического сегментационного анализатора - "демонтирование" составленной им формы изложения, с точки зрения программы процесс сегментации - монтаж логической последовательности «кадров».

Абстрактная структура, в первую очередь, является хорошим инструментом для представления полученных знаний об объекте и удобным средством для описания рассматриваемого объекта, но и сам по себе результат процесса структурирования обладает несколькими важными свойствами [D.Grune, C.Jacobs, 1990]: абстрактная форма позволяет (а) увидеть наиболее общие закономерности, скрытые в линейной репрезентации объекта, (б) продолжить анализ на основе накопленных знаний в структуре, (в) распознать ошибки, допущенные в строении объекта. Все три свойства верны и для синтаксической структуры предложения, которая позволяет (а) увидеть возможные пути трансформации или оценить проективность предложения, (б) перейти на следующий уровень первичного семантического анализа [А. Сокирко, 2001], (в) проверить грамматическую правильность анализируемого предложения. Но синтаксическая структура обладает еще одним уникальным свойством, она физически материализует то, что раньше считалось прерогативой гуманитарного знания, литературоведения или пространных рассуждений эстетов, - это стиль автора. Достаточно беглого взгляда, чтобы увидеть стилистические различия В. Набокова и А. Чехова, представленные структурами фрагментов их произведений на рис. 1 и рис. 2. Обе структуры построены машиной - автоматическим синтаксическим процессором группы ДИАЛИНГ.

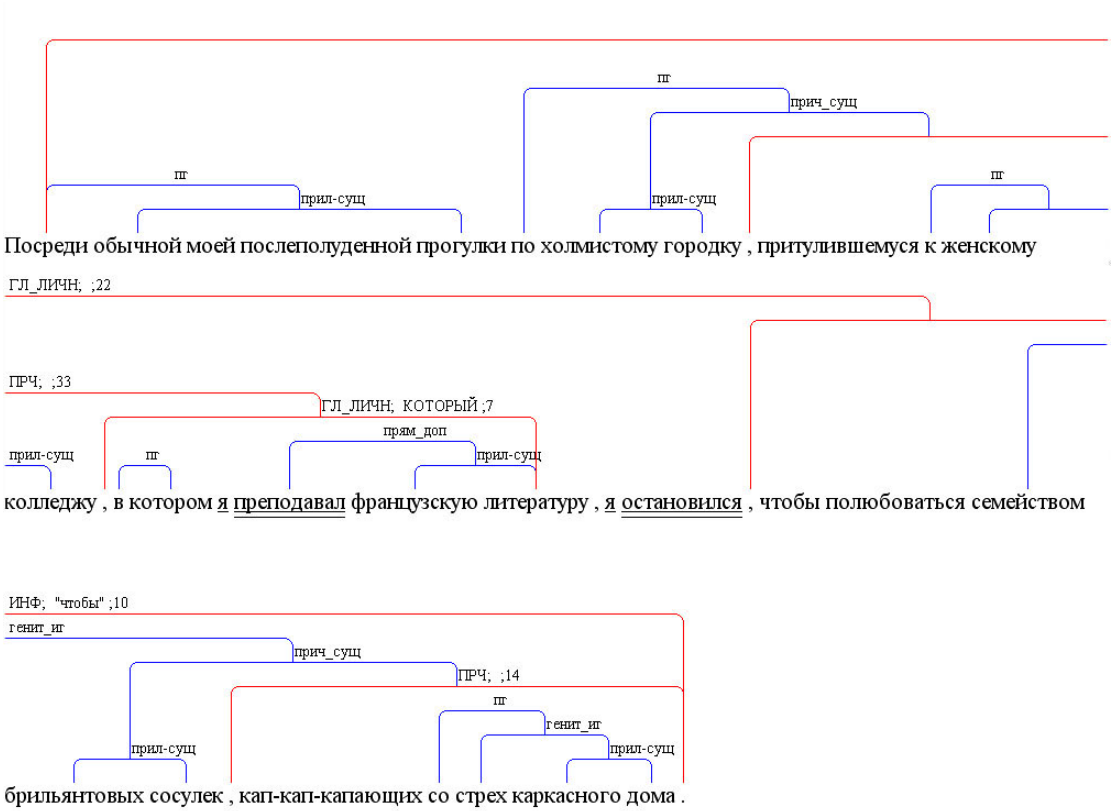


Рис.1 В. Набокова «Сестры Вэйн»

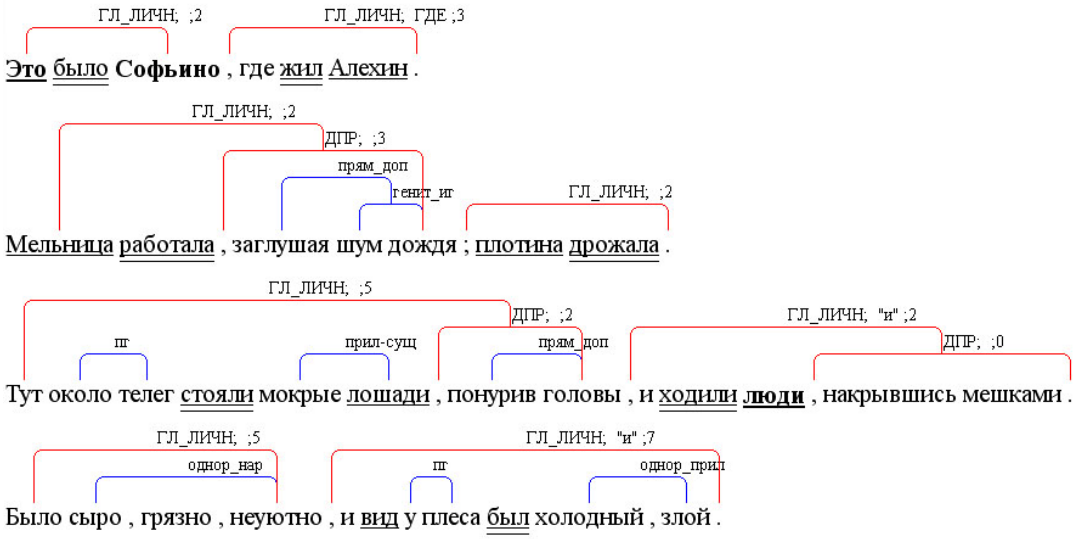


Рис.2 А. Чехова «Крыжовник»

II. Фундамент синтаксического анализа.

В этом разделе обсуждаются основные грамматические средства и понятия (явления), которыми оперирует процессор в ходе автоматического синтаксического анализа. Рассматриваемые явления могут быть как внутренними, относящимися к терминальным единицам и связям между ними, так и внешними, т.е. универсальными структурными законами. Существующий набор явлений в современной теоретической лингвистике намного шире, но, к сожалению, далеко не все из них хорошо формализуемы и могут быть использованы в прикладных моделях. Определения некоторых лингвистических понятий изменены и формулируются только в контексте прикладных моделей, коими являются синтаксический процессор Диалинг и сегментационный анализатор группы ОИС под руководством Д.Г. Лахути.

Все языковые средства, которыми располагает система для определения синтаксических понятий, являются либо свойствами самого объекта, т.е. предложения естественного языка, либо свойствами его элементов, т.е. словоформ и знаков пунктуации (операторов). Синтаксические понятия, по существу, представляют собой функции, где параметрами служат языковые средства, а сами функции используются в условиях грамматических стратегий или правил. Ниже приведены пять языковых средств синтаксического анализа:

1. *Словоизменительные морфологические средства.* Для языков с развитой морфологией, каким является русский, - это основной способ материализации синтаксических связей. Словоформа w_1 морфологически зависит от словоформы w_2 по морфологической категории C, если граммема (значение грамматической категории) g категории C, характеризующей w_1 , выбирается в зависимости от некоторого свойства f словоформы w_2. Словоформа w_2 называется контролером морфологической зависимости, а w_1 - ее мишенью [Я. Тестелец, 2001]. Другими словами, один элемент предложения подстраивается под другой, т.е. принимает грамматическую форму продиктованную вторым элементом. Показателем морфологической зависимости в русском служит флексия, т.к. граммемы в русском обычно приписаны флексии, что позволяет в некоторых случаях обнаружить зависимость между двумя словоформами, отсутствующими в словаре, (например, "глок-ая куздр-а"). Если категория C, по которой наблюдается

морфологическая зависимость, выражается в вершине, налицо вершинное маркирование, если же эта категория выражается в зависимой словоформе - зависимостное маркирование [Я. Тестелец, 2001]. В русском языке граммемы многих форм омонимичны ('ночи' = [[рд., дт., пр., ед.], [им., вн., мн.]] - омонимия числа и падежа), что создает определенные трудности в процессе анализа. Неоднозначность граммем в ходе автоматического синтаксического анализа иногда приводит к возникновению синтаксической омонимии и построению альтернативного синтаксического варианта (графа синтагм). Падежная омонимия с номинативом часто приводит к неоднозначному определению правой границы сегмента и, как следствие, к построению альтернативной структуры сегментации (графа сегментов). Парадокс или скорее взаимовлияние двух уровней анализа морфологического и синтаксического состоит в том, что граммема, являясь эффективным средством поиска морфологической зависимости, которая служит одним из способов реализации синтаксического отношения, может быть однозначно проинтерпретирована только вследствие фиксации этого отношения.

2. *Селективные признаки:* Классифицирующие (селективные) признаки приписываются лексемам в грамматическом словаре, в отличие от граммем, которые вычисляются, исходя из парадигматического класса, для каждой словоформы на этапе морфологического анализа. Наиболее важной для синтаксиса является классификация лексем по категориальным (частеречным) признакам: существительное, глагол, прилагательное, etc. Существует и более дробное деление на субкатегориальные признаки внутри частей речи, так существительные можно разбить на два класса: одушевленные и неодушевленные. Категориальные признаки задают потенциальных участников синтаксической связи и определяют в большинстве случаев вершину в структуре, а также предопределяют понятия управления и согласования. Одушевленность и неодушевленность в русском языке служит контролером согласования для некоторых форм мужского рода или во множественном числе - 'вижу большого [мр., ед., вн.] кролика [мр., ед., вн.] (*большой [мр., ед., вн.] кролика [мр., ед., вн.])' ~ 'вижу большой [мр., ед., вн.] стол [мр., ед., вн.] (*большого [мр., ед., вн.] стол [мр.,

ед., вн.])' или 'вижу четкие фотомодели' ~ 'вижу красивых фотомоделей' (пример Е. Ножовой).

3. *Служебные слова*: предлоги, союзы и союзные слова, вспомогательные компоненты аналитических форм, частицы и т.д. Средства, которые служат в качестве опорных точек анализа. Так, союз может быть использован для определения поверхностного типа сегмента, или вспомогательный компонент аналитической формы содержит недостающие предикату граммемы, или предлог оформляет актант глагола.
4. *Знаки препинания (операторы)*: запятая, тире, точка, вопросительный знак, etc. Это средство не выделяется в теоретических описаниях, так как теоретический синтаксис имеет дело больше с устным языком, чем с письменным, к тому же не все письменные языки, в отличие от русского, имеют жесткие правила расстановки знаков препинания. В первую очередь, операторы определяют границы как сегментов, так и всего предложения. Тире является выражением эллиптированного элемента предложения и часто используется в стратегиях поиска неморфологического предиката. Анализ бифункциональности оператора (когда, например, оператор является одновременно и правой границей сегмента, и оператором сочинения слов) - одна из самых трудных задач сегментации, которая и задает рекурсивный характер как грамматических стратегий анализа, так и методов программной реализации. В теоретических работах принято выделять интонацию как средство синтаксического анализа. Действительно, операторы в письменном тексте являются частичным выражением подмножества синтаксических случаев, характеризующихся интонацией в устном языке. В примере А. Кибрика предложение "В этой гимназии учился впоследствии всемирно известный киноартист", произнесенное с падением интонации на 'впоследствии' имеет синтаксическую связь 'учился → впоследствии', а при отсутствии падения тона - 'известный → впоследствии' [А. Кибрик, 2001]. Такие случаи применения интонации для различения синтаксических связей не фиксируются операторами в письменной форме, поэтому идеальный синтаксический процессор должен решить эту проблему через понятие синтаксической омонимии, построив две равноправных синтаксических структуры предложения.

5. *Порядок слов:* Линейное расположение слов в предложении играет особую роль в изолирующих языках (китайский) и является основным средством для выражения синтаксических отношений в этих языках. Наряду с селективными признаками порядок слов имеет доминирующее значение в проектировании синтаксических анализаторов языков с бедной морфологией (английский). Во многих системах английского синтаксиса порядок слов задает направление поиска хозяина или слуги для каждого класса лексем и типа связи [D. Sleator, D. Temperley, 1991]. Для русского языка это средство анализа имеет второстепенное значение, хотя и применяется в ряде случаев для установление синтаксических связей или оценки омонимичных структур предложения. Несмотря на свободный порядок слов в русском, некоторые синтаксические зависимости имеют обязательным критерием выделения жесткий линейный порядок: генитивное определение должно следовать за определяемым словом ('ножка стол-а', 'сын отц-а'); предлог предшествует существительному ('на стол-е', 'у отц-а'); в 90% случаев определение, выраженное прилагательным или местоименным прилагательным, стоит до существительного (['большой красивый стол', 'седой отец'] ~ ['впечатление необычное']). Порой статистическое расположение синтаксических вершин и их зависимых позволяет разделить все типы синтаксических отношений на три типа: левоветвящиеся (прилагательное существительное: 90%), правоветвящиеся (генитивное определение: 100%) и смешанные (слабые актанты глагола: 50%/50%). Подобные эмпирические распределения могут эффективно использоваться в прикладных моделях. В лингвистической типологии эмпирически установлена универсальная классификация языков мира: языки левого (японский) и правого ветвления (русский и английский). Правда, эта классификация, в основном, строится на статистическом распределении фразовых категорий в линейном порядке предложения, к которым относятся именные (NP), предложные группы (PP) и клаузы (некоторые виды сегментов: придаточные определительные, причастные обороты, etc.). Другая синтаксическая классификация оперирует линейным порядком основных членов предложения: подлежащее (subject), сказуемое (verb) и дополнение (object). Английский относится к языкам Subject Verb Object (SVO) порядка, для русского SVO порядок является статистически доминирующим и наиболее естественным, но грамматически не

единственно возможным. В английском предложении ‘The farmer kills the duckling’ ‘Фермер убивает утенка’ (пример Э. Сепира [Э. Сепир, 1993]) любое изменение порядка слов ведет к изменению смысла всего высказывания ('The duckling kills the farmer' 'Утенок убил фермера.') или к потере грамматической правильности ('* The farmer the duckling kills.'Фермер утенка убил.'), то в русском переводном эквиваленте (‘Фермер убивает утенка’) возможен 3! перестановок, сохраняющих как общий смысл высказывания, так и грамматическую правильность, т.е. в русском варианте данного предложения возможны любые комбинаторные порядки: SVO, SOV, OVS, etc. Таким образом, линейный порядок предложения в автоматическом синтаксическом анализе используется как указатель наиболее вероятного направления поиска слуги или хозяина, и только в редких случаях как обязательный критерий установления синтаксической зависимости.

Ниже будут определены понятия, оперирующие изложенными языковыми средствами, только в рамках их приложения в синтаксическом анализаторе:

1. Согласованием называется пересечение векторов граммем двух словоформ, где ожидаемый результат пересечения определяется категориальными признаками словоформ. Согласование может быть полным или частичным.

Полное согласование:

(а) $V_A \cap V_N$ = [c, Sg, g] || [c, Pl], где V_A - вектор граммем полного прилагательного, причастия или местоименного прилагательного; V_N - вектор граммем существительного; $c \in C$ = [им., рд., вн., дт., тв., пр.] - значение падежа; Sg (ед. ч.) и Pl (мн. ч.) - значения грамматического числа; $g \in G$ = [мр., жр., ср.] - значение грамматического рода.

(б) $V_{Snom} \cap V_P$ = [p≠∅, n] || [g] || [p=∅, Pl], где V_{Snom} - вектор граммем подлежащего, выраженного существительным или местоимением в именительном падеже; V_P - вектор граммем сказуемого, выраженного финитной формой глагола или краткой формой прилагательного или причастия; $p \in P$ = [∅, 1л., 2л., 3л.] - значение грамматического лица; $n \in N$ = [Sg, Pl].

Частичное согласование:

(а) $V_A \cap V_N$ = [c], такой тип согласования используется в дуальных конструкциях (например, "красные стол и стул" или "синий и красный мячи"), в тех случаях

когда еще не построены сочинительные группы. Применение частичного согласования в этих конструкциях зависит полностью от грамматического описания, принятого в прикладной модели. Альтернативный вариант анализа дуальных конструкций состоит в предварительном поиске сочинительных групп, вычисления граммем группы и сведения проверки согласования при последующем установлении атрибутивной связи (именной группы) к полному согласованию типа (а).

(б) $V_{A1} \cap V_{A2} = [c]$, $V_{N1} \cap V_{N2} = [c]$, $V_{P1} \cap V_{P2} = [p \neq \varnothing, n] \parallel [Imptv, n] \parallel [Inf] \parallel [g] \parallel [p=\varnothing, Pl]$, где Imptv - императив, Inf - инфинитив. Подобного рода согласование используется для определения сочинительных конструкций в русском языке.

2. Примитивной моделью управления называется вектор M, определенный в словаре для каждой лексемы L, способной управлять словоформой X. Вектор M лексемы L содержит значения селективных признаков и/или граммемы словоформы X. Вектор $M \subset M| =$ [предлог, подчинительный союз, инфинитив, им., рд., вн., дт., тв., пр.]. Управлением называется пресечение вектора M лексемы L с вектором граммем словоформы X или с значением селективных признаков словоформы X. Явление примыкания и конгруэнтности, а также более сложные случаи управления, не используются в предлагаемых моделях синтаксических анализаторов и считаются прерогативой этапа первичного семантического анализа [А. Сокирко, 2001].
3. Грамматические понятия, построенные на объединении значений селективных признаков в более крупные единицы, используются в синтаксических моделях. Предикат в предложении может быть выражен словоформой с значением части речи $ps \in PS =$ [финитная ф. гл., кр. прил., кр. прич., предикатив]. При построении атрибутивной связи AN A может быть выражено словоформой с значением части речи $a \in A =$ [полное прилагательное, полное причастие, местоименное прилагаельное], а N может быть выражено словоформой с значением части речи $n \in N =$ [существительное, местоимение, субстантивированное прилагательное].

В синтаксических анализаторах изложенные выше понятия обычно оформляются в виде программных функций, которые служат для проверки и установления возможного синтаксического отношения. Таким образом,

изложенные понятия объединяются в более крупных модулях анализа, каковыми являются грамматические правила и стратегии:

1. Каждое грамматическое правило устанавливает один тип синтаксического отношения R(A, B) между двумя единицами анализа и однозначно задает вершину. Число используемых типов отношений, а также их названия, зависит от прикладной модели и конкретной системы, набор универсальных синтаксических отношений для русского языка приведен во многих теоретических работах [Я. Тестелец, 2001]: отпредложное (предлог и управляемое им существительное), определительное (существительное и его согласованное определение), посессивное (существительное и его несогласованное опрделение), субъектное (сказуемое и подлежащее), etc. В роли единиц анализа, на месте A и B, где A - вершина, а B - зависимое, могут выступать как отдельные словоформы, так и целые группы (фразовые составляющие); заполнение A и B во многом зависит от синтаксического аппарата, принятого в анализаторе для описания структуры. Идеальное грамматическое правило в автоматическом синтаксическом анализе характеризуется следующими критериями: (а) описывает только один тип синтаксического отношения; (б) однонаправленность анализа, т.е. зависимое B может находиться только слева или только справа от вершины A; (в) не содержит рекурсивных вызовов или вызовов других правил; (г) обрабатывает только контактно расположенные единицы анализа; (д) результат не зависит от порядка применения правил. Использование грамматических правил задает прозрачность архитектуры процессора и обеспечивает устойчивость системы к изменениям. Перечисленные критерии не являются строгими, но приближают правило к его идеальной форме.
2. Грамматические стратегии, наравне с правилами, используются во всех системах автоматического синтаксического анализа. Типичным примером компонента процессора, построенного на стратегии, является анализ сочинения. Сложность анализа сочинительных конструкций состоит в том, что в процессе построения связи одновременно могут рассматриваться больше чем две единицы анализа; учитываются операторы (знаки препинания и сочинительные союзы) внутри конструкции; нарушается древесность графа, т.к. каждый элемент множества узлов, образующих

сочинительную связь, попарно связан со всеми остальными элементами множества и одновременно является как слугой, так и хозяином узлов, принадлежащих множеству сочинения. Грамматическое сочинение проецируется на все уровни анализа и типы синтаксических единиц, терминальные и нетерминальные: сочинительная конструкция может состоять из теоретически неограниченного числа сочиненных словоформ или именных групп, или предложных групп, или отдельных сегментов (сочиненные придаточные внутри сложного предложения или причастные обороты и т.д.). Стратегии позволяют эффективно организовывать процесс сегментационного анализа. Грамматическая стратегия в прикладных моделях характеризуется следующими критериями: (а) двунаправленность анализа, т.е. зависимое В может находиться как слева, так и справа от вершины А; (б) учитывает единицы, стоящие между потенциальным зависимым и хозяином, в процессе анализа; (в) позволяет строить связи между разрывными составляющими; (г) ищет варианты синтаксической связи для анализируемой единицы, принимая во внимание возможность синтаксической омонимии; (д) может содержать рекурсивные вызовы, оперировать грамматическими правилами и использовать другие стратегии в качестве подпрограмм; (е) оперирует общими структурными ограничениями. Стратегии представляют определенную сложность для программной реализации и гораздо более чувствительны к изменениям в системе, чем правила, но использование стратегий повышает точность анализа, обеспечивает модульность системы и позволяет проектировать сложные схемы взаимодействия компонент модели (см. гл. 3).

Перечислим общие структурные ограничения в прикладных моделях анализа:

1. Проективность. А. Е. Кибрик: Линейная структура предложения проективна, если между каждой парой слов, связанных подчинительной связью, находятся только слова, зависящие (непосредственно или опосредованно) от одного из этих слов [А. Кибрик, 2001]. Я. Г. Тестелец: Предложение называется проективным, если, при том, что все стрелки зависимостей проведены по одну сторону от прямой, на которой записано предложение: (а) ни одна из стрелок не пересекает никакую другую стрелку (принцип непересечения стрелок); (б) никакая стрелка не накрывает корневой узел (принцип необрамления стрелок) [Я. Тестелец, 2001]. Предложения, в

которых нарушается принцип необрамления стрелок, называются слабо проективными, но являются грамматически допустимыми. В реальных системах ограничение на проективность служит для проверки грамматической правильности построенных подструктур в предложении, при этом используется только принцип непересечения стрелок и, как правило, для именных групп (определительная связь) и предложных групп (отпредложная связь), т.к. уже на уровне глагольных групп ограничение на проективность не является строгим и может нарушаться в ряде случаев (в устной речи, художественной литературе или бюрократически-деловых текстах). Структура сегментов предложения является строго проективной, и этот принцип кладется в основу сегментационного анализа (подробнее см. гл. 3). На рис. 1 приведен пример проективной структуры именной группы с несогласованным определением, на рис. 2 показана непроективная и грамматически недопустимая структура именной группы.

Рис. 1

Рис. 2

2. Любая синтаксическая единица (терминальная или нетерминальная) в структуре предложения может непосредственно зависеть только от одной вершины, кроме случая сочинения. В сочинительных конструкциях вершина, входящих в нее единиц, не определена, хотя в некоторых моделях такой вершиной объявляется сочинительный союз, что является формальным допущением, сохраняющим единообразность структурного представления.
3. Простой сегмент предложения содержит только один субъект (подлежащее), кроме случая сочинения субъектов.
4. Простой сегмент предложения содержит только один предикат (сказуемое), кроме случая сочинения предикатов.

Общие структурные ограничения применяются как в ходе синтаксического анализа, так и на этапе оценки равноправных синтаксических представлений, полученных как следствие морфологической или синтаксической омонимии.

III. Гипотеза глубины.

Еще в 1961 году американским ученым В. Ингве была выдвинута гипотеза глубины [В. Ингве, 1965] для синтаксиса естественного языка. Ингве опирался в своих исследованиях на работы в области психологии, где доказывалась, что человек имеет ограничение на объем непосредственной памяти равное семи единицам. Так, человек в среднем способен запомнить с первого раза и правильно воспроизвести около семи десятичных цифр или несвязанных между собой слов.

Ингве использовал в своей работе порождающие грамматики Хомского и компьютерную модель синтеза английского предложения для демонстрации принципа глубины. Линейная последовательность терминальных единиц (с естественным для английского порядком слов - слева направо) порождаемого предложения появляется итерационно, т.е. путем пошагового применения формальных правил разворачивается сверху вниз структура составляющих. При этом, на каждом шаге расширения структура раздваивается на левую составляющую, к которой применяется правило на следующем шаге работы программы, и на правую, хранящуюся в оперативной памяти машины до тех пор, пока левая ветвь синтезируемой структуры не получит интерпретацию на уровне терминальных единиц. Таким образом, чем больше глубина вложения стоящих слева от вершины зависимых, тем больше число промежуточных единиц, хранящихся в оперативной памяти и ожидающих своей интерпретации. На примере английской глагольной группы p_1(p_2(p_3(p_4(very clearly) projected)) pictures) appeared) показана глубина вложения первого элемента ‘very’ в линейной последовательности словосочетания, т.е. в ходе синтеза такого словосочетания, в момент появления терминального узла ‘very’ в порождаемой структуре, в памяти машины будет храниться 4 нетерминальных символа (V, N, A, Adv), соответствующих возможной порождающей грамматике такой группы: VP -> NP + V; NP -> AP + N; AP -> AdvP + A; AdvP -> SecAdv + Adv; V -> ‘appeared’; N -> ‘pictures’; A -> ‘projected’; Adv -> ‘clearly’; SecAdv -> ‘very’.

Анализируя данный пример, Ингве приходит к выводу, что система частей речи обеспечивает способ автоматического подсчета шагов вниз по ветви левостороннего вложения составляющих и прекращения расширения конструкции прежде, чем она пересечет придел глубины. Вершина может быть расширена влево и сентенциальным дополнением, например в английском придаточным предложением с союзом ‘that’: That is true is obvious (“то, что это справедливо, очевидно”). Подобные структуры с расширением вершины влево с n-глубиной вложений называются регрессивными. Регрессивные структуры требуют, чтобы запоминался дополнительный нетерминальный символ для каждого шага развертки вниз. Человек, воспринимающий предложения с регрессивной структурой, также вынужден запоминать слова или группы слов, расположенные до их смысловой вершины. Прогрессивная структура с n-глубиной вложений последовательно расширяется вправо от вершины, сохраняя в оперативной памяти на каждом шаге только один нетерминальный символ. Прогрессивная структура не ограничена объемом памяти, и следовательно может расширяться бесконечно. Примером такой прогрессии служит английское предложение с придаточным ‘that’ в постпозиции к глаголу s(John said cl_1(that Paul said cl_2(that Bill said…))). Синтаксическая регрессия, в отличие от прогрессии, имеет ограничение на глубину вложения, обусловленное объемом непосредственной памяти человека, где максимальное значение n статистически равно приблизительно пяти составляющим. Ингве утверждает, что грамматика естественного языка располагает механизмами для ограничения глубины регрессивных структур, так в английском невозможно вложение придаточных-подлежащих с союзом ‘that’ (That is true is obvious ~ *That that it is true is obvious isn’t clear ~ It isn’t clear that it’s obvious that it’s true), т.е. грамматика языка строится таким образом, что в ней исключаются слишком глубокие конструкции, а взамен их вводятся конструкции меньшей глубины. Ингве отмечает внутреннюю асимметричность структуры английского языка, вызванную накладываемыми ограничениями на возможности ветвления влево, по сравнению с ветвлением направо.

Самым ярким примером регрессивной структуры в русском языке можно считать частотное явление сегментных “матрешек” [Т. Кобзарева, 2002], а именно свойство рекурсивности сегмента. Возвращаясь к примеру,

приведенному в разделе «Синтаксические аналогии», s(девочка, cl_1(решив уже, cl_2(когда ее позвали), задачу), засмеялась) не трудно заметить регрессивный характер этого предложения, но, в отличие от приведенных в работе Ингве английских конструкций, данная русская регрессивная структура не имеет грамматически мотивированного ограничения на глубину и может расширяться, путем вложения новых сегментов, теоретически бесконечно, формально не нарушая грамматической правильности предложения: s(девочка, cl_1(решив уже, cl_2(когда, cl_3(чтобы продолжить, cl_4(…), начатый разговор), ее позвали), задачу), засмеялась). Единственным и естественным ограничением сегментной матрешки ($cl_1(cl_2(\ldots(cl_n())\ldots))$) служит компетенция носителя языка: объем непосредственной памяти, который не позволяет человеку воспринимать предложения с вложенными друг в друга сегментами, где превышается некоторая допустимая глубина. Любопытно совпадение, что в первых компиляторах языка C++ существовало ограничение на глубину вложения шаблонов (templates) [Б. Страуструп, 1999], которое программист вынужден был соблюдать при написании программ.

Впоследствии гипотеза глубины послужила основанием для создания (уже в терминах вершинных грамматик) универсальной типологической классификации, разделившей языки мира на языки левого и правого ветвлений. В терминологии Ингве, языки левого ветвления (аварский, японский) тяготеют к регрессивной структуре предложения, а языки с правым (английский, русский) – к прогрессивной структуре. Нельзя полностью принять утверждение ученого о том, что каждый язык располагает способами ограничения регрессии и способами, дающими возможность обойти ограничения объема памяти. Существование языков левого ветвления и явление сегментной матрешки в русском демонстрируют факультативность таких грамматических механизмов ограничения глубины (хотя попытки решения загадки лево- и правоветвящихся языков предпринимались Дж. Хокинсом через понятие, определенное им, как сфера идентификации составляющих (constituent recognition domain) [Я. Тестелец, 2001]). Ингве приходит к неоспоримому выводу, что глубина есть фактор, влияющий на развитие языка.

Существует три единственно возможных линейных расположения зависящего сегмента от сегмента-хозяина:

1. сегмент-слуга стоит слева от сегмента-хозяина (инверсное распределение);

2. сегмент-слуга стоит справа от сегмента-хозяина (последовательное распределение);
3. зависимый сегмент разрывает главный сегмент, то есть зависимый сегмент вложен в подчиняющий сегмент (гнездование).

При этом, первый и третий типы расположения задают регрессивную структуру предложения, а второй – прогрессивную.

Стоит сделать предположение, что возможно провести аналогичную левому и правому ветвлениям классификацию естественных языков по приведенным выше трем типам распределения сегментов в предложении с использованием информации о грамматических классах сегментов (см. гл. 3). Также интересна оценка грамматически допустимой глубины гнездования в разных языках.

Гипотеза глубины и свойство рекурсивности сегмента являются базисом для понимания структуры сложного предложения и для разработки подхода к задаче автоматической сегментации.

IV. **Head-driven Phrase Structure Grammar (HPSG).**

> В лингвистике гораздо легче создать свою новую теорию, чем разобраться в уже существующей чужой…

В начале 90-х годов в американкой математической лингвистике активно разрабатывался новый класс грамматик для анализа синтаксической структуры естественного языка, основанный на лексическом подходе и критике контекстно-свободных грамматик (CFG). Одним из первых вариантов такой теории явилась грамматика GPSG (Generalized Phrase Structure Grammar), разработанная еще в конце 80-х гг., которая достаточно быстро эволюционировала в грамматику управляемых вершинами фразовых категорий - Head-driven Phrase Structure Grammar (HPSG). Главными идеологами HPSG стали ученые Стэндфордского Университета И. Саг и Т. Васоу, создавшие компьютерную лабораторию для экспериментальных исследований прикладных возможностей HPSG. Данный класс грамматик отличает два тезиса:

- Построение иерархической структуры свойств (feature structure) каждой лексической единицы языка, содержащей грамматическую и семантическую информацию, и проектирование лексикона с иерархической организацией

типов свойств, где каждый тип-потомок может наследовать и переопределять свойства предка (такая система организации лексикона во многом следует объектно-ориентированной модели программирования).

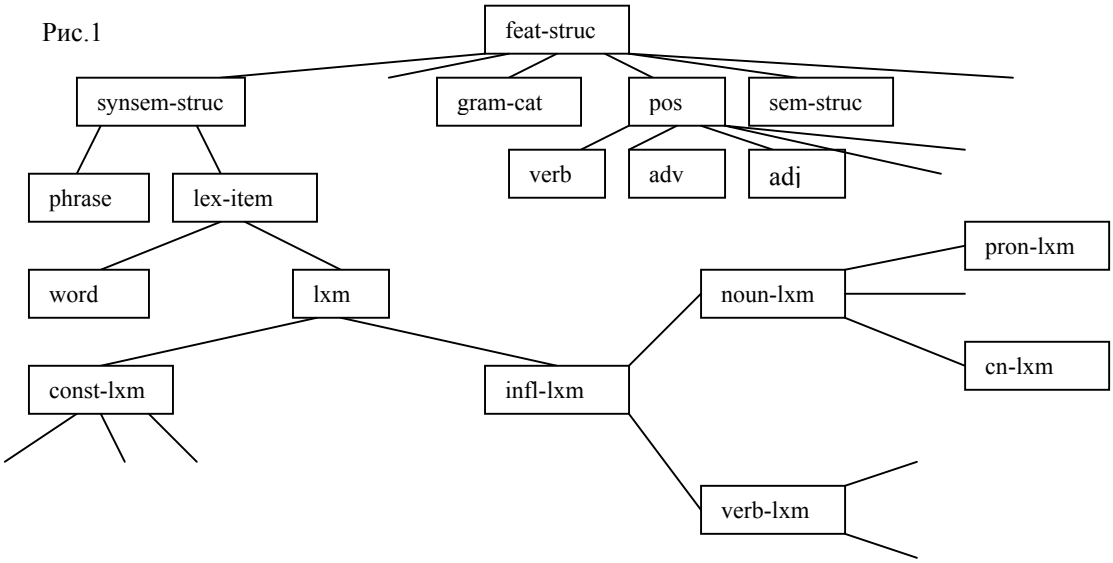
- Унификация – как базовый механизм построения синтаксической структуры.

Многие теоретические постулаты HPSG заимствованы из теории принципов и параметров, позднего варианта порождающей грамматики (ПГ) Хомского, а именно, из ее базового модуля X'-теории. Сторонниками ПГ в X'-теории признается необходимость определения вершины структуры фразовых категорий и производится отказ от базового компонента, т.е. формальных правил генерации предложения [Я. Тестелец, 2001]. В X'-теории доминирующим становится лексический подход (через словарь) к построению синтаксической структуры. Грамматика HPSG не использует понятие проекции составляющей и сохраняет базовый компонент, в качестве дополнительного инструмента механизма унификации, сводя количество правил грамматики к минимуму и делая их максимально общими. Сохраняя правила, HPSG, в отличие от X'-теории, теряет универсальность грамматики, но приобретает практическую значимость для программных реализаций: разработанный механизм унификации позволяет проектировать эффективные прикладные системы синтаксического анализа.

Критика CFG сторонниками лексикализма состоит в том, что (а) контекстно-свободные грамматики произвольны (отсутствие у фразовых категорий вершины и ее свойств); (б) CFG избыточны (простейший случай, когда возникает избыточность, отсутствие возможности проверки согласования).

На рис.1 приведен фрагмент иерархии типов для английского языка, принятый в структуре лексикона HPSG, и таблица свойств/ограничений по умолчанию, присвоенных каждому типу.

Рис.1



Фрагмент таблицы общих типов от базового предка feat-struc (структура свойств) к потомку cn-lxm (нарицательные существительные):

Тип	**Свойства/ Ограничения**	**Комментарии**
feat-struc (структура свойств)		базовый абстрактный тип
Synsem-struc (синтактико-семантическая структура)	[SYN gram-cat; SEM sem-struc]	SYN – свойство, описывающее грамматический компонент лексемы, которое задается структурами свойств, определенных в типе gram-cat; SEM - свойство, описывающее семантический компонент лексемы, которое задается структурами свойств, определенных в типе sem-struc.
gram-cat (грамматическая категория)	[HEAD pos; COMPS list(synsem-struc); SPR list(synsem-struc)]	HEAD – структура свойств вершины; COMPS – список возможных комплементов, заданных значениями структурного типа synsem-struc; SPR - список возможных спецификаторов, заданных значениями типа synsem-struc.
lex-item (лексическая единица)	[ARG-ST list(synsem-struc)]	ARG-ST – аргументная (актантная) структура (argument structure), заданная списком synsem-struc.
lxm (лексема)	[SEM [MODE / none]]	MODE – модальность, одно из “подсвойств” свойства SEM, принимает по умолчанию пустое значение и может быть

		переопределено в значении типа-потомка.
infl-lxm		Абстрактный тип
noun-lxm	[SYN [HEAD [noun: AGR [PER / 3rd]; ANA / -]]; ARG-ST / <>; SEM [MODE / ref]]	Свойство HEAD состоит из значения типа noun (существительное): AGR (согласование) имеет в своем составе свойство PER (лицо) со значением по умолчанию 3, ANA (анафор) с отрицательным значением по умолчанию (которое может быть переопределено в типе-потомке для референциальных местоимений); ARG-ST по умолчанию задается пустым списком. MODE в SEM присваивается значение по умолчанию ‘референция’.
cn-lxm (common noun lexeme)	[SYN [HEAD [AGR (1)]; SPR <[]>]; ARG-ST / <[DetP; AGR (1)]>]	Значение согласования AGR сворачивается до идентификатора (1), SPR – свойство вершины присоединять спецификатор, заданный по умолчанию списком, состоящим из одного элемента; ARG-ST состоит из одного элемента, выраженного детерминатором и ограничением AGR, которое должно совпадать по идентификатору с аналогичным свойством вершины.

Пример словарного входа лексемы ‘book’:

<book, [cn-lxm: ARG-ST <[COUNT +]>]; SEM […]>, где положительное значение свойства COUNT (исчисляемость) – ограничение на значение аргумента. Здесь мы опускаем значение семантического компонента, т.к. нас интересует, в первую очередь, устройство синтаксической структуры (семантический компонент HPSG описывает ситуацию, используя смысловые отношения, и позволяет вычислять смысл всего предложения путем конкатенации значений семантических свойств его составляющих).

Применив принцип наследования от типа-предка для лексемы ‘book’, мы получим полную структуру свойств:

<book, [cn-lxm: SYN [HEAD [noun: AGR (1); ANA / -; SPR<[]>]]; ARG-ST <[DetP; AGR (1); COUNT +]>]; SEM [MODE / ref; …]>

В лексиконе HPSG существует множество лексических правил (Lexical Rules), позволяющих построить словоформу и ее свойства от данной лексемы. Простейшими примерами таких правил могут служить правило для

единственного числа существительного (Singular Noun Lexical Rule): <(1), [noun-lxm]> => <(1), [word; SYN [HEAD [AGR [NUM sg]]]]> или правило множественного числа (Plural Noun Lexical Rule): <(1), [noun-lxm, AGR-ST <[COUNT +]>]> => <F_{NPL}(1), [word; SYN [HEAD [AGR [NUM pl]]]]>, где F_{NPL} – морфологическая функция, присоединяющая флексию для формы множественного числа: (book) => F_{NPL}(book) = books.

Основной недостаток такого лексикона для прикладных моделей анализа – трудоемкость разработки. Очевидно, что для русского языка число типов и лексических правил сильно возрастет. Отсутствие разделения анализа на уровни и словари (морфологический, синтаксический и семантический) лишает архитектуру лексикона прозрачности. Для языков с развитой морфологией намного эффективней задавать ограничения (constraints) по согласованию процедурно. Алгоритмический подход к синтаксическому анализу позволяет сводить к минимуму использование статических данных, тогда как лексикализм и успешность работы грамматик, построенных на унификации, целиком зависят от полноты лексикона.

Унификацией называется наиболее общий метод, позволяющий двум совместимым дескрипциям структуры свойств соединять информацию, которую они содержат, в одну (обычно большую) дескрипцию. Две дескрипции являются совместимыми в том случае, если они не содержат в своих структурах конфликтующих типов или разных атомарных значений одних и тех же свойств. Если дескрипция D_1 определена множеством структур свойств σ_1 и D_2 определена множеством σ_2, тогда унификация D_1 и D_2 определена пересечением σ_1 и σ_2. Допустим, существует частное грамматическое правило для построения фразовой структуры с учетом согласования, типа [phrase: POS (1); NUM (2)] → [word: POS (1); NUM (2)] + NP, где POS – свойство селективного признака и NUM – свойство категории числа. Тогда два вхождения идентификатора (1) и два вхождения идентификатора (2) означают, что значение свойства POS и значение свойства NUM материнского узла в левой части правила и соответствующие значения первого дочернего узла в правой части правила должны быть унифицированы. В лексиконе HPSG разные структуры свойств могут быть вложены одна в другую, что позволяет создавать сложные иерархические структуры, а значение селективного признака определяется

лексическим типом (pos → verb, adj, noun, etc. см. рис.1). Таким образом, свойство вершины HEAD для лексической единицы можно определять через тип, соответствующий значению селективного признака, где каждому такому типу приписано свойство согласования AGR: например, [HEAD [noun: AGR [PER 3rd; NUM pl]]], свойство HEAD является сложной иерархической структурой. В этом случае можно утверждать, что два элемента согласованы, если унифицированы спецификации их свойств AGR. Теперь можно переписать приведенное выше частное правило в более общем виде: [phrase] → H[word] + NP, где 'H' маркирует вершинный дочерний узел, который содержит идентифицируемую с материнским узлом структуру свойств HEAD.

В HPSG вводится два универсальных синтаксических принципа:

- Принцип вершины HFP (Head Feature Principle)

  Для любой фразовой категории, где определена вершина, значение свойства HEAD материнского узла и значение свойства HEAD дочернего узла должны быть унифицированы.
- Принцип модели управления (The Valence Principle)

  Значения свойств SPR (спецификатор) и COMPS (комплементы) материнского узла идентичны значениям аналогичных свойств вершинного дочернего узла.

Аналогичным образом метод унификации используется и при построении семантической структуры (свойство SEM), для этого в грамматике определяются дополнительные принципы.

Базовый компонент грамматики HPSG в упрощенном виде состоит из четырех максимально общих синтаксических правил [I. Sag, T. Wasow, 1999]:

1. Правило комплемента вершины (Head-Complement Rule)

   [phrase: COMPS <>] → H[word: COMPS <(1),…,(n)>] (1) … (n) , где n – идентификатор комплемента.

   Фразовая категория может состоять из лексической вершины и следующих за ней комплементов; в частном случае список комплементов пуст.
2. Правило спецификатора вершины (Head-Specifier Rule)

   [phrase: SPR <>] → (1) H[phrase: SPR <(1)>]

   Фразовая категория может состоять из фразовой вершины и предшествующего ей спецификатора.

3. Правило модификатора вершины (Head-Modifier Rule)

   [phrase] → H(1)[phrase] [phrase: MOD (1)]

   Фразовая категория может состоять из фразовой вершины и следующего за ней совместимого фразового модификатора.

4. Правило сочинения (Coordination Rule)

   [SYN (0); IND s_0] → [SYN (0); IND s_1] … [SYN (0); IND s_{n-1}] [HEAD conj; IND s_0] [SYN (0); IND s_n], где семантическое свойство IND - индекс некоторой ситуации.

   Любое число вхождений элементов с одинаковой синтаксической структурой (свойство SYN) могут быть соединены в один сочинительный элемент той же структуры.

Приведенный базовый компонент грамматических правил обладает тремя недостатками: (а) жесткий линейный порядок составляющих в правой части правила, что не позволяет использовать такого рода правила в языках с относительно свободным порядком синтаксических составляющих, каким является русский (то же относится и к структурным свойствам лексикона HPSG, где строго определен порядок следования комплементов лексемы, так [COMPS <NP, PP>] означает, что в линейной цепочке предложения именная группа, управляемая данной лексемой, должна стоять перед предложной); (б) правила не способны анализировать слабо проективные структуры, грамматически допустимые во многих языках; (в) абсолютная зависимость синтаксических правил от правильности и полноты структур свойств отдельно взятого словарного входа лексикона.

Рассмотрим пример синтаксического анализа грамматикой HPSG английского предложения ‘They sent us a letter’ (‘Они послали нам письмо’) без учета семантической структуры:

<they, [word: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]>	<sent, [word: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS < NP_j[CASE acc], NP_k[CASE acc]>]]>	<us, [word: SYN [HEAD [noun: CASE acc; AGR [PER 1rd; NUM pl]]; SPR <>; COMPS <>]]>	<a, [word: SYN [HEAD [det: COUNT +; AGR [3sing]]]]>	<letter, [word: SYN [HEAD [noun: AGR [3sing; GEND neut]]; SPR <D[AGR [3sing; GEND neut]; COUNT +]>; COMPS <(PP)>]]> *круглые скобки комплимента означают его факультативность,*

				в данном случае предложной группы PP
<they, [word: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]>	<sent, [word: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS < NP_j[CASE acc], NP_k[CASE acc]>]]>	<us, [word: SYN [HEAD [noun: CASE acc; AGR [PER 1rd; NUM pl]]; SPR <>; COMPS <>]]>	<a, [word: SYN [HEAD [det: COUNT +; AGR [3sing]]]]>	<NP, [phrase: SYN [HEAD [noun: AGR (1)]; SPR <D[AGR (1); COUNT +]>; COMPS <>]]> *обнуляется список комплементов в соответствии с правилом комплемента вершины и идентифицируются значения свойств HEAD и SPR в соответствии с HFP*

<they, [word: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]>	<sent, [word: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS < NP_j[CASE acc], NP_k[CASE acc]>]]>	<NP, [phrase: SYN [HEAD [noun: CASE acc; AGR [PER 1rd; NUM pl]]; SPR <>; COMPS <>]]> *правило комплемента вершины и HFP*	<NP, [phrase: SYN [HEAD [noun: AGR [3sing; GEND neut]]; SPR <>; COMPS <>]]> *унификация произошла в соответствии с правилом спецификатора вершины и отвечает принципу HFP*

<NP, [phrase: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]> *правило комплемента вершины и HFP*	<VP, [phrase: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS <>]]> *принцип модели управления, правило комплемента вершины и HFP*
<VP, [phrase: SYN [HEAD [verb]; SPR <>; COMPS <>]]> *принцип модели управления, правило спецификатора вершины и HFP*	

Порядок применения синтаксических правил в HPSG – свободный.

Успешными унификациями называется цепочка унификаций, которая приводит к построению связного синтаксического дерева, т.е. структура предложения сворачивается до уровня одной вершины с единой структурой свойств. Одними из факторов, влияющих на количество ложных унификаций в ходе анализа, являются факультативные (слабые) комплементы и морфологическая омонимия, опущенная в рассмотренном выше примере (‘letter’

имеет значение как существительного, так и глагола в английском). Так же очевидно, что в языке со свободным порядком составляющих, с высоким коэффициентом глубины вложения и возможностью прерывания составляющих, число ложных унификаций сильно увеличится, а значит, и уменьшится скорость анализа.

На основе грамматики HPSG в Стэндфордской лаборатории создается система автоматического синтаксического анализа английского предложения, программная реализация процессора осуществляется на функциональном языке программирования LISP [S. Oepen, J. Caroll, 2000]. Пока что объем лексикона и скорость процессора не позволяют проводить анализ сложных предложений. Для отладки работы и развития анализатора используется приложение тестового обеспечения для естественно-языковых процессов TSNLP (Test Suites for Natural Language Processing), которое содержит базу данных тестовых примеров и результатов анализа [S. Oepen, K. Netter, 1997]. Синтаксический процессор HPSG контролируется системой, осуществляющей наблюдение (profiling) и оценку скорости работы отдельных функций анализатора, что позволяет вести протокол эволюции программной модели с учетом вносимых в нее изменений. Такой инструмент profiling имеет около сотни параметров оценки (таких как число ложных и успешных унификаций, время работы процессора ЭВМ для отдельно взятой операции, объем выделенной динамической памяти, etc.) и дает возможность выявлять критические зоны для скорости работы синтаксического анализа [S. Oepen, J. Caroll, 2000].

Несмотря на указанные недостатки подхода лексикализма и недостатки базового компонента унифицирующей грамматики, необходимо признать большой экспериментальный потенциал построенной на HPSG модели для исследователей в области ИИ.

V. Link Grammar Parser (LinkParser).

Противоположный HPSG подход к синтаксическому анализу английского языка был разработан группой американских исследователей (Daniel Sleator, Davy Temperley и др.) в самом начале 90-х гг., этот проект получил название Грамматика Соединений (Link Grammar). Базовое отличие Link Grammar состоит в том, что используемая модель анализа является

контекстно-свободной грамматикой CFG [D. Sleator, D. Temperley, 1991]. В отличие от HPSG, абстрактной и универсальной синтаксической теории ЕЯ, Link Grammar с самого начала создавалась как аппарат для автоматической системы анализа предложения, что позволило авторам отойти от академических представлений, принятых в лингвистической традиции.

Каждая единица словаря грамматики описывается формулой, состоящей из соединителей (коннекторов connector). Коннектор состоит из имени типа связи (например, S – субъект, O – объект, CL – сегмент и т.д.), в которую может вступать рассматриваемая единица анализа, и суффикса, определяющего вектор направления соединения (‘+’ право-направленный коннектор и ‘-’ лево-направленный коннектор). Лево-напрпавленный и право-направленный коннекторы одного типа образуют связь (соединение link). Так, два слова W_1 и W_2, имеющие словарные входы W_1: A^- и W_2: A^+, образуют соединение A в линейной последовательности W_2W_1, но не связаны в цепочке W_1W_2.

Язык формул, оперирующий коннекторами, использует четыре связки: оператор конъюнкции &, оператор дизъюнкции or, фигурные скобки {} для обозначения факультативности и неограниченность повторения @ (эквивалент оператора + Клини). Так, в формуле слова W: D^- & {@A^-} выражение ‘@A^-’ означает, что должна быть реализована связь с дескриптором A слева от W хотя бы один раз, и может повторяться неограниченное число раз; выражение ‘{@A^-}’ означает, что связь A факультативна. Конъюнкция несимметрична для однонаправленных коннекторов и задает строгий порядок слов в предложении: в формуле W: A^+ & B^+ слово, реализующее соединение A, должно находится ближе к W в линейной последовательности предложения, чем слово, реализующее соединение B, в той же последовательности. Для разнонаправленных коннекторов конъюнкция симметрична: формулы W: A^- & B^+ и W: B^+ & A^- эквивалентны.

Проблема избыточности словаря решается в системе LinkParser путем разбиения слов английского языка на 23 класса, где каждому такому классу приписывается своя формула. Разумеется, существует слова и подмножества слов-исключений, которые получают отдельную от основных классов формульную интерпретацию (к ним относятся, например, описание модальных глаголов или референциальных местоимений). Слова обобщаются в классы по селективным и субкатегориальным признакам. В ходе анализа словам в системе

приписываются значения их базовых классов – селективных признаков (‘cat.n ran.v’).

Тип коннектора задается именем, где начальные заглавные буквы являются базовым дескриптором, а нижний составной индекс, как правило, задает значение граммемы, что позволяет косвенно проверять согласование или необходимое управление при установлении связи (например, ‘S^{+}’ – существительное, ‘dogs ideas: Sp^{+}’ – существительное во множественном числе, ‘dog idea: Ss^{+}’ - существительное в единственном числе). Таким образом, могут соединяться либо равные коннекторы, либо два коннектора, один из которых выше уровнем: ‘Spa^{+}’ может соединяться с ‘S^{-}’, ‘Sp^{-}’ или ‘Spa^{-}’, но не с ‘Ss^{-}’ или ‘Spb^{-}’. В анализаторе LinkParser используется около ста различных коннекторов, различающихся преимущественно нижнем индексом, число базовых дескрипторов - сравнительно небольшое.

В LinkParser вводятся общие структурные ограничения:

- Проективность: связи между словами в предложении не пересекаются.
- Полнота связей: все слова в линейной последовательности должны быть соединены между собой.
- Порядок: в линейной цепочке предложения должен выполняться порядок реализаций соединений, заданный в формуле несимметричной конъюнкцией для однонаправленных коннекторов.
- Исключение: для одной пары слов не может быть проведено больше одной связи.

Рассмотрим пример анализа простого предложения ‘The cat chased a snake’ (‘Кошка преследовала змею’).

Фрагмент словаря:

Словоформа	Формула
the a	D^{+}
cat snake	D^{-} & (O^{-} or S^{+})
Chased	S^{-} & O^{+}

Результат анализа:

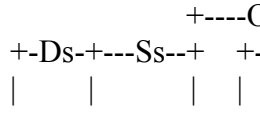


рис. 1

Нетрадиционность модели Link Grammar состоит и в том, что разработчики отказались от системы составляющих, столь популярной для

представления синтаксической структуры английского языка, и используют формализм, идеологически близкий к теории зависимостей, описанной в работах И. Мельчука. В отличие от деревьев зависимостей, бинарные связи, строящиеся LinkParser, не содержат вершины и не имеют направления. Используя информацию о селективных дескрипторах, приписанную терминальным единицам предложения, и тип коннекторов, маркирующих соединения, можно транслировать построенную LinkParser проективную структуру (linkage) в классическое дерево зависимостей, такая же трансляция возможна, рассматривая вложения соединений, и в систему непосредственных составляющих, определенных в выходной структуре анализатора.

Чтобы получить для каждого слова множество его однозначных интерпретаций (т.е. последовательностей лево-направленных и право-направленных коннекторов), формула, приписанная каждому слову в предложении, приводится к ее дизъюнктивной форме. Дизъюнктивной формой называется конечное множество дизъюнктов формулы. Дизъюнкт имеет вид $((L_1, L_2, \ldots, L_m) (R_n, R_{n-1}, \ldots, R_1))$, где $L_1, L_2, \ldots, L_m$ лево-направленные коннекторы, а $R_1, R_2, \ldots, R_n$ право-направленные. В стандартной форме дизъюнкт можно представить в виде формулы, использующей только оператор конъюнкции: $(L_1 \& L_2 \& \ldots \& L_m \& R_1 \& R_2 \& \ldots \& R_n)$. Тогда формулу

$(A^- \text{ or } ()) \& D^- \& (B^+ \text{ or } ()) \& (O^- \text{ or } S^+)$

можно представить в дизъюнктивной форме как множество из восьми дизъюнктов:

((A, D) (S, B))
((A, D, O) (B))
((A, D) (S))
((A, D, O) ())
((D) (S, B))
((D, O) (B))
((D) (S))
((D, O) ())

Дизъюнкты используются в алгоритме автоматического синтаксического анализа на основе Link Grammar.

Морфологическая омонимия задается при анализе дизъюнкцией формул двух омонимов (например, терминальной единице ‘letter’ в предложении соответствуют два словарных входа ‘letter.n: f_1’ и ‘letter.v: f_2’, тогда ‘letter: f_1 or

f_2’). Синтаксическая омонимия – множество всех построенных структур одного предложения, отвечающих выше перечисленным структурным ограничениям.

Для решения задачи сегментации в грамматику LinkParser введены специальные типы коннекторов: TH, CL, CO, X, B, etc. Глаголы, способные в качестве дополнения присоединять сегмент (clause), содержат в своей формуле TH^+ (придаточное с союзом ‘that’) или CL^+. Глагол с коннектором TH^+ образует связь TH с подчинительным союзом ‘that’, а союз ‘that’ с субъектом придаточного сегмента. В тех случаях, когда придаточный сегмент занимает позицию перед главным, используется $\{CO^-\}$ для субъекта главного сегмента и CO^+ для придаточного союза. Для определения правой границы вложенного сегмента служит факультативный коннектор $\{Xc^+\}$ в формуле подчинительного союза и Xc^- у запятой (‘,:Xc^-‘). На рис.2 продемонстрирован результат анализа сложного предложения, содержащего придаточный сегмент с союзом ‘that’ и предложный сегмент, где ‘after: (CL^+ or J^+) & ($\{Xc^+\}$ & CO^+)’ выступает одновременно в роли союза и предлога:

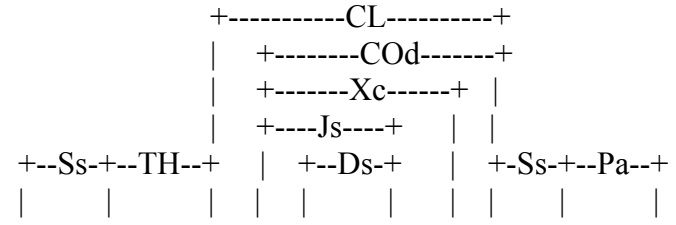


рис. 2

Для вложенных относительных придаточных, где возможна ситуация опущения союза (‘who’, ‘which’, etc.), используется коннектор $\{B^+\}$ для существительных (потенциальный субъект главного сегмента) и B^- в формуле транзитивных (переходных) глаголов (потенциальный предикат относительного придаточного). На рис. 3 показан результат анализа вложенного в главный сегмент относительного придаточного:

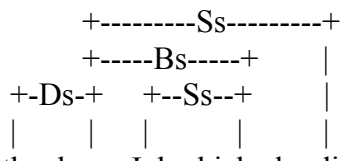


рис. 3

В системе LinkParser существует постпроцессор, предназначенный для работы с уже построенными альтернативными структурами предложения. Основная концепция постпроцессора заключается в разделении структуры на домены (domains) по одному или нескольким определенным типам связи.

Доменами (областями определения) называются полученные в результате деления независимые фрагменты предложения. Принципы деления на домены, как правило, определяются для каждого отдельного типа связи. В большинстве случаев используются сегментные связи (CL, CO, B, etc.) для нахождения доменов. Так, в предложении ‘John thinks there might be a problem’ выделяется два домена, соответствующие делению сложного предложения на простые сегменты: ‘John thinks’ и ‘there might be a problem’. Группой называется все множество связей определенных в пределах одного домена. На группах определены множества правил типа: группа, которой принадлежит связь X, должна содержать либо связь Y, либо Z. Основная цель такого постпроцессора - создать дополнительные ограничения (фильтр, реализующийся в системе как правила группы домена) на уже построенные синтаксические варианты, отвечающие общим структурным ограничениям.

Алгоритм синтаксического анализа в процессоре LinkParser основан на методе динамического программирования [D. Grinberg, J. Lafferty, 1995], т.е. в ходе анализа предложения все множество синтаксических единиц, входящих в предложение S, разбивается на перекрывающиеся подмножества (подзадачи) с сохранением исходного линейного порядка, где каждое такое подмножество является (в случае успешного построения связей между его элементами) поддеревом полного графа S и называется частичным решением (partial solution). Пусть S состоит из конечного множества упорядоченных словоформ $S = [W_1, W_2, \ldots, W_n]$, тогда процедура синтаксического анализа P порождает для S некоторое первоначальное множество M пар регионов (regions), где $M = [([W_1 .. W_2] [W_2 .. W_n]), ([W_1 .. W_3] [W_3 .. W_n]), \ldots, ([W_1 .. W_i] [W_i .. W_n]), \ldots, ([W_1 .. W_{n-1}] [W_{n-1} .. W_n])]$. Регионом называется отрезок предложения $S' = [W_i .. W_j]$, где i и j – границы региона, а все словоформы, входящие в отрезок, включая и границы, - единицы региона. Рекурсивный вызов процедуры P для региона S’ порождает новое множество $M' = [([W_i .. W_{i+1}] [W_{i+1} .. W_j]), ([W_i .. W_{i+2}] [W_{i+2} .. W_j]), \ldots, ([W_i .. W_{j-1}] [W_{j-1} .. W_j])]$ и т.д. Рекурсивный вызов процедуры P для региона S’ определен только в том случае, если между W_i и W_j удалось установить связь X, т.е. дизъюнкт W_i содержит коннектор X^+, а дизъюнкт W_j X^-. Очевидно, что процедура P в процессе анализа может неоднократно вызываться для одного и то же региона R, для того, чтобы избежать повторного вычисления R, используется прием memoization[Т. Кормен и др., 2001, стр. 298]: когда в

процессе выполнения алгоритма регион встречается в первый раз, его решение заносится в хеш-таблицу, и в дальнейшем решение для этого региона берется непосредственно из таблицы. Каждая словоформа состоит из множества дизъюнктов $W_j = [D_1, D_2, \ldots, D_m]$, поэтому полное определение возможного региона выглядит как $S' = [W_{ik} .. W_{jl}]$, где i и j – индексы словоформ, а k и l – индексы их дизъюнктов. Так, в упрощенном виде, выглядит основной алгоритм построения структуры в LinkParser, благодаря ограничению на проективность, являющийся модификацией алгоритма оптимальной триангуляции выпуклого многоугольника [Т. Кормен и др., 2001, стр. 306]. Скорость рекурсивного алгоритма синтаксического анализа экспоненциально зависит от количества слов в предложении, но применение memoization позволяет решить задачу построения синтаксической структуры в Link Parser за время[2] $O(n^3)$, где n – количество слов в предложении. Число различных подзадач (как и в задаче оптимальной триангуляции многоугольника [Т. Кормен и др., 2001, стр. 309]) в основном процессе вычисления синтаксической структуры составляет $\Theta(n^2)$. На скорость выполнения алгоритма сильно влияет общее число дизъюнктов в последовательности словоформ. Например, из формулы существительного ‘time’ порождается 770 дизъюнктов.

Для ускорения работы алгоритма синтаксического анализа в LinkParser предложен ряд решений, в том числе и эмпирических. Перед началом анализа устанавливается фильтр, удаляющий все дизъюнкты, содержащие «непарные» коннекторы: если для некоторого коннектора X^- дизъюнкта D, принадлежащего словоформе W, слева в линейной последовательности S не найдено X^+, то D будет удален, аналогично для право-направленного коннектора X^+. Другой метод ускорения вводит эмпирическое ограничение на длину возможного соединения в зависимости от типа связи. Несмотря на применяемые методы оптимизации, тестирование системы показывает, что в большинстве случаев анализ сложных предложений, длина которых превышает 25-30 слов, приводит к комбинаторному взрыву, и результатом работы анализатора становится “панический” граф, как правило, случайный вариант синтаксической структуры, зачастую несвязанной.

[2] Уточнение относительно времени работы основного алгоритма Link Parser было внесено Сергеем Протасовым.

К сожалению, использование грамматики LinkParser для русского языка представляется невозможным по ряду причин:

- Основная идея грамматики - использование лево- и право-ветвящихся коннекторов – теряет свою силу для языка с относительно свободным направлением связей (особенно для глагольных групп).
- Если предположить, что каждое возможное направление связи можно маркировать отдельным типом коннектора, то в этом случае резко возрастет как число базовых коннекторов, так и число дизъюнктов словоформ, что негативно скажется на скорости работы процессора.
- Избыточность и значительно возрастающий объем словаря, которые возникают в силу морфологической развитости флективного языка: каждая морфологическая форма описывается отдельной формулой, где нижний индекс входящего в нее коннектора должен будет обеспечить процедуру согласования, что приведет к усложнению составления коннекторов и к увеличению их общего числа в грамматике.

Тем не менее, LinkParser по праву считается одним из самых элегантных и детально проработанных решений задачи синтаксического анализа английского языка, а лингвистическая прозрачность грамматики и программная реализация алгоритмов на языке С придают процессору полную завершенность.

VII. Сегментационный анализатор немецкого предложения (STP).

Немецкими учеными из Исследовательского Центра Искусственного Интеллекта в Saarbruecken в конце 90-х гг. был создан Поверхностный Текстовый Процессор (Shallow Text Processor) [G. Neumann, J. Piskorski, 2001]. STP, как и лингвистические процессоры русского языка, относится к классу модульных систем (vs. HPSG и Link Grammar), характеризующихся разделением на функционально независимые компоненты, каждый из которых соответствует одному из уровней лингвистического анализа. STP первоначально разрабатывался для немецкого языка, хотя сейчас предпринимаются попытки перенести технологию анализатора на материал английского и японского языков. Архитектура процессора делится на две составляющих: лексический уровень и сегментный уровень. Графематический анализ (разбиение предложения на слова, выделение знаков препинания,

аббревиатур, etc.), морфологический анализ (лемматизация входных словоформ), модуль снятия частеречной омонимии, выделение шаблонных групп (темпоральные группы в тексте (время и даты), организации, персоналии, географические имена) относятся к лексическому уровню системы. Сегментный уровень состоит из трех модулей: сегментация предложения, построение именных (NP) и предложных (PP) групп внутри сегментов, установление грамматических функций (поиск комплементов глагола с использованием глагольной модели управления).

Принципиальным решением в процессоре называется отказ от традиционного анализа «снизу вверх» и применение принципа «разделяй и властвуй» [Т. Кормен и др., 2001, стр. 26] для вычисления синтаксической структуры предложения. В первой фазе синтаксического анализа определяется топологическая структура (topological structure) предложения (т.е. выделение глагольных групп (VP) и сегментов), во второй фазе происходит выделение фразовых категорий в пределах, определенных границами сегментов. Таким образом, в первой фазе анализ предложения проводится «сверху вниз», во второй – «снизу вверх», но на фрагментах меньше длины предложения. Нужно сказать, что идея необходимости разделения сегментационного и непосредственно синтаксического (в смысле установление связей между отдельными словами) анализа – параллельное построение сверху и снизу структуры предложения – существовала в московской прикладной лингвистике еще в 70-ые годы. Такая стратегия позволяет значительно снизить стоимость вычислений. Процессор не ставит своей целью построения полного графа предложения, поэтому результатом синтаксического разбора является частично связанное дерево зависимостей. При таком подходе сегментационный анализ становится центральным идеологическим компонентом архитектуры системы.

Большая часть алгоритмов в системе, включая определение топологической структуры предложения, реализована на машинах конечных состояний (FSM - finite-state machine) [Т. Кормен и др., 2001, стр.788], что значительно повышает скорость вычислений и эффективность распределения памяти. Большая часть грамматик фразовых категорий (NP и PP) представлена на регулярных выражениях [Дж. Фридл, 2001], преобразующихся в конечные автоматы. В STP используется два основных типа FSM: простые (FST - finite-state transducer) и весовые (WFST) трансдюсеры. FST называется автомат, в

котором каждый переход между состояниями в сети имеет выходную помету в дополнение к входной [XRCE MLTT, 1995]. Например, FST служит для представления контекстных правил в модуле снятия частеречной омонимии. WFST позволяет присваивать веса своим переходам и состояниям в сети.

Топологической структурой предложения называется разбиение всего предложения, равно как и его отдельных сегментов, на определенные зоны-поля (fields), границы которых определены глагольной группой и подчинительным союзом (союзным словом) в случае придаточного. Для топологической структуры немецкого языка использовано свойство аналитической глагольной группы, которая может быть разделена на две части (“haette ueberredet werden muessen”: “haette” и “ueberredet werden muessen”): левую (LVP) и правую (RVP). Как следствие такого деления, возникают поля структуры: фронтальное (FF), LVP, среднее (MF), RVP и остаток (RF). Для простого предложения “Er haette gestern ueberredet werden muessen” (“Он должен бы был быть предупрежден вчера”) структура выглядит следующим образом:

FF	LVP	MF	RVP	RF
Er	Haette	gestern	ueberredet werden muessen	ПУСТОЕ

То же верно и для придаточного предложения только лишь с той разницей, что LVP будет либо пустым, либо занято подчинительным союзом (союзным словом), а RVP займет полная (неразрывная) глагольная группа. Каждое отдельное поле может быть произвольно сложным. Структура вложенного в главное придаточного определительного предложения “Der Mann, der gestern haette ueberredet werden muessen, lief nach Hause” (“Человек, который должен бы был быть предупрежден вчера, уехал домой”):

FF	LVP	MF	RVP	RF
ПУСТОЕ	Der	gestern	haette ueberredet werden muessen	ПУСТОЕ

С помощью такого рода топологической структуры изящно обыгрывается принцип полноты (наличие предиката, союза и их окружение) всего предложения, а вместе с тем и его отдельных сегментов. Алгоритм, осуществляющий сегментацию, начинает с топологической структуры вложенных сегментов и, сворачивая подчинительные до уровня нетерминальных единиц, заканчивает структурой простого предложения в составе сложного. Алгоритм сегментации – рекурсивный и состоит из четырех ступеней анализа, каждой из которых соответствует своя стратегия (грамматика):

1. идентификация глагольных групп (VG);
2. идентификация базовых фрагментов (BC);
3. комбинирование последовательностей базовых фрагментов (CC) с целью формирования расширенных единиц (полных сегментов); если расширенная единица не идентифицирована, то перейти на шаг 4, иначе вернутся на шаг 2;
4. идентификация главных (простых) сегментов (MC);

VG выделяет как отдельные глаголы, так и цепочки глагольных форм в линейной последовательности предложения. Каждому глаголу приписывается его морфологическое значение, а в случае грамматической омонимии значения глагольных форм связаны оператором дизъюнкции.

BC членит исходное предложение по знакам пунктуации на отдельные фрагменты, присваивая каждому фрагменту свой тип, исходя из наличия/отсутствия подчинительного союза или глагольной формы. Фрагменты с подчинительными союзами называются базовыми.

CC анализирует рекурсивные вложения базовых фрагментов на основе их топологической структуры. Выделяются два возможных типа рекурсивных вложений: (а) среднего поля (MF-рекурсии), когда вложенный сегмент заключен между левой LVP и правой RVP частями глагольной группы подчиняющего сегмента; (б) остатка (RF-рекурсии), когда вложенный сегмент следует за RVP подчиняющего сегмента. Через операцию вложения подчинительных сегментов CC расширяет базовые фрагменты, рекурсивно доводя их до полных сегментов.

MC осуществляет анализ топологической структуры простого сегмента в составе сложного предложения и определяет сочинение сегментов.

Рассмотрим последовательность действий для сегментации предложения “Weil die Siemens GmbH, die vom Export lebt, Verluste erlitt, musste sie Aktien verkaufen” (“Потому что фирма Siemens GmbH, которая зависит от экспорта, понесла убытки, они вынуждены были продавать акции”):

Weil die [*company* Siemens GmbH], die … [Verb-Fin], V. [Verb-Fin], [Modv-Fin] sie A. [FV-Inf]	Шаг 1. (VG)
Weil die [*company* Siemens GmbH] [Rel-Cl], V. [Verb-Fin], [Modv-Fin] sie A. [FV-Inf]	Шаг 2. (BC)
[Subconj-CL], [Modv-Fin] sie A. [FV-Inf]	Шаг 3. (CC) MF-рекурсия и возврат

	на шаг 2.
[Subconj-CL], [Modv-Fin] sie A. [FV-Inf]	Шаг 3. (CC) (без изменений) переход на шаг 4
[clause]	Шаг 4 (MC)

Результат сегментации в скобочной записи: [*MAIN-CL* [*SUB-CL* Weil die [*company* Siemens GmbH] [*REL-CL* , die vom Export lebt], Verluste erlitt], musste sie Aktien verkaufen.]

После завершения работы сегментации проводится построение NP и PP групп внутри сегментов и установление грамматических функций на основе лексикона, где хранится около 12 тысяч глаголов с возможными моделями управления (subcategorization frames).

В описание процессора не включена информация о построении или разрешении синтаксической омонимии на уровне сегментов, т.е. возможность рассмотрения структурных вариантов сегментации предложения с разными границами сегментов. Также нет упоминания о сочинении предикатов – важной составляющей анализа для правильного определения границ сегментов.

Программная реализация системы первоначально выполнена на языке LISP, а затем переведена на C++. Тестирование STP немецкого языка демонстрирует высокую точность и скорость анализа.

ГЛАВА 2. МОРФОЛОГИЧЕСКИЙ И ПРЕДСИНТАКСИЧЕСКИЙ АНАЛИЗ

Настоящая глава посвящена проектированию морфологического анализа и некоторым методам снятия грамматической омонимии и построения именных групп, которые являются неотъемлемой частью современного синтаксического процессора. На нынешнем этапе развития промышленные варианты лингвистических анализаторов заканчиваются на уровне выделения именных np-групп, и по-прежнему центральным звеном таких систем остается морфология. В разделах III и IV главы рассматривается один из наиболее успешных проектов промышленного лингвистического процессора.

I. Прикладной морфологический анализ без словаря.

В 60-70 гг. все экспериментальные исследования в области машинной морфологии начинались с создания машинного словаря. Не было единого общепринятого формата и структуры такого словаря. Эти обстоятельства имели два последствия: во-первых, все алгоритмы автоматически становились словарнозависимыми, во-вторых, каждый алгоритм разрабатывался под определенный формат словаря. На современном этапе развития информационных технологий морфологический компонент стал неотъемлемой частью интеллектуальных информационно-поисковых систем (ИПС).

Основная проблема в разработке машинно-ориентированного алгоритма для лингвистических процессоров состоит в объеме исходных данных, используемых программой, то есть в объеме словарей, которые приходится составлять вручную. Исследования в этой области направлены на минимизацию исходных данных. Работы, посвященные морфологии, можно условно разделить на две категории:

1. теоретические, в некоторых представлены описания морфологических законов и формальные модели русской морфологии;
2. прикладные, описание програмно-реализованных систем с морфологическим модулем.

В теоретических работах строятся многоуровневые формальные модели морфологии, в большинстве своем, предназначенные для синтеза. Такие модели морфологического синтеза подразумевают наличие больших словарей со сложной структурой. Они описывают широкий круг морфологических явлений. Многие компоненты этих моделей избыточны для задач машинного анализа (фонетическая реализация слова, акцентная парадигма, большое число словообразовательных аффиксов).

В теоретической работе “Формальная модель русской морфологии” [Н.А.Еськова, И.Г.Бидер и др.] дается полное описание морфологических явлений русского языка и нестандартные решения для их формализации. Перечислим важные особенности данной модели: (а) различение морфологического и (б) синтаксического рода[3]; (в) отнесение темы глагола (‘-

[3] ***Пример:***

‘Мужчина’ : морфологический род = женский; синтаксический род = мужской.

‘Подмастерье’ : морфологический род = средний; синтаксический род = мужской.

ов-', '-у-', '-а-' и т.д.) к флексии; (г) метод описания чередований для существительных и различение для супплетивных основ[4]; (д) выделения специальных признаков глагола, различные комбинации значений которых покрывают все возможные в русском языке способы видообразования (всего 32 комбинации); (е) отсечение отрицания (частицы 'не') у существительных и прилагательных. Недостатками такой модели является ее сложность: (а) несколько уровней представления морфологической информации, специальные грамматики для перехода с одного уровень на другой; (б) избыточность грамматических признаков, часть из которых выделены в модели для описания частных случаев.

Модели, которые используют словарь, способны дать более полный анализ словоформы (т.е. оперировать большим числом грамматических признаков). Степень точности такого анализа выше, по сравнению с моделями, которые не используют словаря. В разделе II текущей главы будет рассмотрен ряд систем морфологического анализа с использованием грамматических словарей. На пространстве реальных текстов системы, использующие словарь, часто дают сбои. Это обусловлено тем, что не существует полных словарей. Лексика языка непрерывно пополняется - появляются новые слова. Для каждой предметной области существует своя терминология, свое подмножество лексики языка, и включить в общий словарь всю существующую терминологию - невозможно. Равно как невозможно и перечислить все существующие имена и фамилии, которые имеют регулярное склонение.

Алгоритмы программ, работающих без словаря, используют вероятностно-статистические методы и лексиконы суффиксов или квази-суффиксов, основ или квази-основ, построенных эмпирически. В статье "Эмпирическое моделирование в вычислительной морфологии" [С.О.Шереметьева, С.Ниренбург, 1996] описана работающая модель морфологического анализа, не требующая объемных словарей основ открытых классов слов. Модель разработана в русле инженерной лингвистики. Модель использует следующие лексиконы:

1. Лексикон окончаний и рефлексивов;

---

[4] ***Пример:*** *небо'-'небес' (тема 'ес'); 'мать'- 'матери' (тема 'ер').*

2. Лексикон суффиксов;

3. Лексикон квази-корней;

4. Лексикон префиксов;

5. Лексикон баз;

6. Лексикон основ.

Каждой единице такого лексикона приписаны все возможные грамматические характеристики словоформ, частью которой может являться данная единица. Пример единицы лексикона квази-корней:

-ени-

существительное, 11, -е,

существительное, 8, -й,

глагол, -ть;

где 11, 8 - тип склонения.

Анализ словоформы в модели построен на правилах поиска и сочетания единиц разных лексиконов, что приводит к унификации гипотез.

Такой анализ не использует возможности текстов, поступающих на вход системы. По сути, предлагаемый метод сводится к эмпирическому сжатию исходного словаря словоформ. Для этого выделяются общие цепочки букв в множестве словоформ, и каждой цепочке букв приписываются все возможные значения грамматических категорий этих словоформ. Эмпирическое сжатие грамматического словаря русского языка приводит к созданию большого числа разрозненных лексиконов разной структуры, каждый из которых требует отдельной процедуры считывания данных. В статье не описана технология формирования лексиконов. Данный подход к морфологическому анализу нельзя назвать, в полной мере, бессловарным.

Похожий метод используется в работах Г.Г.Белоногова [Г.Г.Белоногов, 1984], где дается описание вероятностно-статистических методов для создания вспомогательных лексиконов на основе исходного корпуса текстов.

Все алгоритмы такого рода имеют одни и те же недостатки:

(1) не используются точные лингвистические методы анализа;

(2) большой объем лексиконов;

(3) вероятностно-статистические методы плохо работают с малой выборкой.

Точность такого анализа намного ниже, чем для систем, работающих со словарем. Эти алгоритмы не позволяют выбирать уникальные грамматические характеристики, хотя в большинстве случаев позволяют построить общую основу или квази-основу для множества словоформ и лемматизировать словоформу.

Наиболее свободная форма анализа была разработана в Чикагском Университете [J. Goldsmith, 1999]. Модель позволяет путем статистической обработки большого массива текстов, анализируя частоту встречаемости последовательности символов в словоформах, выделять множество аффиксов и корневых морфем, релевантных для заданного языка. Программа работает с большинством европейских языков, включая русский. Работа проводилась в рамках научного исследования и не получила прикладного внедрения.

В этом разделе предлагается описание модели прикладного морфологического анализа без словаря, разработанной автором диссертации в НТЦ "Система" в период с 1997 по 1998 гг. Алгоритмы морфологии построены на самообучении программы на открытых массивах реальных текстов и совмещают два подхода: лингвистический - формализованная грамматика для построения морфологических гипотез и математический - метод корреляции, позволяющий унифицировать морфологическую гипотезу. Морфологический анализ без словаря является центральной компонентой системы автоматической индексации текстовой базы данных (БД), реализованной в СУБД Oracle8i. Выходным результатом системы является автоматически построенный грамматический словарь основ и связанный с ним индекс документов, предназначенный для полнотекстового поиска по БД.

Сущность интеллекта состоит в способности принимать разумные решения в условиях отсутствия полноты данных и фактов [M. Boden, 1990]. Интеллектуальность системы повышается с уменьшением объема статической информации, используемой в процессе анализа данных. В нашем случае, речь идет об использовании лингвистической информации при морфологическом анализе в задачах автоматической индексации текстовых БД. Ниже будут выделены основные критерии, отличающие большинство интеллектуальных систем, которых придерживается описываемый процессор автоиндексации текстов:

- Способность системы объяснить каждой шаг принятых решений. В процессе анализа не используются вероятностные и статистические методы.
- Использование правил и свойств, характеризующих данный предмет анализа. Для построения морфологических гипотез словоформ используется формализованная грамматика и то свойство русского языка, что большая часть грамматических категорий в русском вычисляется из флексии.
- Модульность системы, которая обеспечивает эффективное изменение и пополнение правил и свойств, а также задает возможность настраивать анализатор на другие естественные языки с развитой морфологией.
- Множественность интерпретаций. Анализатор оставляет все омонимы значений словоизменительных категорий.
- Самообучаемость и механизм исправления принятых ранее неверных решений. Объем прочитанных текстов пополняет число словоформ, используемых в процессе анализа, тем самым повышая точность морфологического анализа и позволяя корректировать неправильно построенные основы и значения их грамматических категорий.
- Моделирование интеллектуального поведения человека. В данном случае, речь идет о попытке эмулировать размышления человека, изучающего иностранный язык, перед которым стоит задача классифицировать слова данного языка, в условиях, когда в его распоряжении находится большой массив текстов, некоторые знания о морфологии языка и отсутствует словарь языка, на котором написаны тексты. Надо сказать, что при разработке алгоритмов не ставилось задачи опровергнуть мысленный эксперимент Джона Сёрля “Китайская комната” [см. гл. 1, раздел II].

Модель будет рассмотрена на уровне общего описания процессора - взаимодействие его модулей и функциональная схема алгоритма морфологического анализа [И. Ножов, 2000].

Схема процесса автоматической индексации представлена на рис.1: на вход процесса автоиндексации поступает все множество текстов, хранящихся в

базе данных, на выходе формируется словарь основ и таблица соответствий (текст ⇔ основа), которая отображает поток индексированных текстов.

Блоки, которые осуществляют процесс автоиндексации, представлены на рис.2.

Процессы (рис.2):

1. Графематический анализ.
2. Морфологический анализ.

На рис.3 показана схема таблиц для хранения потоков данных, сформированных процессами графематического и морфологического анализа.

Потоки данных (рис.3):

1. Тексты;
2. Полные словоформы;
3. Аббревиатуры;
4. Цифровые и символьные комплексы;
5. Основы и значения их грамматических категорий;

Основная цель графематического блока получить выборку полных словоформ из массива текстов БД. Графематический анализ работает с внешним представлением текста и использует таблицу стоп-слов. В этой таблице хранятся цифры, спецсимволы и частотные слова языка, нерелевантные для поиска по текстам.

Графематический анализ выполняет три функции:

1. отсечение стоп-слов в тексте;
2. разбиение данных на три потока;
3. индексация каждого потока.

Единицей графематического анализа является цепочка символов, выделенная с двух сторон пробелами. Выделенная цепочка символов подвергается последовательной обработке эвристическими правилами: отсечь знаки пунктуации, проверить присутствие гласных внутри цепочки, чередование верхнего и нижнего регистров и т.д. В зависимости от результатов обработки полученная цепочка символов направляется в один из трех потоков данных:

— цифровые и символьные комплексы (‘кг’, ‘ст.’, ’12.01.99’);

— аббревиатуры - названия государств, организаций, предприятий (‘СССР’, ‘ЮНЕСКО’, ‘ДорСтройСервис’);

— полные словоформы;

Каждой записи из любого потока ставятся в соответствие коды документов, в которых она встретилась. Первых два потока данных считаются проиндексированными, причем только аббревиатуры являются релевантным поисковым образом. Графематику можно считать лишь вспомогательным звеном для морфологического анализа. Графематический и морфологический процессы способны проиндексировать массивы текстов независимо от предметной области конкретной базы данных.

Полные словоформы поступают на вход морфологического анализа, цель которого разбить все множество словоформ на подмножества по признаку принадлежности к той или иной лексеме[5], привести все элементы каждого такого подмножества к уникальной основе, однозначно определить грамматические характеристики лексемы и проиндексировать тексты по встретившимся в них основам.

[5] *Лексема - это множество словоформ, отличающихся друг от друга только словоизменительными значениями [И. Мельчук, 1997].*

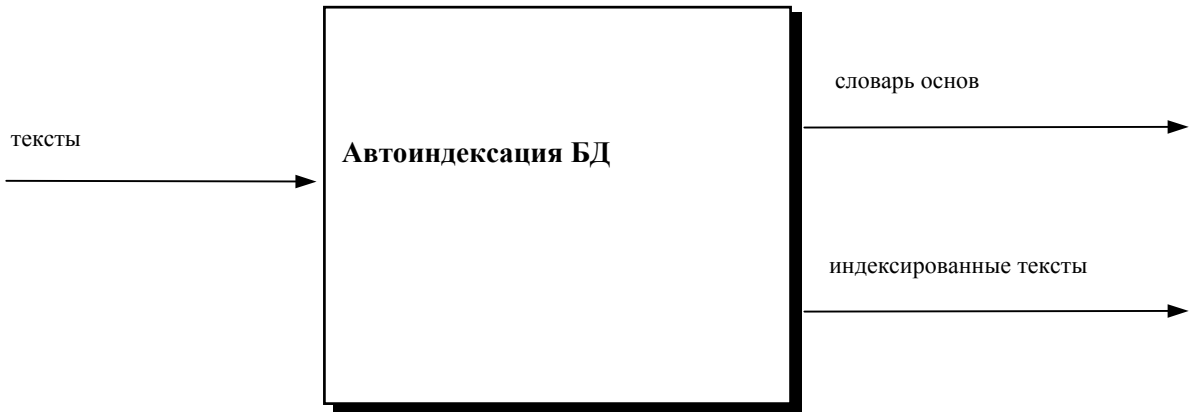

тексты
Автоиндексация БД
словарь основ
индексированные тексты

рис.1

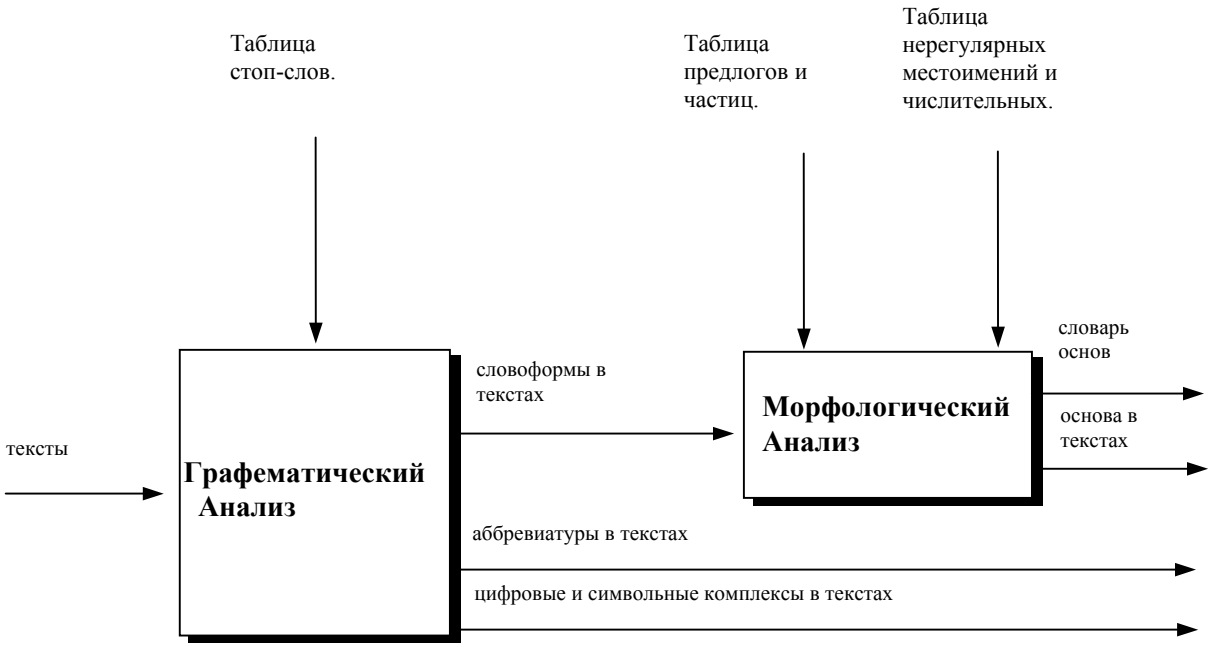


Таблица стоп-слов.
Таблица предлогов и частиц.
Таблица нерегулярных местоимений и числительных.
тексты
Графематический Анализ
словоформы в текстах
Морфологический Анализ
словарь основ
основа в текстах
аббревиатуры в текстах
цифровые и символьные комплексы в текстах

рис.2

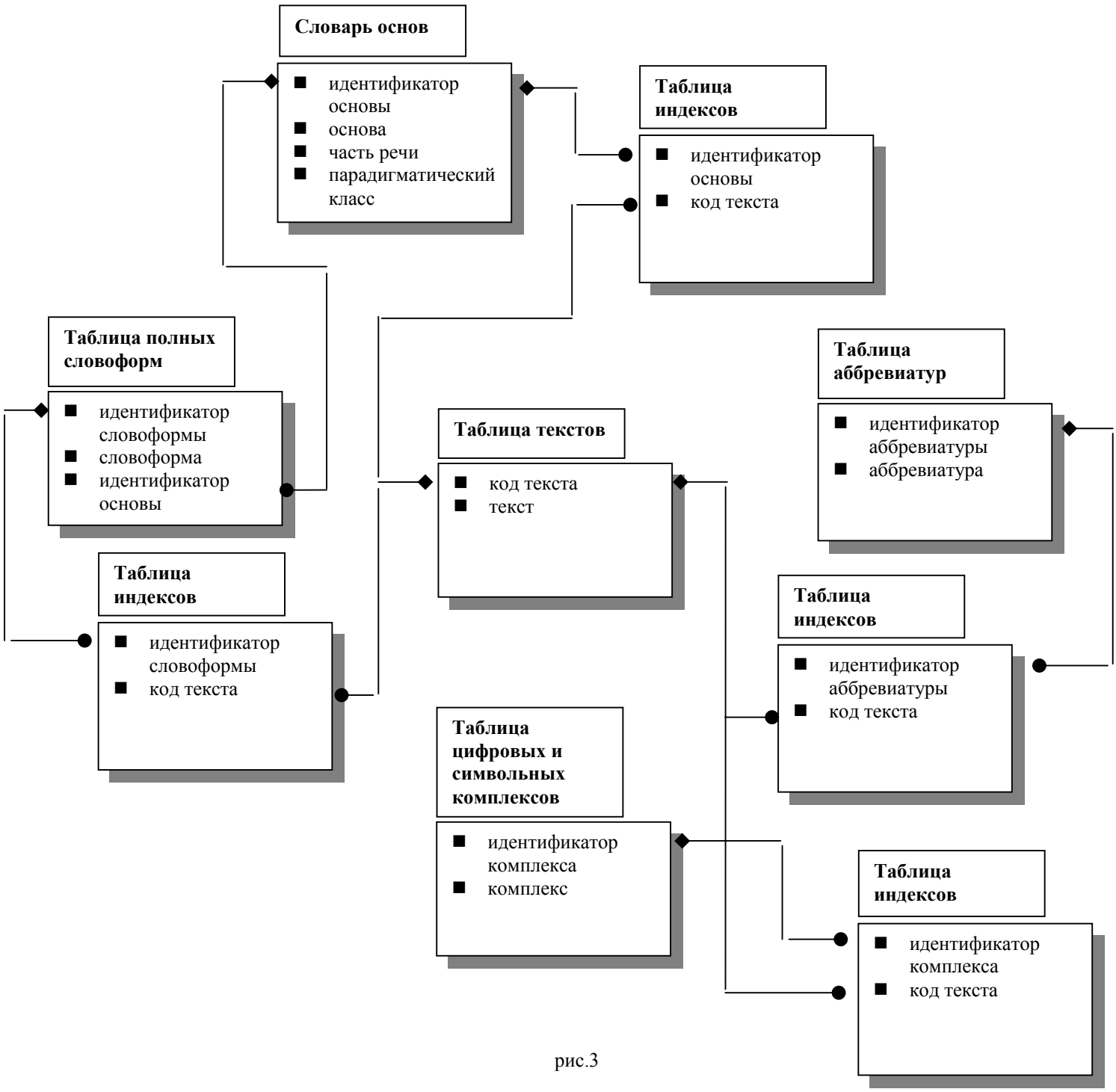

Словарь основ
идентификатор основы
основа
часть речи
парадигматический класс
Таблица индексов
идентификатор основы
код текста
Таблица полных словоформ
идентификатор словоформы
словоформа
идентификатор основы
Таблица текстов
код текста
текст
Таблица аббревиатур
идентификатор аббревиатуры
аббревиатура
Таблица индексов
идентификатор словоформы
код текста
Таблица индексов
идентификатор аббревиатуры
код текста
Таблица цифровых и символьных комплексов
идентификатор комплекса
комплекс
Таблица индексов
идентификатор комплекса
код текста

рис.3

Блок морфологического анализа использует минимальный объем исходной информации:

— таблицу предлогов;
— таблицу местоимений и числительных, имеющих нерегулярное склонение.

На выходе морфологического анализа формируется словарь основ данной БД, уникальность записи в таком словаре задается тройкой значений [основа, часть речи, парадигматический класс]. Морфологический анализ состоит из трех модулей и соблюдает определенную последовательность действий.

Первый модуль содержит статический массив флексий и правила формализованной грамматики русской морфологии, построенной на основе работ А.Зализняка [А.Зализняк, 1980]. Выделение парадигматических классов в модели полностью соответствует парадигматическим классам в словаре А.Зализняка. Это - восемь типов склонения существительных и прилагательных и шестнадцать типов парадигмы глагола, которым соответствует первое или второе спряжение. В словаре А.Зализняка глагольная тема (*‘ов’, ‘у’* и т.д.) входит в окончание глагола. В нашем случае вводится термин расширенная флексия глагола. Расширенной флексией глагола называется конкатенация чередующейся глагольной темы и флексии.

Данный модуль может быть заменен формализованной морфологией любого другого флективного языка. Методы, описанные в модулях два и три, являются универсальными, независящими от языка.

Второй модуль, используя правила формализованной грамматики, позволяет строить морфологическое дерево словоформы, в узлах которого хранятся все возможные гипотезы об основах и значениях грамматических категорий словоформы. Морфологические правила делятся на два класса. Первый класс правил, которые порождают некоторые грамматические характеристики для гипотез, и второй класс правил накладывает определенные ограничения на гипотезы. Пример правил первого класса: *если гипотеза об основе оканчивается на согласную ряда {‘к’, ‘г’, ‘х’}, то тип склонения равен трем* или *если исходная словоформа не оканчивается на гласную, то построить гипотезу о существительном с нуль-флексией.* Пример правил второго класса: *если гипотеза о флексии равна ‘ет’ [3 лицо, ед. ч.] или ‘ю’ [1*

лицо, ед. ч.], и гипотеза об основе оканчивается на сегмент первой ступени чередования [А.Зализняк, 1980], *то гипотеза о глаголе не верна.*

Традиционно в синтаксических и семантических теориях используется представление языковой структуры с помощью деревьев. В описываемой системе, пожалуй, впервые данный формализм оправдано был применен к морфологии.

Третий модуль содержит метод подбора словоформ на одну лексему[6], то есть выбор коррелятов для дерева исходной словоформы. После того, как набраны корреляты, для каждой словоформы также строится морфологическое дерево всех возможных гипотез, в результате чего образуется "лес деревьев" [Ф.Харари, 1973]. Метод корреляции[7] осуществляет сравнение морфологических деревьев внутри леса и унификацию гипотез. Корреляция проводится по гипотезам основ и значениям классифицирующих грамматических категорий, таких как часть речи, парадигматический класс, спряжение глаголов и род существительных. Значения словоизменительных категорий в корреляции не участвуют. Во время работы корреляции происходит удаление ложных гипотез: ветвей дерева или полного дерева коррелята. Этот модуль позволяет построить уникальную гипотезу об основе и значениях ее грамматических категорий для всех словоформ одной лексемы, найденных в текстах. Метод корреляции очищает лес от ложных коррелятов, оставляя, таким образом, только словоформы, принадлежащие одной лексеме. Уникальная основа, единая для всех словоформ, участвовавших в корреляции, значение части речи и парадигматического класса добавляются в словарь основ. По сути, основа в словаре репрезентирует лексему.

Для унификации гипотезы метод корреляции использует матрицы корреляций. Лесом называется множество деревьев словоформ F = $\{T_1,..,T_j,..,Tn\}$. Множество всех построенных гипотез об основе в F обозначим U = $\{s_1,..,s_i,..,s_m\}$. Параметром корреляции t называется значение грамматической категории. Матрицей корреляции A(t) = $\|a_{ij}\|$ леса F с m гипотезами об основах и

[6] *Словоформы, которые гипотетически принадлежат одной лексеме, для сокращения записи мы будем называть "словоформы на одну лексему" [прим. автора].*

[7] Данный метод корреляции был разработан специально для задачи морфологического анализа и не имеет ничего общего с его вероятностно-статистическим аналогом, предназначенным для решения других задач [прим. автора].

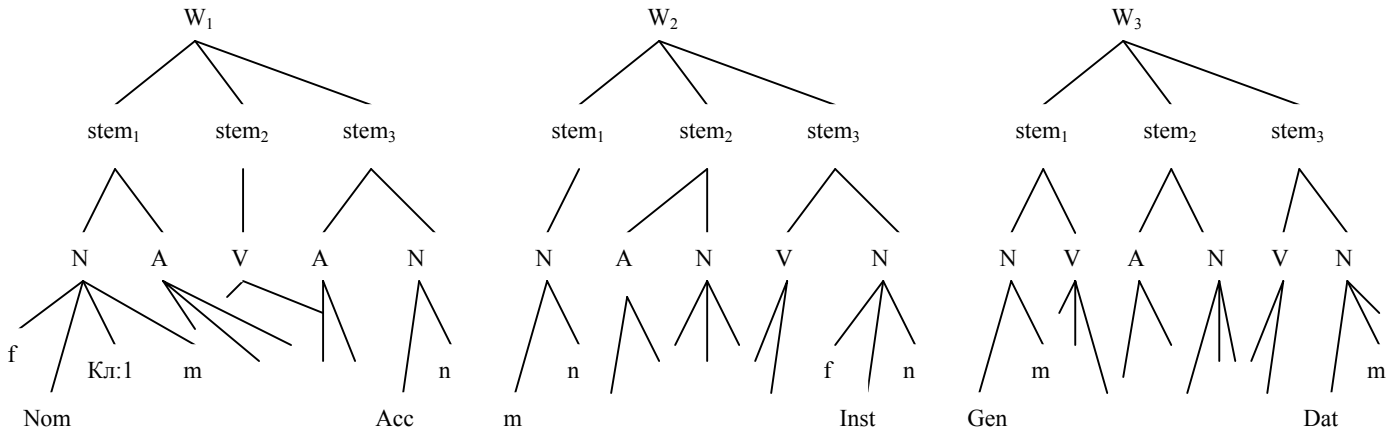
n деревьями словоформ называется $(m \times n)$-матрица, в которой $a_{ij} = 1$, если заданный параметр корреляции t определен для s_i в T_j, и $a_{ij} = 0$ в противном случае.

В процессе корреляции отдается приоритет гипотезам исходной словоформы, на основе которых подбираются корреляты, что позволяет избежать ситуации, когда лес вырождается в пустое множество. Число матриц корреляции внутри одного типа корреляции определяется по числу возможных значений грамматической категории: так, в процессе корреляции по роду существительных для русского будет построено три матрицы, соответствующие трем возможно задействованным в деревьях значениям грамматического рода. Для каждой матрицы корреляции находится

$$k = \max_{i:a_{i1}\neq 0} \sum_{j=1}^{n} a_{ij}$$

после чего из множества значений k внутри одного типа корреляции также выбирается максимальное значение, которое и соответствует унифицированной гипотезе. Узлы не получившие максимального значения удаляются из деревьев словоформ. Условие $a_{i1} \neq 0$ задает приоритет гипотезам дерева исходной словоформы T_1.

Допустим в прочитанных программой текстах было подобрано два коррелята для исходной словоформы W_1, тогда лес F состоит из трех деревьев словоформ W_1, W_2 и W_3 (рис.4):



Р

ис.4

Корреляция по части речи:

матрица корреляции | значение k | максимальное значение внутри типа корреляции

$$\text{Noun} \begin{bmatrix} 111 \\ 011 \\ 111 \end{bmatrix} \bar{1} = \begin{bmatrix} 3 \\ 2 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 3 \\ stem1 & stem3 \end{bmatrix}$$

$$\text{Adj} \begin{bmatrix} 100 \\ 011 \\ 100 \end{bmatrix} \bar{1} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ stem1 & stem3 \end{bmatrix} \text{ --- } [3, 3, 1, 1, 1] \Rightarrow \text{Noun} \begin{bmatrix} 3 & 3 \\ stem1 & stem3 \end{bmatrix}$$

$$\text{V} \begin{bmatrix} 001 \\ 100 \\ 011 \end{bmatrix} \bar{1} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ stem2 \end{bmatrix}$$

Удаляются ложные узлы деревьев словоформ леса F (рис. 5):

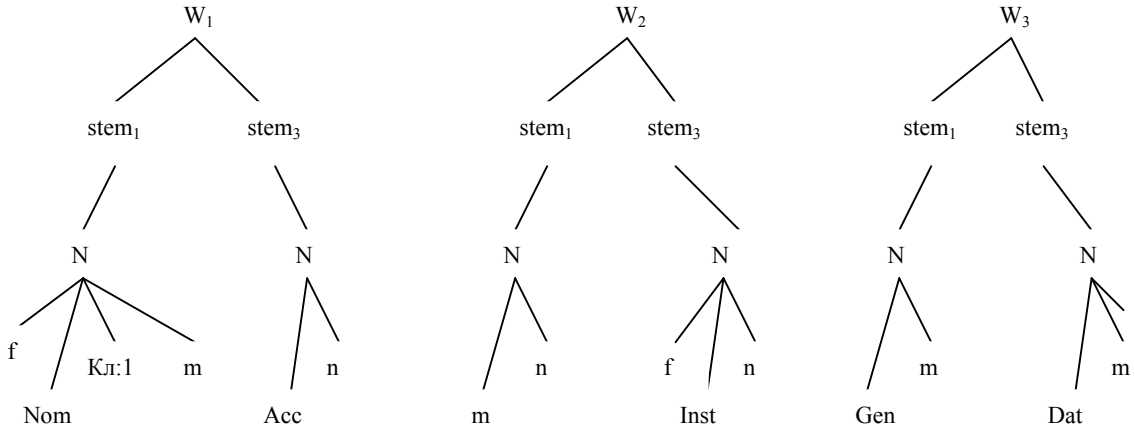


Рис.5

Корреляция по роду:

матрица корреляции | значение k | максимальное значение внутри типа корреляции

$$m \begin{bmatrix} 111 \\ 001 \end{bmatrix} \bar{1} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 \\ stem1 \end{bmatrix}$$

$$n \begin{bmatrix} 010 \\ 110 \end{bmatrix} \bar{1} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 \\ stem3 \end{bmatrix} \quad \text{--- } [3, 2, 1] \Rightarrow m \begin{bmatrix} 3 \\ stem1 \end{bmatrix}$$

$$f \begin{bmatrix} 100 \\ 010 \end{bmatrix} \bar{1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ stem1 \end{bmatrix}$$

После завершения корреляции по роду и удаления не получивших максимального значения узлов гипотеза унифицирована: W_1[$stem_1$[N[Кл:1, m, Nom, …]]]; W_2[$stem_1$[N[m, …]]]; W_3[$stem_1$[N[m, Gen, …]]].

Часто задаваемый вопрос - почему в качестве формализма выбраны деревья, а не кортежи. Деревья позволяют сделать метод корреляции универсальным, независимым от выбранного для анализа естественного языка. Как видно из примеров, ширина дерева произвольна, а высота фиксирована и равна трем для русского языка. Высота дерева, также как и ширина, может изменяться при переходе от одного анализируемого языка к другому и определяется морфологической грамматикой, т.е. существующими

зависимостями между грамматическими категориями и их показателями в каждом конкретном языке, что делает использование кортежей затруднительным, а «древесный» формализм сохраняет независимость метода корреляции от морфологических правил рассматриваемого языка.

Рассмотрим реальный пример. Визуальный интерфейс программы морфологического анализа позволяет наблюдать состояние леса до корреляции и после, как это показано на примере анализа глагола ‘текут’ (Рис.6 и Рис.7).

Подбор коррелятов и построение леса деревьев возможных гипотез для глагола ‘текут’:

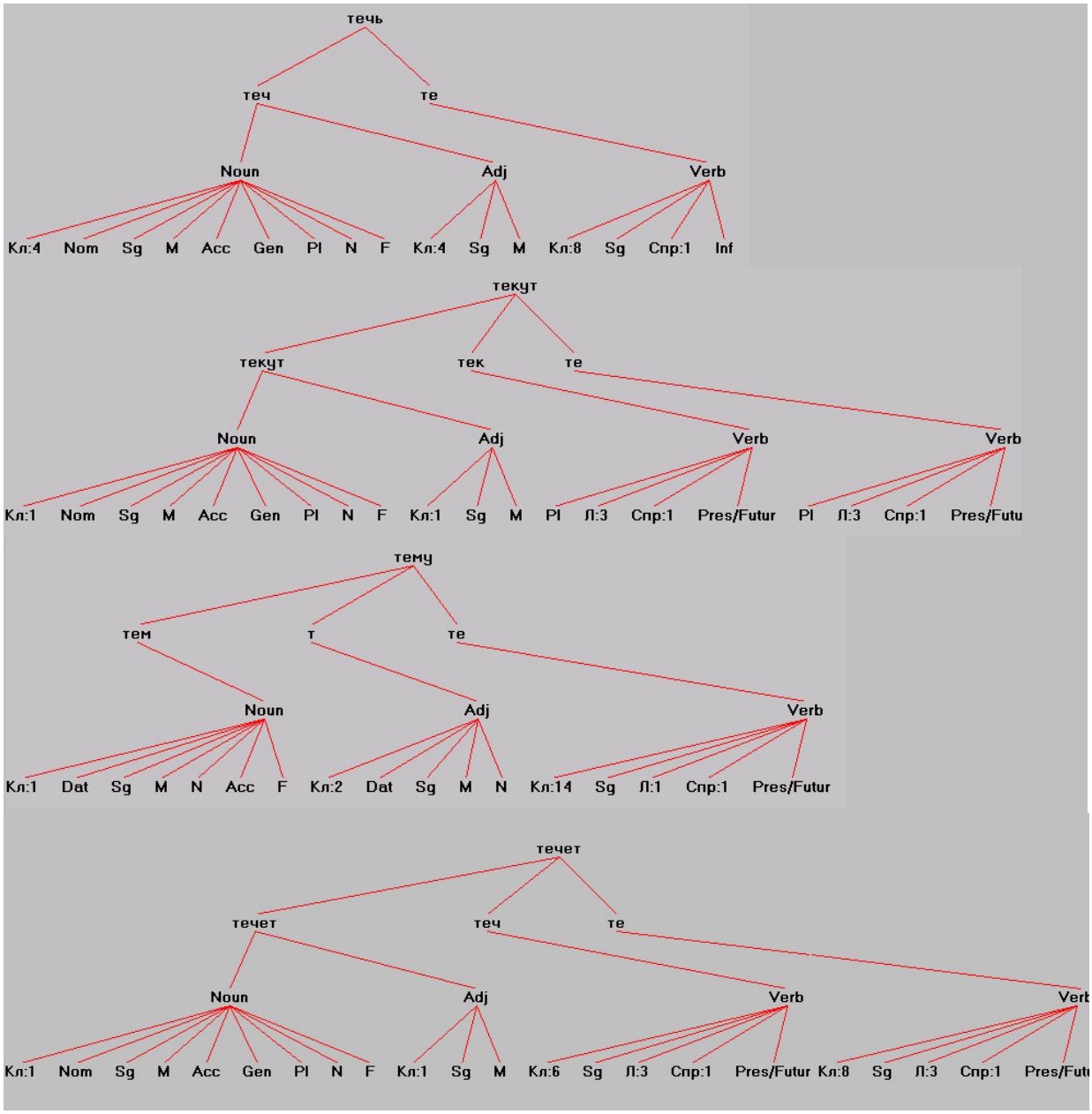


рис.6

Применение метода корреляции (унификация гипотезы и удаление ложных коррелятов):

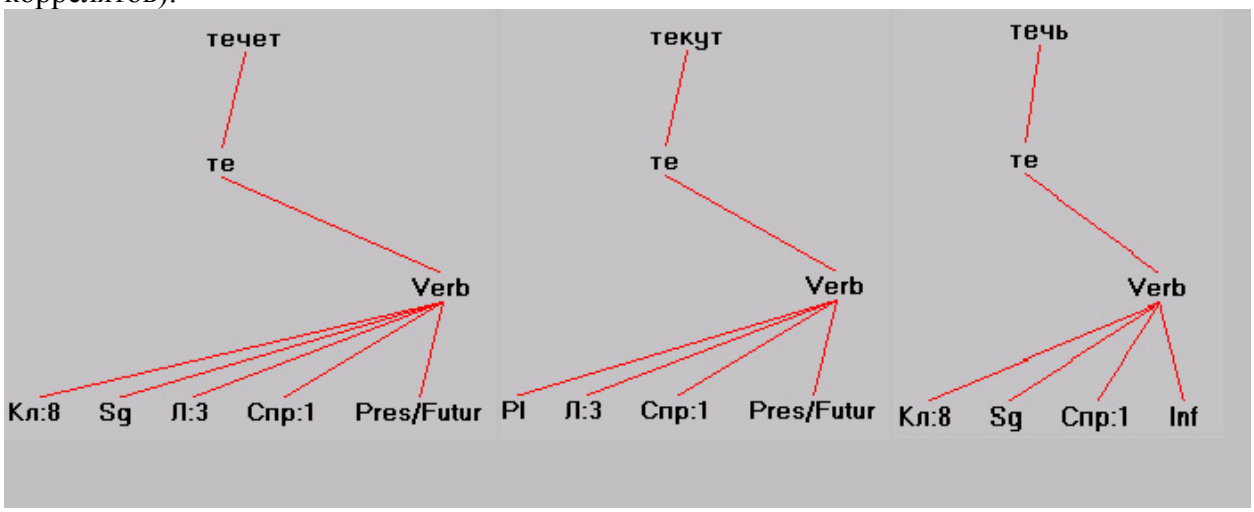


рис.7

В результате остается три дерева коррелятов с уникальными гипотезами об основе, части речи и грамматических характеристиках.

Последовательность шагов (Д1..Д13) алгоритма морфологического анализа без словаря представлена на рис.8.

Общая схема алгоритма.

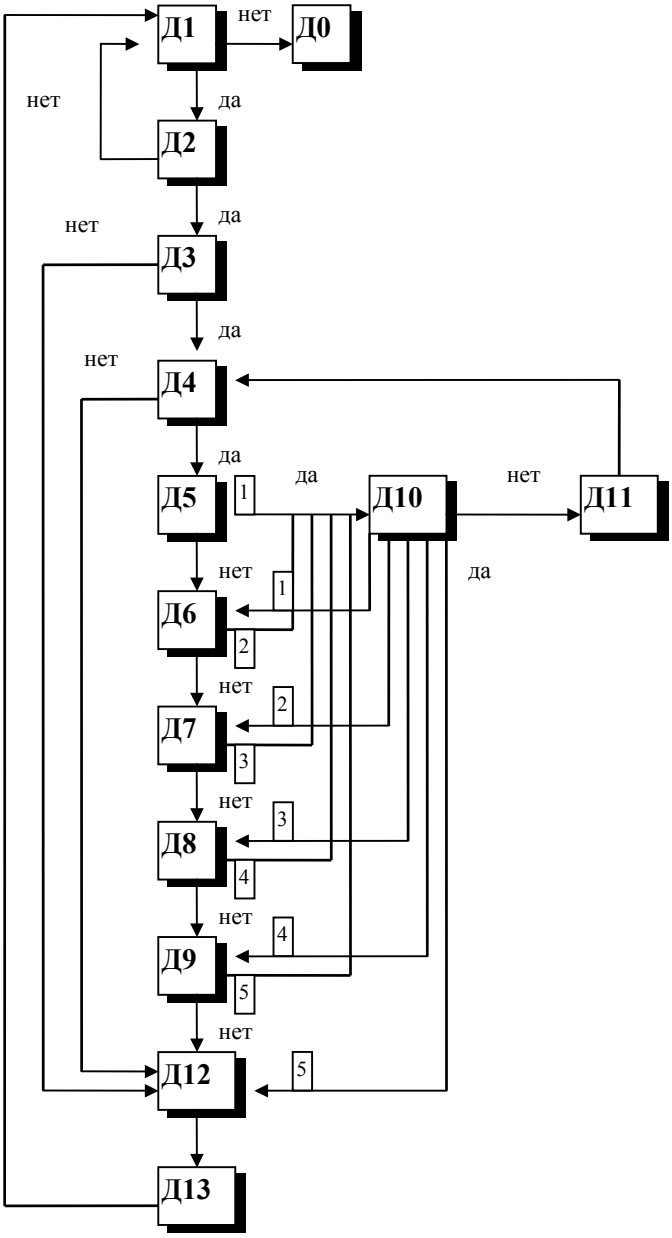


рис.4

Д0. Выход из программы.

Д1. Выбрать из таблицы полных словоформ (рис.3) непроиндексированную словоформу, то есть словоформу, для которой еще не построена основа (ДА: словоформа выбрана; НЕТ: все словоформы в таблице проиндексированы).

Д2. Проверить, что данная словоформа не является предлогом или местоимением. Построить дерево всех возможных гипотез для данной словоформы. (ДА: не является; НЕТ: является)

Д3. Выбрать из таблицы полных словоформ (рис.3) словоформы на одну лексему. Создать список коррелятов.

(ДА: корреляты выбраны; НЕТ: список коррелятов пуст)

Д4. Если список коррелятов непустой, то построить деревья всех возможных гипотез для каждого коррелята.

Д5[8]. Провести корреляцию по гипотезам основ.

Д6. Провести корреляцию по значениям части речи.

Д7. Провести корреляцию по значениям спряжения глагола.

Д8. Провести корреляцию по значениям рода существительных.

Д9. Провести корреляцию по значениям парадигматического класса.

Д10. Проверить, что корреляция не привела к удалению полного дерева (дерева коррелята) из леса. (ДА: не привела; НЕТ: привела)

Д11. Удалить ложный коррелят из списка коррелятов.

Д12. Выбрать уникальную основу и ряд грамматических характеристик к данной основе. Проиндексировать тексты, то есть выбрать для построившейся тройки [основа, часть речи, парадигматический класс] коды текстов, в которых встретились словоформы, принадлежащие данной основе.

Д13. Применить метод распределения элементов пересеченных множеств коррелятов.

Несмотря на появление объемных лексиконов для многих европейских языков и все возрастающую популярность словарного анализа, системы морфологического анализа без словаря не теряют своего прикладного значения.

[8] Для Д5 - Д9 ДА: корреляция прошла успешно, то есть в деревьях словоформ были обнаружены ложные ветви и удалены; НЕТ: корреляция прошла неуспешно, то есть ложных ветвей не

В задачах автоматической индексации изложенные выше алгоритмы позволяют формировать грамматические словари, являющиеся точным отображением лексики проиндексированных документов. Бессловарная морфология сохраняет свою актуальность в задачах автоматического пополнения лексиконов. Точность такого анализа выше, чем стандартная процедура предсказания по конечной последовательности символов в слове [см. раздел II]. Использование деревьев для представление морфологической структуры словоформы и унификация гипотезы роднит задачи морфологического и синтаксического анализа, демонстрируя общность формализма и алгоритмических методов на разных уровнях лингвистического анализа. Тестирование программы, разработанной на основе полученной методики, показало работоспособность предложенной системы автоматической индексации. Метод корреляции, разработанный для настоящей задачи, позволяет выбирать уникальные гипотезы и строить словарь основ при сравнительно небольшой выборке.

II. Проектирование словарной морфологии.

Существует два базовых подхода к проектированию морфологических машинных словарей (лексиконов) для флективных языков. Первый копирует академическую лингвистическую модель описания, где выделяются основные парадигматические классы, соответствующие типу склонения и спряжения, и правила регулярных альтернаций (фонетических чередований), а нерегулярные формы (например, сильные глаголы в немецком и английском языках) задаются перечислением. Такого типа лексиконы для русского языка составляются на базе модели грамматического словаря А.Зализняка, разрабатывая 8 классов именного склонения и 16 глагольного спряжения, а чередования в основе и глагольной темы выносятся в отдельное множество пост-морфологических правил альтернаций. Второй подход рассматривает любого вида регулярное и нерегулярное чередование как часть расширенной псевдо-флексии (в таком случае, основа словоформы ‘день’ – ‘д’, а флексия – ‘-ень’; для словоформы ‘песок’: ‘пес’ и ‘-ок’). В подобной модели описания число парадигматических классов для русского языка возрастает до 3000, но рост числа классов при

обнаружено. Цифровые индексы на стрелках задают маршрут продвижения по схеме, то есть индекс стрелки выхода из блока Д10 должен совпасть с индексом стрелки входа.

проектировании компенсируется однородностью лексикона и отсутствием как исключений, так и правил альтернаций.

Внутреннее устройство лексиконов первого и второго типов не влияет ни на процесс лемматизации – приведение словоформы к нормальной форме слова, репрезентирующей лексему, ни на морфанализ – определение граммем словоформы. Анализаторы, построенные на разных типах лексиконов, могут одинаково эффективно использоваться как для морфологического анализа, так и для синтеза.

Первый подход к проектированию лексиконов для построения морфологических анализаторов европейских и восточных языков был применен в научно-исследовательском центре Xerox (Гренобль) в середине 90-ых, а позже усовершенствован и доведен до промышленного использования в исследовательских отделах Inxight Software (Санта-Клара, США и Антверпен, Бельгия) в 2000-2002 гг. Конечный продукт Inxight LinguistX Platform 3.5 включает в себя морфологии 26 языков: 5 восточных (арабский, корейский, японский, etc.) и 21 европейский (английский, голландский, испанский, русский, etc.). Наиболее разработанные языковые модули, такие как английский, немецкий и русский, имеют четыре уровня текстового анализа: tokenizer – графематика, осуществляющая деление исходного текста на предложения и словоформы; stemmer – лемматизация входных словоформ; tagger – снятие морфологической омонимии и унификация значений грамматических характеристик; и np-grouper – синтаксическое выделение именных составляющих NP из текстов.

Морфологии языков в LxPlatform состоят из двух компонент: (1) лексикон, в котором хранятся леммы (нормальная форма слова), а также парадигмы и значения их грамматических категорий; (2) множество правил альтернаций и орфографических правил. Лексикон состоит из подлексиконов (sublexicons), делящихся по селективным признакам и парадигматическим классам. Структура подлексиконов образует связанный граф, в вершине которого стоит корневой (root) лексикон, начинающий анализ входного слова [Lauri Karttunen, 1993]. Все правила второго компонента морфологии записываются на языке регулярных выражений [XRCE MLTT, 1995]. Технология анализа построена на разновидности конечных автоматов FST (finite-state transducer). FST называется автомат, в котором каждый переход

между состояниями в сети (network) имеет выходную помету в дополнение к входной [XRCE MLTT, 1995]. Исходный морфологический лексикон компилируется в lexicon transducer, а компонент правил - в two-level rule transducer. Результирующий лексический конечный автомат, т.е. полное морфологическое представление языка, - lexical transducer - получается композицией lexicon transducer и rule transducer [Lauri Karttunen, 1994]. Sigma называется символьный алфавит конечного автомата [Finite-State Network, 1995]. Sigma лексического FST состоит из алфавита анализируемого естественного языка и специальных грамматических помет (tags), выражающих значение селективных признаков и граммем (+Verb – глагол, +Active – активный залог, +P1 – 1-ое лицо, +Pl – множественное число, etc.).

Построим содержащий описание глаголов ‘вписывать’ и ‘восторжествовать’ фрагмент морфологического лексикона для русского языка:

```
LEXICON Root
        Nouns; Verbs;
LEXICON Nouns
        ……
LEXICON Verbs;
        водить+Verb+Imperf:во(д/ж)ь          V2;
        вписывать+Verb+Perf:впи(с/ш)         V1;
LEXICON V1;
        +Inf+Active:^Нать  #;
        +Imperf+Inf+Passive:^Наться  #;
        +Ind+NotPast+P1+Sg+Active:^Sьу  #;
        +Ind+NotPast+P2+Sg+Active:^Sьэшь  #;
        +Ind+NotPast+P3+Sg+Active:^Sьэт  #;
        +Ind+NotPast+P1+Pl+Active:^Sьэм  #;
        +Ind+Past+Sg+Masc+Active:^Нал  #;
        +Ind+Past+Sg+Fem+Active:^Нала  #;
        +Ind+Past+Sg+Neut+Active:^Нало  #;
        ……
LEXICON V2;
        +Inf+Active:^Ньыть  #;
        +Imperf+Inf+Passive:^Ньыться  #;
        +Ind+NotPast+P1+Sg+Active:^Sьу  #;
        +Ind+NotPast+P2+Sg+Active:^Ньышь  #;
        +Ind+NotPast+P3+Sg+Active:^Ньыт  #;
        +Ind+NotPast+P1+Pl+Active:^Ньым  #;
        +Ind+Past+Sg+Masc+Active:^Ньыл  #;
        +Ind+Past+Sg+Fem+Active:^Ньыла  #;
        +Ind+Past+Sg+Neut+Active:^Ньыло  #;
        ……
```

Корневой лексикон осуществляет вызов подлексиконов. Выражения в лексиконах представляют собой пару форм: лексическая (lexical) и поверхностная (surface) формы, разделенные двоеточием. Строящий FST компилятор интерпретирует такую пару как регулярное отношение. Решетка ‘#’ маркирует конечное состояние. Уникальность пути переходов в сети конечного автомата дает однозначность морфологической интерпретации. Приводящие в конечное состояние варианты пути в сети FST задают множественность интерпретаций для поверхностной формы, что соответствует морфологической омонимии. Так, поверхностная форма ‘стекло’ получит две лексических формы: ‘стекло+Noun…’ и ‘стекать+Verb…’. Так, лексическая форма ‘вписывать+Verb+Perf+Ind+NotPast+P3+Sg+Active’ соответствует поверхностной форме ‘впи(с/ш) ^Sьэт’.

Используя регулярные выражения [подробнее см. раздел IV], построим фрагмент компонента правил альтернации и орфографических правил для русского языка:

```
[ %(с%/ш%)->ш,
  %(д%/ж%)->ж,
|| _ ?* %^S ];

[ %(с%/ш%)->с,
  %(д%/ж%)->д,
|| _ ?* %^H ];

[%^H->0, %^S->0];

[ь ь -> ь, й ь -> й];

[ [ь|й]а->я, [ь|й]у->ю, [ь|й]э->е, [ь|й]ы->и ];
```

Процент ‘%’, поставленный перед символом, переводит зарезервированный оператор в языке регулярных выражений в простой символ алфавита. ‘?’ – любой (any) символ; ‘*’ – звезда Клини; бинарный оператор ‘A -> B’ осуществляет замещение последовательности символов в левой части выражения на последовательность в правой части; выражение ‘A -> B || L _ R’ является операцией замещения, ограниченной по контексту, т.е. строке A предшествует L, и R следует непосредственно за A; ‘|’ означает дизъюнкцию. Таким образом, после применения правил к строке ‘впи(с/ш) ^Sьэт’ поверхностная форма примет свой окончательный вид: ‘впишет’.

Для английского языка в LxPlatform был разработан лексикон, включающий информацию об активном словообразовании. Модуль деривации для русской морфологии в LxPlatform отсутствует. Реализация морфологического анализатора на основе технологии конечных автоматов позволяет достичь максимальной скорости анализа.

Морфологический лексикон проекта Диалинг спроектирован с использованием второго подхода, где основа представлена неизменяемой частью слова, а все регулярные и нерегулярные чередования (в том числе и в корневой морфеме) являются частью расширенной псевдо-флексии. Лексикон насчитывает свыше 3000 парадигматических классов. Идеология морфологического анализа заимствована из работ Ж.Г.Аношкиной [Ж.Г.Аношкина, 1995].

Описание парадигмы лексемы представляет собой множество пар [псевдо-флексия, набор аношкинских кодов]. Аношкинским кодом называется уникальный двухбуквенный идентификатор, который соответствует некоторой комбинации значений селективных признаков и граммем. Конечное множество аношкинских кодов исчисляет все встречающиеся в данном языке комбинации морфологических характеристик. Всего в морфологическом анализаторе русского языка системы Диалинг насчитывается 870 таких кодов.

Приведем фрагмент описания парадигмы для лексемы ‘рукоплескать’:

1740
%СКАТЬ*ка%СКАВШАЯ*мз%ЩУ*кб%ЩУТ*кж%ЩУЩЕГО*лблглп%….
……
РУКОПЛЕ 1740

‘Рукопле’ – основа слова в лексиконе; ‘1740’ – уникальный идентификатор парадигматического класса; ‘%’ маркирует начало псевдо-флексии; ‘*’ маркирует начало аношкинского кода; ‘ка’, ‘кб’, ‘лб’, ‘лг’, etc. – код. В таблице приведена расшифровка аношкинских кодов, использованных в примере:

код	часть речи	словообразовательные характеристики	словоизменительные характеристики	Пример
ка	Г	нс,нп	дст,инф	рукоплескать, расти
мз	Г	нс,нп,прч	прш,дст,ед,жр, им	рукоплескавшая, росшая
кб	Г	нс,нп	дст,нст,1л,ед	рукоплещу, расту
кж	Г	нс,нп	дст,нст,3л,мн	рукоплещут, растут
лб	Г	нс,нп,прч	нст,дст,ед,мр, рд	рукоплещущего, растущего
лг	Г	нс,нп,прч	нст,дст,ед,мр,	рукоплещущего,растущ

			вн	ий
лп	Г	нс,нп,прч	нст,дст,ед,ср,рд	рукоплещущего,растущего

Также в множество аношкинских кодов морфологического анализатора Диалинг включены специальные коды для аналитических форм глагола, которые строятся в системе на этапе синтаксического анализа. Комбинация значений морфологических характеристик для аналитической формы глагола получается путем объединения исходных характеристик всех составляющих аналитической формы. Примеры такого кода:

Ил	П	нс,нп,	буд,мр,3л,ед,кр	будет умен
Юа	Г	нс,пе,	дпр,нст,жр,ед,кр	будучи умна

Система морфологического анализа Информэлектро, разработанная в начале 70-ых гг. в секторе (затем отделе) Д.Г.Лахути группой лингвистов под руководством Г.А.Лесскиса, является одной из первых версий машинной морфологии. Морфологический лексикон Информэлектро также можно отнести к моделям второго типа. Не используя правил альтернаций, для лексем, имеющих более одной основы, в словарь вводятся все ее основы или словоформы так, чтобы минимальным числом единиц обеспечить идентификацию всей парадигмы данной лексемы. Например, для слова «станок» вводится основа ‘станк-’ и словоформа ‘станок’. Отличительной особенностью лексикона является примитивная модель управления, которая определяется в статьях лексикона для лексем, имеющих синтаксическое управление. Примитивная модель управления (ПМУ) может принимать следующие грамматические значения: (1) управление предлогом; (2) управление родительным падежом; (3) управление дательным падежом; (4) управление винительным падежом; (5) управление творительным падежом; (6) управление предложным падежом; (7) управление подчинительным союзом; (8) управление инфинитивом. Так, для глагола ‘увидеть’ ПМУ=4,7.

В морфологических анализаторах Диалинг и Информэлектро предсказание значений селективных признаков и граммем словоформ, ненайденных в словаре, устроено однотипно. Если входная словоформа не была найдена в словаре, то используется алгоритм предсказания, который ищет в словаре словоформу, максимально совпадающую с конца со входной словоформой [А. Сокирко, 2001] (так называемое предсказание по «хвостам»). Парадигма найденной словоформы используется как образец для создания

парадигмы входной словоформы. Необходимо отметить, что в анализаторе Информэлектро модель управления неопознанных словоформ также предсказывается по «хвостам».

Все три рассмотренные системы морфологий с использованием лексиконов демонстрируют сравнимые по скорости и точности анализа результаты.

III. Метод снятия морфологической омонимии (tagger).

Еще в начале 60-ых годов американский лингвист Ч. Хоккеттом указал на возможность использования конечной марковской цепи в качестве модели для описания процесса синтаксического анализа, возникающего в голове слушающего после восприятия каждого последующего слова, произнесенного говорящим, в предложении [Ч. Хоккетт, 1961]. В компьютерной лингвистике скрытые марковские модели нашли свое применение в задачах разрешения омонимии словоформы по синтаксическому контексту в предложении.

Входными данными модуля tagger в LxPlatform служат результаты графематического и морфологического анализов, полученных модулями tokenizer и stemmer. Tagger представляет собой скрытую марковскую модель, способную запоминать последовательности длиной от 4 до 6 синтаксических единиц. Коэффициенты вероятностей выбора морфологических значений вырабатываются в цепи путем обучения марковской модели на размеченном тексте. Каждой словоформе в размеченном тексте присваивается морфологическая помета (tag). Для того, чтобы сократить размеры как самой скрытой модели, так и размеченного текста, необходимого для обучения, используются усеченные морфологические пометы, которые позволяют сократить комбинаторно возможные варианты синтаксических контекстов. Так, полная морфологическая помета словоформы ‘красивому’ ‘+Adj+Plain+Sg+MascNeut+Dat’ будет усечена в tagger до пометы ‘Adj-Obl’, такую же помету получат и другие формы прилагательного ‘красивый’, стоящие в косвенных падежах. Все финитные формы глагола используют единую помету Verb-Fin. В таблице перечислены все морфологические пометы, составляющие алфавит марковской модели для русского языка:

Помета	Описание	Примеры

Adj-Nom	Прилагательное в номинативе	красивый, красивая, красивое, красивые
Adj-Acc	Прилагательное в аккузативе	красивого, красивую, красивое, красивые
Adj-Gen	Прилагательное в генитиве	красивого, красивой, красивых
Adj-Obl	Прилагательное в косвенном падеже	красивым, красивой, красивому, красивыми
Adj-Comp	Сравнительная степень прил.	краше
Adj-Brf	Краткая форма прил.	красив, красива, красиво, красивы
Adv	Наречие	быстро
Conj	Союз	и, но, чтобы
Det-Nom	Местоименное прил. в номинативе	этот
Det-Acc	Местоименное прил. в аккузативе	эту
Det-Gen	Местоименное прил. в генитиве	этого
Det-Obl	Местоименное прил. в косвенном падеже	этому
Dig	Цифровой комплекс	1999, 100Мб
Pron-IntRel-Nom	Относительные местоимения в номинативе	кто
Pron-IntRel-Acc	Относительные местоимения в аккузативе	кого, что
Pron-IntRel-Gen	Относительные местоимения в генитиве	кого, чего
Pron-IntRel-Obl	Относительные местоимения в косвенном падеже	кому
Interj	Междометие	ага, ах, ба
Nn-Nom	Существительное в номинативе	сестра, сестры
Nn-Acc	Существительное в аккузативе	сестру, сестер
Nn-Gen	Существительное в генитиве	сестры, сестер
Nn-Obl	Существительное в косвенном падеже	сестрой, сестрами
Num	Числительное	три, восемь
Ord	Цифра	7., 3.
Pron-Pers-Nom	Личное местоимение в номинативе	я, ты
Pron-Pers-Acc	Лич. местоим. в аккузативе	меня, тебя
Pron-Pers-Gen	Лич. местоим. в генитиве	меня, тебя
Pron-Pers-Obl	Лич. местоим. в косвенном падеже	мной, тобой
Prep-Nom	Управляющий номинативом предлог	плюс, минус
Prep-Acc	Управляющий аккузативом предлог	за
Prep-Gen	Управляющий генитивом предлог	без, накануне
Prep-Obl	Управляющий косвенным падежом предлог	благодаря, к
Pron-Nom	Местоимение в номинативе	все, ничто
Pron-Acc	Местоимение в аккузативе	все, ничто
Pron-Gen	Местоимение в генитиве	всего, ничего
Pron-Obl	Местоимение в косвенном падеже	всеми, ничем
Prop-Nom	Имя собственное в номинативе	Москва, Мальцев

Prop-Acc	Имя собственное в номинативе	Москву, Мальцева
Prop-Gen	Имя собственное в генитиве	Москвы, Мальцева
Prop-Obl	Имя собственное в косвенном падеже	Москве, Мальцеве
Part	Частица	аж, же
Part-Int	Вводное	авось, конечно
Part-Sent	Предикатив	аминь
Aux	Вспомогательный глагол	быть
Verb-Fin	Финитная форма глагола	делай, делает, делал
Verb-Ger	Деепричастие	делав, делавши, делая
Verb-Inf	Инфинитив	делать
Verb-Acc	Причастие в аккузативе	делавшего, делавшее, делавшую
Verb-Gen	Причастие в генитиве	делавшего, делавшей
Verb-Nom	Причастие в номинативе	делавший, делавшее, делавшая
Verb-Obl	Причастие в косвенном падеже	делавшим, делавшей
Verb-Brf	Краткое причастие	делан, делано, делана

С уменьшением числа морфологических помет понижается и точность синтаксического контекста, а вместе с ним и анализа. Такая вероятностно-статистическая модель, учитывающая синтаксический контекст, косвенно лишена проверки полного согласования. Но экспериментальные данные доказывают, что даже такого числа усеченных помет достаточно для 95% точности при выборе леммы и грамматического значения словоформы, т.е. минимальный объем модели позволяет с высокой точностью снимать морфологическую омонимию. Действительно, обучение скрытой марковской модели на размеченном приведенными в таблице пометами тексте, размер которого не превышает 300 Кб, позволяет вычислять ожидаемые вероятностные коэффициенты для выбора правильного грамматического значения в простых и частотных случаях контекстного распределения.

Приведем результаты анализа модулями stemmer и tagger двух пар предложений, содержащих омонимичные словоформы, принимающие разные леммы и грамматические значения в зависимости от контекстного распределения.

Исходный текст:

На завод привезли стекло.
Масло стекло на пол.

Данные эксперименты являются ошибочными.
Последние данные являются ошибочными.

Результат лемматизации stemmer:

На на

завод завод
привезли привозить

```
стекло   стекло | стекать
.        .
Масло    масло
стекло   стекло | стекать
на       на
пол      пол | пола | полый
.        .
Данные           давать | данные | данный
эксперименты     эксперимент
являются         являть | являться
ошибочными       ошибочный
.        .
Последние        последние | последний
данные           давать | данные | данный
являются         являть | являться
ошибочными       ошибочный
.        .
```

Результат выбора значений tagger:

```
На          [Prep-Acc]      на
завод       [Nn-Acc]        завод
привезли            [Verb-Fin]      привозить
стекло      [Nn-Acc]        стекло
.           [Punct-Sent]    .
Масло       [Nn-Nom]        масло
стекло      [Verb-Fin]      стекать
на          [Prep-Acc]      на
пол         [Nn-Acc]        пол
.           [Punct-Sent]    .
Данные      [Adj-Nom]       данный
эксперименты        [Nn-Nom]        эксперимент
являются            [Verb-Fin]      являть | являться
ошибочными          [Adj-Obl]       ошибочный
.           [Punct-Sent]    .
Последние           [Adj-Nom]       последний
данные      [Nn-Nom]        данные
являются            [Verb-Fin]      являть | являться
ошибочными          [Adj-Obl]       ошибочный
.           [Punct-Sent]    .
```

Метод снятия омонимии, основанный на скрытой марковской цепи, - редкий случай, когда вероятностно-статистическая модель эффективно работает в лингвистике.

IV. Методика выделения именных групп (np-grouper).

Язык регулярных выражений – формальный язык, во многом схожий с формулами булевой логики. Он обладает простым синтаксисом, но выражения могут быть произвольно сложными. Каждое выражение обозначает множество. Позволяя создавать гибкие образцы (шаблоны) для любых последовательностей элементов, язык регулярных выражений широко применяется для быстрого поиска подстрок и обработки нечетких запросов. Регулярные выражения

компилируются в конечные автоматы, что позволяет достигать высокой скорости при поиске шаблонов.

Модуль np-grouper в LxPlatform предназначен для выделения из предложений именных составляющих NP. Фактически, np-grouper можно считать начальным этапом синтаксического анализа предложения. Такая технология используется в задачах автоматической обработки текстов (автоматическое построении таксономии и классификация информационного потока) с последующим статистическим анализом найденных NP. Для создания образцов NP, последовательностей элементов внутри группы, используется язык регулярных выражений, где каждое выражение представляет собой грамматический образ некоторой именной группы или ее подгруппы. Множество таких выражений составляет грамматику именных групп.

Регулярное выражение заключено в квадратные скобки '[…]'. Определение '**define** *name* […] ' присваивает уникальное имя выражению. В круглые скобки '(…)' заключается факультативная последовательность элементов внутри выражения. Символ '?' означает любой (any) символ. Унарные операторы языка: '*' – звезда Клини; '+' – плюс Клини. Бинарные операторы языка: пробел между двумя элементами означает их конкатенацию; '|' означает дизъюнкцию.

Построим грамматику именных групп для русского, используя морфологические пометы, введенные в предыдущем разделе, которые приобретают характер синтаксических элементов в нашей грамматике.

```
# определим подгруппы полных прилагательных и причастий, модифицированных наречием

define ANOM [Adj%-Nom | Verb%-Nom];
define AACC [Adj%-Acc | Verb%-Acc];
define AGEN [Adj%-Gen | Verb%-Gen];
define AOBL [Adj%-Obl | Verb%-Obl];

define AdjPNom [ ANOM+ [[Conj | Punct%-Comma] (Adv) ANOM]* ];
define AdjPAcc [ AACC+ [[Conj | Punct%-Comma] (Adv) AACC]* ];
define AdjPGen [ AGEN+ [[Conj | Punct%-Comma] (Adv) AGEN]* ];
define AdjPObl [ AOBL+ [[Conj | Punct%-Comma] (Adv) AOBL]* ];

# объединим существительные и имена собственные для каждого падежа соответственно

define NNNOM [Nn%-Nom | Prop%-Nom];
define NNACC [Nn%-Acc | Prop%-Acc];
define NNGEN [Nn%-Gen | Prop%-Gen];
define NNOBL [Nn%-Obl | Prop%-Obl];

# определим числительные
```

```
define NUMBER [Num+ | Dig];

# определим пре-модифицированные именные группы

define ANPNom [ (AdjPNom) NNNOM ];
define ANPAcc [ (AdjPAcc) NNACC ];
define ANPGen [ (NUMBER (Adv)) (AdjPGen) NNGEN ];
define ANPGen1 [ (Det%-Gen) (Adv) (AdjPGen) NNGEN ];
define ANPObl [ (AdjPObl) NNOBL ];

# определим однородные именные группы

# определим однородные пары именных групп, соединенных сочинительным союзом

define HOMOGENPairNom [ ANPNom Conj (Adv) ANPNom ANPGen1* ];
define HOMOGENPairAcc [ ANPAcc Conj (Adv) ANPAcc ANPGen1* ];
define HOMOGENPairGen [ ANPGen Conj ANPGen1+ ];
define HOMOGENPairObl [ ANPObl Conj (Adv) ANPObl ANPGen1* ];

# определим однородные цепочки именных групп длиной от 4 элементов и больше.
# определение трехсоставных однородных цепочек может повлечь
# высокий процент ошибочно построенных однородных групп

define HOMOGENNom [ANPNom [Punct%-Comma (Adv) ANPNom]+ Punct%-Comma (Adv)
HOMOGENPairNom];
define HOMOGENAcc [ANPAcc [Punct%-Comma (Adv) ANPAcc]+ Punct%-Comma (Adv)
HOMOGENPairAcc];
define HOMOGENGen [ ANPGen [Punct%-Comma ANPGen1]+ Punct%-Comma (Adv)
HOMOGENPairGen];
define HOMOGENObl [ ANPObl [Punct%-Comma (Adv) ANPObl]+ Punct%-Comma (Adv)
HOMOGENPairObl];

# определим пост-модификацию именной группы существительным в генитиве

define NNS [ ANPNom | ANPAcc | ANPGen | ANPObl ];

define NNS2 [ [(AdjPNom) Nn%-Nom] | [(AdjPAcc) Nn%-Acc] | [(NUMBER (Adv)) (AdjPGen)
Nn%-Gen] | [(AdjPObl) Nn%-Obl] ];

define NPGENIT [ NNS2 ANPGen1+ (Conj ANPGen1) ];

define NUMBERNP [ NUMBER Nn%-Gen NUMBER Nn%-Gen ];

# определим пост-модификацию именной группы прилагательным

define ANPPOSTMNom [ NNNOM Adj%-Nom ];
define ANPPOSTMAcc [ NNACC Adj%-Acc ];
define ANPPOSTMObl [ NNOBL Adj%-Obl ];

# определим именную группу состоящую из цепочки существительных и имен собственных

define NPPROPNOM [ NNNOM [Prop%-Nom]+ ];
define NPPROPACC [ NNACC [Prop%-Acc]+ ];
define NPPROPGEN [ NNGEN [Prop%-Gen]+ ];
define NPPROPOBL [ NNOBL [Prop%-Obl]+ ];

# определим разные классы NP

define NPHOMOGENPAIR [ HOMOGENPairNom | HOMOGENPairAcc | HOMOGENPairGen |
HOMOGENPairObl ];
```

```
define NPHOMOGEN [ HOMOGENNom | HOMOGENAcc | HOMOGENGen | HOMOGENObl ];

define NPPOSTM [ ANPPOSTMNom | ANPPOSTMAcc | ANPPOSTMObl ];

define NPPROP [ NPPROPNOM | NPPROPACC | NPPROPGEN | NPPROPOBL ];

# определим NP образец

define NPS [
NNS |
NPHOMOGENPAIR |
NPHOMOGEN |
NPGENIT |
NUMBERNP |
NPPOSTM |
NPPROP
];
```

Таким образом, такая грамматика именных групп способна определять именные составляющие NP следующих типов: (а) существительные, пре-модифицированные прилагательным, причастием или числительным (‘очень разумная идея’, ‘белый, красный и зеленый шар’, ‘два стола’); (б) сочиненные именные группы (‘брат и сестра’, ‘низкий стол, стул, широкий табурет и шкаф’); (в) генитивные группы (‘рука власти’, ‘Министерство Финансов’); (г) цепочки существительных (‘город Москва’, ‘Сергей Петрович Иванов’); (д) определение в постпозиции, выраженное прилагательным (‘впечатление необычное’).

Основным недостатком такого формализма является невозможность описания разрывных составляющих на языке регулярных выражений. Усеченные морфологические пометы лишают возможности проверки полного согласования, оставляя только частичное падежное согласование в грамматике. Подобная модель шаблонов именных групп не способна выделять два типа NP, определенных для русского языка: необособленное согласованное определение и пост-модификация именной группы, выраженная предложной группой.

Экспериментальные данные и проведенное тестирование модуля np-grouper доказывает работоспособность методики и относительно высокую точность (не менее 98%) построения NP. Достоинством грамматики именных групп, сформулированной на языке регулярных выражений, является ее краткость и прозрачность.

Все приведенные в настоящей главе морфологические и предсинтаксические компоненты анализа потенциально являются неотъемлемой частью идеальной модели полного синтаксического процессора, а также

позволяют демонстрировать общность формализма и методов решения задач на разных уровнях лингвистического анализа.

ГЛАВА 3. СЕГМЕНТАЦИОННЫЙ АНАЛИЗ РУССКОГО ПРЕДЛОЖЕНИЯ

I. Поверхностный синтаксический процессор группы Диалинг.

Введение

Поверхностный синтаксический процессор русского языка разработан группой Диалинг[9] [А. Сокирко, 2001] в период с 1998-2001 гг. Фундаментом для исследований группы ДИАЛИНГ послужила система французско-русского автоматического перевода (ФРАП), разработанная в ВЦП совместно с МГПИИЯ им. М. Тореза в 1976-86 гг., и система анализа политических текстов (ПОЛИТЕКСТ), разработанная в Центре информационных исследований совместно с ВЦ ИСК РАН в 1991-97 гг [Н. Леонтьева, 1995].

Так же как и STP, процессор относится к алгоритмическим системам модульного типа. На вход синтаксическому анализатору подаются результаты работы графематики и морфологии, где каждая словоформа предложения представлена множеством морфологических омонимов. Принципиальным отличием в архитектуре анализатора можно считать двунаправленное взаимодействие модуля сегментации[10] и синтаксиса (т.е. построения синтаксических групп слов в предложении) [Д. Панкратов и др., 2000]. В случае STP такое взаимодействие было однонаправленным: топологическая структура с результатами сегментации поступала на вход грамматического модуля, отвечающего за выделение фразовых категорий. В процессоре Диалинг два модуля работают параллельно, чередуясь и обмениваясь накопленными знаниями (сегментация $\Leftrightarrow$ синтаксис).

[9] В разное время над созданием процессора работали А. Сокирко, Д. Панкратов, Л. Гершензон, Т. Кобзарева, И. Ножов.
[10] Фрагментация в терминах группы Диалинг (www.aot.ru).

Общая схема действий анализа

Общую схему действий в анализаторе можно представить в виде последовательности шагов:

1. Членение предложения по знакам пунктуации и сочинительным союзам на исходные отрезки; будем их также называть начальными сегментами. Объединение исходных отрезков с простыми случаями однородных рядов прилагательных, наречий, существительных, etc. Определение вершин и типов начальных сегментов.
2. Построение аналитических форм глагола внутри исходных отрезков.
3. Выделение именных групп терминов внутри исходных отрезков с помощью тезаурусов: общего, компьютерного и финансового.
4. Интерпретация вершин начальных сегментов, содержащих тире, и попытка восстановления тире в отрезках с нулевым Copul.
5. Декартово произведение омонимов внутри начальных сегментов – построение множества однозначных морфологических интерпретаций (МИ) одного сегмента. Построение простых синтаксических групп для каждой МИ сегмента подмножеством синтаксических правил: КОЛИЧ (“двадцать восемь”), ПРИЛ-СУЩ (“большой дом”), ПГ (“на столе”), ОДНОР_ПРИЛ (“умный и обаятельный”), etc.
6. Правила для объединения сочиненных начальных сегментов.
7. Построение сочиненных синтаксических групп для каждой МИ подмножеством синтаксических правил: ОДНОР_ИГ (“сын своего отца и дочь своей матери”), Р_С_ОДНОР_СУЩ (“как новые книги, так и пыльные папки”), Р_С_ОДНОР_ИНФ (“если не писать, так читать”), etc.
8. Правила для вложения контактно расположенных сегментов (причастных, деепричастных, придаточных определительных, etc.) – установление иерархии на сегментах.
9. Построение групп подмножеством синтаксических правил для каждой МИ, где вершина – синтаксическая группа подчиняющего сегмента, а зависимое – вершина вложенного сегмента.
10. Правила для объединения разрывных сегментов. Завершение полной сегментации предложения.

11. Построение групп множеством всех синтаксических правил для каждой МИ. Всего в модели процессора используется 40 типов групп.
12. Оценка синтаксического покрытия каждой МИ (лучший вариант получает максимальный вес) и установление порядка на множестве морфологических интерпретаций каждого сегмента.

Таким образом, расширение границ сегментов происходит последовательно с постепенным накоплением синтаксической информации внутри каждого начального сегмента. В свою очередь, каждая новая итерация синтаксических правил пытается расширять границы синтаксических групп, вкладывая уже построенные ранее группы в новые составляющие.

Морфологические интерпретации

МИ сегмента являются результатом декартова произведения омонимов словоформ, входящих в данный отрезок. Сегментация позволяет избежать полного декартова произведения предложения, что значительно сокращает число рассматриваемых вариантов. В словосочетании “рабочие стали” (рабочие: рабочий.N и рабочий.Adj; стали: сталь.N и стать.V) строится четыре однозначных МИ, для каждой из которых будет построена своя структура групп. Лучший вес получают МИ с максимальным количеством синтаксических групп. Множества синтаксических групп двух МИ, как правило, имеют непустое пересечение. Покрытием МИ называется число словоформ, вошедших в синтаксические группы данной МИ. Построим пример анализа отрезка (‘масса рабочего стекла’), демонстрирующего отличие синтаксической структуры групп на уровне разных МИ. Для данного отрезка строится четыре МИ (рабочего: рабочий.N и рабочий.Adj; стекла: стекло.N и стечь.V). Рассмотрим две из них, синтаксически равноправных и имеющих одинаковое покрытие (рис.1):

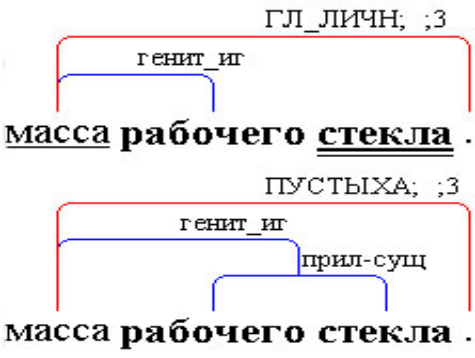


рис.1

Первая МИ задается комбинацией ‘рабочий.N’ и ‘стечь.V’, вторая – ‘рабочий.Adj’ и ‘стекло.N’. Структура групп первой МИ ПОДЛ(‘стекла’, ГЕНИТ_ИГ(‘масса’, ‘рабочего’)), второй - ГЕНИТ_ИГ(‘масса’, ПРИЛ_СУЩ (‘стекла’, ‘рабочего’)).

Внутрисегментный анализ

Внутрисегментный анализ в процессоре Диалинг – это синтаксис неразрывных групп, определенный в работе Сущанской как «синтаксис первого ранга» [Н. Сущанская, 1989]. Для синтаксической группы определены: (а) координаты (номер первого и последнего слова); (б) тип группы (ПРИЛ-СУЩ, ПГ, ОДНОР_ИГ, ПРЯМ_ДОП, etc.), который задается правилом; (в) координаты главной подгруппы группы. Например, группа “высокий дом” имеет соответствующие параметры: (а) {1, 2}; (б) ПРИЛ-СУЩ; (в) {2, 2}. Каждый тип группы строится отдельным правилом, которое имеет направление поиска зависимого и определяет вершину группы. Используя параметры группы, легко вычислить синтаксически главное слово, что позволяет переводить структуру вложенных групп в размеченное дерево зависимостей. Порядок применения синтаксических правил жестко задан как на отдельных подмножествах правил (шаги 5, 7, 9 в схеме), так и на множестве всех правил (шаг 11 в схеме). Так, правило ПРИЛ-СУЩ (согласованные прилагательное + существительное) должно отрабатывать раньше правила ГЕНИТ_ИГ (существительное + существительное в родительном): “высокий дом отца” ГЕНИТ_ИГ(ПРИЛ-СУЩ(1, 2), 3) = {1, 2} → {3, 3} и {2, 2} → {1, 1}, обратный порядок применения правил дает неправильную иерархию групп *ПРИЛ-СУЩ(1, ГЕНИТ_ИГ(2, 3)) = {2, 3} → {1, 1} и {2, 2} → {3, 3}. Подключение тезаурусов на первоначальном этапе синтаксического анализа (этап 3 в схеме) позволяет сохранять синтаксически правильную иерархию групп в тех случаях, когда устойчивые словосочетания входят в состав предложения. Для каждого текстового входа в тезаурусе определена синтаксическая модель, включающая в себя набор синтаксических групп, которые должен построить процессор [А. Сокирко, 2001]. Так, для отрезка предложения “высокий бюджет государства” на этапе 3 синтаксическим анализатором, при обращении к финансовому тезаурусу, будет обнаружено словосочетание ‘бюджет государства’, с приписанной к нему группой ГЕНИТ_ИГ; тогда для данного отрезка результирующая структура

анализа выглядит как ПРИЛ-СУЩ(1, ГЕНИТ_ИГ(2, 3)), что было запрещено порядком применения правил в предыдущем примере. Синтаксические правила не используют модели управления слов (что неминуемо привело бы к построению разрывных групп и противоречит синтаксису первого ранга), кроме тех случаев, когда управление напрямую вычисляется по морфологическим характеристикам слова. Единственное исключение в процессоре делается для группы подлежащее-сказуемое, которая в большинстве случаев является разрывной составляющей. Анализатор не ставит цели построить полную синтаксическую структуру внутри сегмента.

Синтаксические группы

В нижеследующей таблице приведен полный перечень типов синтаксических групп, определенных в процессоре. Каждый тип группы в синтаксическом анализе строится отдельным правилом, способным учитывать грамматическое согласование, предложное управление и линейный порядок подгрупп в сегменте предложения.

Тип	**Название**	**Пример**
Количественная группа (последовательность числительных)	КОЛИЧ	Двадцать восемь
Последовательность чисел	СЛОЖ-ЧИСЛ	12,3, II-III
Группа существительного, пре-модифицированная одним или несколькими прилагательными	ПРИЛ-СУЩ	Длинная тяжелая дорога, двигающийся человек
Группа существительного, пре-модифицированная наречным числительным	НАР-ЧИСЛ-СУЩ	Много ребят, мало стульев
Группа существительного, пре-модифицированная числительным	СУЩ-ЧИСЛ	Восемь попугаев, два человека
Предложная группа	ПГ	В дом, на холме
Группа однородных прилагательных	ОДНОР_ПРИЛ	смелый, красивый и умный
Глагол, пре-модифицированный наречием	НАРЕЧ_ГЛАГОЛ	злостно нарушает, тяжело жить
Полное или краткое прилагательное, пре-модифицированное наречием	НАР_ПРИЛ	очень красивый, весьма полезный, особенно хорош.
Цепочка наречий	НАР_НАР	как легко, так интересно
Аналитическая форма сравнительной степени прилагательного или наречия	СРАВН-СТЕПЕНЬ	гораздо сильнее; значительно больше
Отрицательная частица ‘не’ + глагол	ОТР_ФОРМА	Не любить; не знать
Группа контактно расположенного справа прямого дополнения	ПРЯМ_ДОП	Рубить дрова; смотреть фильм
Генитивное определение в постпозиции	ГЕНИТ_ИГ	Рука человека; стол отца; набор грузов
Группа однородных наречий	ОДНОР_НАР	Очень и так
Группа глагола контактно справа	ПЕР_ГЛАГ_ИНФ	пойти выпить; позвать гулять.

пост-модифицированного инфинитивом		
Группа однородных инфинитивов	ОДНОР_ИНФ	гулять, думать и говорить
Группа имя + фамилия	ФИО	Владимир Набоков
Группа однородных именных групп	ОДНОР_ИГ	красивый дом и густой лес
Прилагательное, пре-модифицированное ‘такой’ или ‘самый’	МОДИФ_ПРИЛ	такая красивая
Группа существительного, пост-модифицированная причастным оборотом (сегментом)	ПРИЧ_СУЩ	Дом, построенный …
Группа подлежащее-сказуемое	ПОДЛ	Человек идет
Группа приложения	ПРИЛОЖЕНИЕ	Его отца, очень обидчивого человека, эта реплика вывела из себя. (отец -> человек)
Группа существительного, пост-модифицированная группой обособленного прилагательного	СУЩ_ОБС_ПРИЛ	сестра, совсем больная,… (сестра -> больная)
Группа однородных наречий, Р_С: сочиненных повторяющимися или разрывными союзами	Р_С_ОДНОР_НАР	не только вчера, но и сегодня
Группа однородных Р_С прилагательных	Р_С_ОДНОР_ПРИЛ	хотя и очень больной, но довольно сильный
Группа однородных Р_С причастий	Р_С_ОДНОР_ПРИЧ	как работающий, так и преуспевющий
Группа однородных Р_С сущ-ных	Р_С_ОДНОР_СУЩ	как книги, так и папки
Группа однородных Р_С мест-ний	Р_С_ОДНОР_МС	ни он, ни она
Группа однородных Р_С инфинитивов	Р_С_ОДНОР_ИНФ	если не писать, так читать
Группа однородных Р_С деепричастий	Р_С_ОДНОР_ДЕЕПР	если не думая, то говоря
Предикатив, пре-модифицированный наречием	НАР_ПРЕДИК	очень интересно
Группа сущ-ного, пост-модифицированного необособленным прил.	ПРИЛ_ПОСТПОЗ	впечатление необычное
Прил., пре-модифицированное ‘более’ или ‘менее’	АНАТ_СРАВН	более сильный, менее привлекателен
Группа однородных Р_С предложных групп	Р_С_ПГ	как на шкафу, так и в столе
Конструкция ‘каждый’ или ‘один’ + ПГ с предлогом ‘из’	ЭЛЕКТ_ИГ	Один из них, каждый из ваших людей
Формат электронного адреса	ЭЛ_АДРЕС	www.aot.ru
Сравнительное степень прил. + именная группа в генитиве	ОТСРАВН	левее сапога, умнее человека

Структура сегмента

Для каждого сегмента определены: (а) координаты (номера слов в предложении, соответствующих левой и правой границе сегмента); (б) вершина сегмента: номер слова и тип вершины $h \in H = \{$ ГЛ_ЛИЧН (глагол в личной форме), КР_ПРЧ (краткое причастие), КР_ПРИЛ (краткое прилагательное), ПРЕДК (предикативное слово), ПРИЧ (причастие), ДПР (деепричастие), ИНФ (инфинитив), ВВОДН (вводное слово), ПУСТЫХА $\}$, где ПУСТЫХА означает

пустую вершину, а все типы в H иерархически упорядочены; (в) союз или союзное слово. Тип вершины сегмента задается по значению селективного признака вершины и может быть представлен в виде множества значений, соответствующих допустимым значениям морфологических омонимов. В сегменте “когда ему весело” (весело: веселый (кр. прил.), весело (предикатив), весело (наречие)) определены соответствующие параметры: (а) {1, 3}; (б) [3] и { КР_ПРИЛ, ПРЕДК, ПУСТЫХА }; (в) ‘когда’.

Порядок выполнения сегментационных правил (отраженный в этапах 1, 6, 8 и 10 схемы) определен последовательностью работы их подмножеств, внутри этих блоков правил порядок выполнения – свободный. В процессоре определено три операции на сегментах: объединение, вложение и деление. Каждое правило, фактически, является проверкой ряда условий на возможность применения той или иной операции над сегментами.

Операция объединения сегментов

Помимо условий, накладываемых на вершины двух претендующих на объединение сегментов, основным критерием объединения служит возможность построения группы (однородных именных групп, групп с разрывными союзами или группы подлежащего-сказуемого) на границе двух сегментов. Получившейся в результате объединения сегмент наследует вершину того начального сегмента, чей тип вершины находится выше в иерархии H. Морфологические интерпретации нового сегмента получаются путем перемножения МИ объединившихся начальных сегментов. Пример операции объединения (рис. 2)[11]:

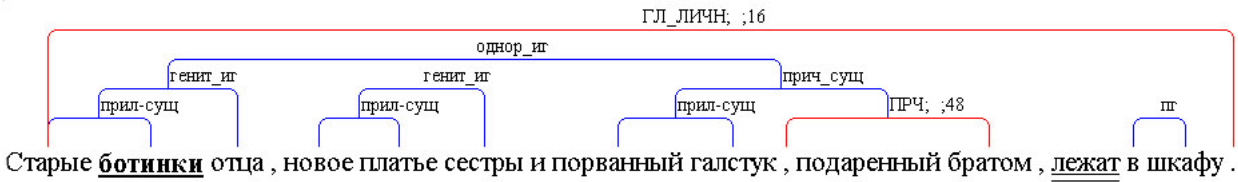


рис.2

Операция вложения сегментов

Вложение одного сегмента в другой может быть как согласованным, так

и произвольным. Правила, отвечающие за согласованное вложение сегмента, ищут вершину придаточного определительного или причастного оборота в левостоящем сегменте. Для выполнения операции вложения вводится специальное понятие – юнит (unit) [Д. Панкратов и др., 2000]. Юнитом в сегменте может быть либо слово, представленное множеством своих омонимов, либо вложенный сегмент, представленный множеством омонимов своей вершины. Тогда морфологическая интерпретация (МИ) сегмента есть линейная последовательность омонимов его юнитов. Понятие юнит позволяет строить синтаксические группы для вложенных сегментов. Так, на рис.2 построена группа ПРИЧ_СУЩ, где главной подгруппой является ПРИЛ_СУЩ(порванный, галстук), а зависимой – вложенный сегмент с вершиной ‘подаренный’. Произвольное вложение сегментов происходит в случае объединения двух дистантно расположенных исходных отрезков, тогда все сегменты, находящиеся между ними, вкладываются в новый сегмент, полученный в результате объединения. Пример произвольного вложения (рис. 3):

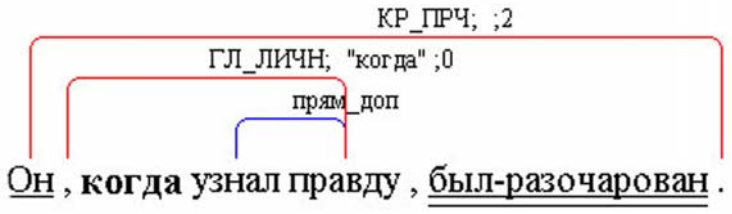


рис.3

При вложении одного сегмента в другой МИ подчиняющего сегмента умножаются на омонимы вершины подчиняемого сегмента с сохранением в МИ ранее построенных синтаксических групп.

Операция деления сегментов

Операция деления сегмента выполняется только для одного правила: выделения необособленного согласованного определения (НСО). Не имея границ, выраженных знаками препинания, НСО вычленяется алгоритмически в отдельный подсегмент. Пример деления (рис. 4):

[11] Использован графический интерфейс процессора Диалинг.

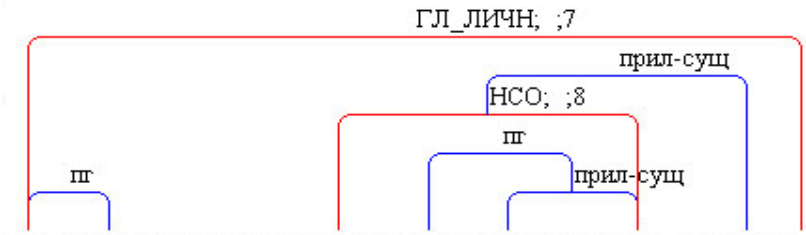


рис.4

Преобразование групп в бинарные отношения

Возможность вычисления синтаксически главного слова для каждой группы позволяет преобразовывать иерархическую структуру групп в множество бинарных отношений. Вычисление лексической вершины сегмента также позволяет переводить иерархическую структуру вложений сегментов во множество бинарных связей, где синтаксически главным словом является вершина вышестоящего сегмента, а зависимым – вершина непосредственно вложенного в него сегмента. Так, синтаксическая структура предложения

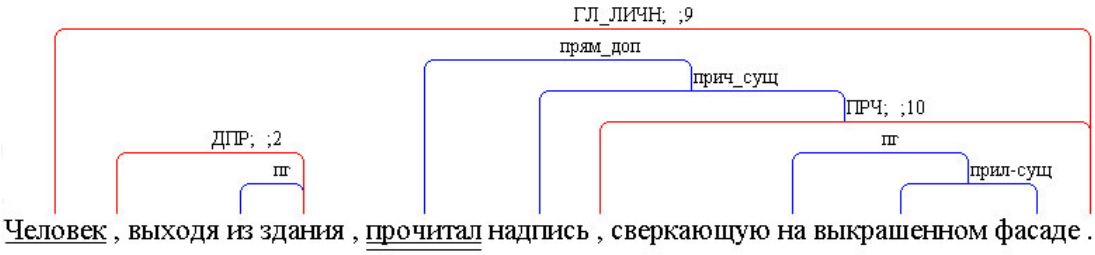


рис.5

автоматически преобразуется в множество бинарных отношений:

ИЗ [ПРЕДЛ][] --- ЗДАНИЕ [С,ср,но][ед,рд] : ПГ;

ФАСАД [С,мр,но][ед,пр] --- ВЫКРАСИТЬ [ПРИЧАСТИЕ,св,пе][мр,ср,ед,пр,стр,прш] : ПРИЛ-СУЩ;
НА [ПРЕДЛ][] --- ФАСАД [С,мр,но][ед,пр] : ПГ;
НАДПИСЬ [С,жр,но][ед,им,вн] --- СВЕРКАТЬ [ПРИЧАСТИЕ,нс,нп][жр,ед,вн,дст,нст] : ПРИЧ_СУЩ;
ПРОЧИТАТЬ [Г,св,пе][мр,ед,дст,прш] --- НАДПИСЬ [С,жр,но][ед,им,вн] : ПРЯМ_ДОП;
ПРОЧИТАТЬ [Г,св,пе][мр,ед,дст,прш] --- ЧЕЛОВЕК [С,мр,од][мн,ед,им,рд] : ПОДЛ;
ПРОЧИТАТЬ [Г,св,пе][мр,ед,дст,прш] --- ВЫХОДИТЬ [ДЕЕПРИЧАСТИЕ,нс,нп][дст,нст] : ДЕЕПР;

где в левой части находится главное солво, а в правой – зависимое; все словоформы внутри связей лемматизированы; для каждой леммы внутри первых квадратных скобок выведены значения ее селективных признаков, во

вторых квадратных скобках указаны ее граммемы в анализируемом предложении; двоеточием отделяется тип связи, в большинстве случаев совпадающий с типом синтаксической группы.

Заключение

Алгоритмическая прозрачность (rule-based подход) процессора Диалинг, к сожалению, приводит к большим вычислительным затратам. Недостатком системы является отсутствие синтаксической омонимии как на уровне иерархии и границ сегментов, так и на уровне групп; также в анализаторе не обрабатываются случаи сочинения предикатов, что служит иногда препятствием для объединения исходных отрезков и приводит к ошибочному построению сегментационной структуры.

Программная реализация процессора выполнена на языке C++. Взаимодействие между графематическим, морфологическим и синтаксическим модулями в программе организовано через стандартный COM интерфейс, результаты синтаксического анализа также доступны внешним приложениям через COM интерфейс. Неоспоримым достоинством процессора Диалинг является его завершенность: программная реализация доведена до уровня промышленного использования, - система характеризуется приемлемой скоростью анализа и устойчивостью на открытом пространстве реальных текстов.

II. Сегментационный процессор группы ОИС.

Введение

Синтаксический анализатор научный группы отделения интеллектуальных систем (ОИС) Института Лингвистики РГГУ (Д.Г. Лахути, Т.Ю. Кобзарева, И.М. Ножов) был создан в 1999-2003 гг. при финансовой поддержке РФФИ и ФЦП «Интеграция высшего образования и фундаментальной науки». Предлагаемый проект продолжает развиваться и содержит наиболее полную реализацию идей сегментации русского предложения. Фундаментом для проводимых исследований послужила модель автоматического поверхностно-синтаксического анализа русского предложения, разработка которой была начата еще в 1971 г. в Информэлектро в секторе (затем

отделе) Д.Г.Лахути группой лингвистов под руководством Г.А.Лесскиса. Последняя версия этой модели, положенная в основу описываемой реализации, разработана Т.Ю. Кобзаревой.

Стратегии

Сохраняя принцип модульности проектируемой системы, процессор не использует в своей модели правила, а применяет грамматические стратегии для анализа предложения, каждая из которых является независимым модулем, вызываемым в качестве подпрограммы. Стратегии позволяют эффективно работать с морфологической и синтаксической омонимиями в системе. Анализ в процессоре однонаправлен: первая фаза работы анализатора выполняет построение предложных PRN и именных NRA групп (PRN-NRA модуль), во второй фазе осуществляется сегментация сложного предложения с вызовом (только в алгоритмически зафиксированных случаях) модуля сочинения, - PRN-NRA модуль⇒сегментация.

Клауза, включающая в себя хотя бы одну другую клаузу, называется сложной клаузой, или полипредикативной конструкцией [Тестелец, 2001]. Поскольку цель нашего сегментационного анализатора состоит в выделении простых сегментов в составе сложного предложения, на данном этапе присутствие полипредикативной конструкции лишь иногда фиксируется межсегментной связью в построенном графе.

Выделяются два алгоритмически обоснованных класса сегментов [Т.Ю.Кобзарева, 2002]: α- и β-сегменты. β-сегментами называется множество простых предложений в составе сложного, α-сегментами – все остальные. α-сегменты делятся на следующие подклассы:

1. Придаточные предложения - SubS сегменты.
2. Деепричастные обороты - DvS сегменты.
3. Причастные обороты и обособленные определительные обороты - AS сегменты.
4. Предложные обороты - PS сегменты.
5. Вводные обороты - PrtS сегменты.

В начале работы системы отрабатывает блок алгоритмов (PRN-NRA модуль), отвечающий за построение основных синтагм (синтаксических связей

между словоформами), без которых невозможно последующее выделение простых сегментов.

Последовательность построения синтагм:

1. Предложные группы (PRN), где предлог - хозяин, существительное - слуга.
2. Группы прилагательное-существительное (NRA), где существительное - хозяин, прилагательное - слуга.
3. В алгоритмически определенных случаях фиксируется группа управления существительным (LRN), где лексема L - хозяин, существительное - слуга.

NRA и PRN позволяют элиминировать часть потенциальных границ сегментов – операторы (запятые и сочинительные союзы), которые служат разделителями в однородных цепочках, вложенных в проективные синтагмы. Также NRA и PRN позволяют косвенно снимать некоторые типы омонимии в граммемах: например, вошедшее в PRN (‘на стол’) или вложенное в NRA (‘сменяющая день ночь’) существительное (‘стол’ и ‘день’) утрачивает именительный падеж и больше не может претендовать на позицию субъекта в сегменте. Синтагмы, так же как и сегменты, обладают свойством рекурсивности. Предложные группы PRN могут иметь неограниченной глубины вложения между предлогом и существительным-слугой, а согласованные определения NRA – любое количество параллельных вставлений произвольной глубины. Построенные PRN и NRA сворачиваются до уровня одной синтаксической единицы со всеми вложениями, находящимися внутри границ, определенных связью [Т. Кобзарева и др., 2001].

Следующий этап анализа разбивает предложение на первоначальные отрезки, границами которых являются знаки пунктуации, и приписывает каждому отрезку значение класса или подкласса. Завершающий и основной этап анализа (непосредственно сегментация) объединяет отрезки, тем самым укрупняя узлы и формируя простые сегменты. Сегментация состоит из двух модулей: α- и β-анализа. В обусловленных алгоритмически случаях, в процессе сегментации фиксируются цепочки сочиненных составляющих, связь подлежащее-сказуемое или связь глагола с прямым дополнением. Поиск и

фиксация всех таких синтагм в предложении, в отличие от PRN и NRA, не являются обязательным условием анализа.

Таким образом, любое предложение естественного языка в системе описывается двумя графами: граф синтагм и граф сегментов [И. Ножов, 2002]. Узлами графа синтагм являются терминальные единицы (словоформы), дуга в графе образует синтагму и задает тип связи. Узлами графа сегментов являются нетерминальные единицы - сегменты, - дуга в графе задает межсегментную связь. Грамматическое сочинение терминальных единиц в графе синтагм и сочинение однородных сегментов в графе сегментов нарушает древесность графов, так как каждый элемент множества узлов, образующих сочинительную связь, попарно связан со всеми остальными элементами множества и одновременно является как слугой, так и хозяином всех узлов, принадлежащих множеству сочинения. Таким образом, граф синтагм и граф сегментов - ориентированные графы, содержащие контуры и замкнутые пути [А. Белоусов, С. Ткачев, 2001]. Как следует из выше изложенного подхода к построению синтагм и сегментов, связность графа не является обязательным условием.

Морфологическая и синтаксическая омонимии

Сегментационный анализатор работает с двумя типами омонимии: морфологической и синтаксической. Морфологическая омонимия в предложении обусловлена присутствием словоформ, принадлежащих одновременно двум и более лексемам. Так во фразе ‘*Женщина мыла оконное стекло*’: слово ‘*мыла*’ омонимично: родительный падеж существительного ‘мыло’ и прошедшее время глагола ‘мыть’; слово ‘*стекло*’ также омонимично: винительный падеж существительного ‘стекло’ и прошедшее время глагола ‘стечь’. Декартово произведение омонимов дает четыре комбинаторно возможных интерпретации данного предложения, а следовательно порождает четыре графа синтагм (МИ в процессоре Диалинг). Ниже будет описан метод активизации омонимов, который часто позволяет избежать декартова произведения. Результаты сегментации также сокращают число комбинаторных вариантов (как и в системе Диалинг), так как построение связного дерева синтагм проводится в пределах одного простого сегмента, а не в рамках всего предложения. Морфологическая омонимия в сегментационном анализаторе определена только на графе синтагм.

Синтаксическая омонимия представляет собой гораздо более абстрактный случай и задается стратегиями анализа в алгоритмах. Так, синтаксическая омонимия выражается наличием или отсутствием той или иной связи в графе синтагм. Например, синтаксическая омонимия часто возникает в случаях интерпозиции обособленного определительного оборота, когда зависимое определение может иметь хозяина как слева, так и справа. Синтаксическая омонимия в графе сегментов возникает при объединении первоначальных отрезков в более крупные узлы. Омонимичный граф сегментов, порожденный таким типом омонимии, будет отличаться от своего родителя границами сегментов и, возможно, числом узлов. Синтаксическая омонимия определена как на графе синтагм, так и на графе сегментов.

Граф синтагм

Внешними составляющими в механизме построения синтагмы - направленной синтаксической связи между словоформами - служат:

- Линейное распределение словоформ в цепочке терминальных единиц предложения.
- Процедура проверки полного согласования.
- Процедура проверки частичного согласования для множественного числа.
- Процедура проверки согласования по управлению.

Линейная структура предложения S естественного языка состоит из множества словоформ $S = \{W_1, W_2, \ldots, W_n\}$, где каждая словоформа представлена множеством морфологических омонимов $W_i = \{h_1, h_2, \ldots, h_m\}$, где h_i является кортежем значений {часть речи, граммемы, примитивная модель управления}. Таким образом, предложение можно представить как упорядоченную цепочку элементов $S' = \{e_{11}, e_{12}, \ldots, e_{1m}, \ldots, e_{np}\}$, где первый индекс элемента соответствует номеру словоформы в предложении, а второй – номеру морфологического омонима словоформы. Первоначальный этап синтаксической сегментации, отвечающий за построение графа синтагм, начинает работать с линейным представлением S. При построении синтагм и поиске предикатов происходит активизация омонимов, в результате чего возникают смешанные цепочки типа $S'' = \{W_1, e_{2j}, W_3, \ldots, e_{np}\}$. Таким образом,

графом синтагм предложения S называется граф G=(S'', E), где S'' - множество узлов, состоящих из элементов смешанной цепочки S'', а E - множество упорядоченных пар на S'', то есть множество синтагм; в частном случае, при отсутствии морфологических омонимов, G=(S', E). Существует динамически пополняемый список омонимичных графов синтагм L = {G_1=(S''_1, E_1), G_2= (S''_2, E_2),…, G_k= (S''_k, E_k)}, активизация нового омонима является событием, которое вызывает пополнение списка. Каждый G_i содержит минимальное число синтагм, необходимых для дальнейшей сегментации (т.е. связность графа – необязательное условие).

Алгоритм активизации омонимов построен на принципе ленивых вычислений. Аргументом функции, принимающей решение о порождении омонимичного графа, является проверяемое алгоритмом условие или тип построенной синтагмы.

Здесь приводится пример[12] построения анализатором графа синтагм фрагмента предложения, на котором отчетливо проявляется свойство рекурсивности синтагм:

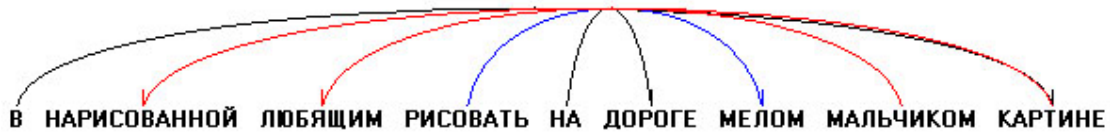


рис.1

Граф сегментов

Операция объединения задана на множестве первоначальных отрезков. Условием объединения служит:

- Сочинение.
- Процедура определения и устранения разрыва, образованного вклиниванием α-сегмента, между двумя первоначальными отрезками.

Синтаксическая сегментация проводится для каждого графа синтагм из списка L, собирая разорванные вложениями α- и β- сегменты. Предложение S представляется в виде графа, в узлах которого находятся сегменты, а дуги являются связями между сегментами, такой граф сегментов GS=(ST, SE) можно представить как множество узлов ST = {$Segm_1$, $Segm_2$, …, $Segm_n$}, где $Segm_i \subset$

[12] Использован графический интерфейс процессора.

S'', и множество SE межсегментных связей на ST. Каждому графу синтагм G из списка L соответствует множество графов $L' = \{GS_1=(ST_1, SE_1), GS_2=(ST_2, SE_2), \ldots, GS_m=(ST_m, SE_m)\}$, множественность интерпретаций графа синтагм G обусловлена возникновением синтаксической омонимии. После того, как проанализированы все элементы списка L, мы получаем множество всех возможных графов сегментов данного предложения $M = \{GS_1, GS_2, \ldots, GS_q\}$, из которых в дальнейшем должны выбираться лучшие структуры. Множественность синтаксических интерпретаций зачастую определяется естественной смысловой омонимией в предложении. Уже на этой точке выбора отсекается часть активизированных морфологических омонимов. После завершения сегментации возможно проведение полного синтаксического анализа внутри простых синтаксических единиц, каковыми являются α- и β-сегменты.

Ниже приведены два омонимичных графа сегментов предложения: “Он постоянно видел отца, красящего забор младшей сестры, старый дом и сарай.”

рис.2

рис.3

Оба варианта являются синтаксически и семантически допустимыми вариантами интерпретации этого предложения.

Сегментная проективность

Базовым ограничением, на котором строится анализ в процессоре, является проективность сегментной структуры предложения. Первоначально линейная последовательность предложения S разбивается по знакам препинания на исходные α- и β-отрезки в соответствии с приведенной

классификацией. Класс и подкласс отрезка определяется по наличию в нем «сегментообразующего» слова: подчинительного союза, деепричастия, полного причастия/прилагательного, оставшегося после анализа NRA без хозяина, и т.д. α-отрезки на этом этапе – еще не сегменты, но безусловные «начала» α-сегментов. Все остальные отрезки называются β-отрезками [Т. Кобзарева и др., 2000]. Рассмотрим известный уже нам пример предложения "Девочка, решив уже, когда ее позвали, задачу, засмеялась", разделив его на исходные отрезки:

β_1[девочка] α_1[DvS: решив уже] α_2[SubS: когда ее позвали] β_2[задачу] β_3[засмеялась]

Строя сегменты ([1β]'девочка засмеялась', [2α]'решив уже задачу' и [3α]'когда ее позвали') путем объединения исходных отрезков, получаем схему такого объединения: $\alpha_1 \rightarrow \beta_2$ и $\beta_1 \rightarrow \beta_3$, где стрелки не демонстрируют привычные грамматические связи, а лишь отражают тот факт, что подсоединяемый β-отрезок переносится влево. Между уже собранными полными сегментами определяется структура связей: в данном примере [1β]→[2α]→[3α]. Построим предложение с более сложной и глубокой иерархией вложений “Мать, когда мальчик, выйдя во двор, где стояла машина, к которой было необходимо подойти, споткнулся, не заметив приступка, и упал в сугроб, наметенный за ночь, выбежала ему помочь”:

β_1[мать] α_1[SubS: когда мальчик] α_2[DvS: выйдя во двор] α_3[SubS: где стояла машина] α_4[SubS: к которой было необходимо подойти] β_2[споткнулся] α_5[DvS: не заметив приступка] β_3[и упал в сугроб] α_6[AS: наметенный за ночь] β_4[выбежала ему помочь]

Схема объединения исходных отрезков в полные сегменты представлена на рис.4:

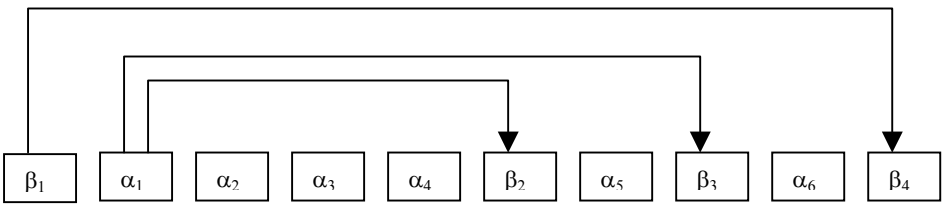


рис.4

В результате процедуры объединения образовано семь сегментов: [1β]'мать выбежала ему помочь', [2α]'когда мальчик споткнулся и упал в сугроб', [3α]'выйдя во двор', [4α]'где стояла машина', [5α]'к которой было необходимо

подойти', [6α]'не заметив приступка', [7α]'наметенный за ночь'. Структура межсегментных связей показана на рис.5:

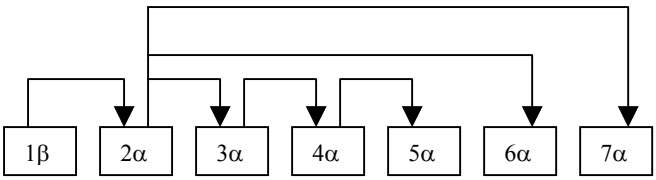


рис.5

Получившаяся схема объединения (рис.4) проективна, т.е. не содержит пересечения стрелок. Такого рода проективность, связанная с сегментной структурой предложения, - абсолютно жесткая, в отличие от традиционной синтаксической, которая может нарушатся для русского языка в устной речи или не определена для языков с полностью свободным порядком слов (австралийский язык вальбири [Я. Тестелец, 2001]). Связи между уже построенными полными сегментами (рис.5) также проективны. Такое свойство схемы объединения и свойство структуры полных сегментов будем называть сегментной проективностью [Т.Ю.Кобзарева, 2002]. Гипотетически сегментная проективность не может нарушатся ни в одном из существующих естественных языков, но для доказательства этого утверждения необходимо проводить типологическое исследование.

Таким образом, можно вывести три простых структурных ограничения на сегментации:

1. Сегментная проективность.
2. β-отрезок может быть присоединен к α-отрезку только в том случае, если он находится справа от α в линейной последовательности предложения.
3. Два α-отрезка не могут быть объединены; в случае сочинения сегментов (сочиненные цепочки придаточных или причастных оборотов, или простых сегментов в составе сложного предложения, etc.) сохраняется независимость каждого сегмента в цепочке сочиненных как отдельной единицы анализа.

Метод монтажа

Методом монтажа (в соответствии с приведенной аналогией) будем называть α- и β-анализ, опирающийся на свойство рекурсивности сегментов и определенные выше структурные ограничения. α-анализ, являясь центральной и

алгоритмически наиболее сложной стратегией сегментации [Т. Кобзарева и др., 2002], предшествует β-анализу и позволяет ликвидировать обусловленные вложением α-сегментов разрывы простых β-сегментов в составе сложного предложения S. Алгоритм α-анализа идет справа налево по S, осуществляя поиск α-отрезков, и заново вызывается рекурсивно для текущего α-отрезка после каждого нового присоединения правостоящего β-отрезка, что отражает свойство рекурсивности сегмента в стратегии алгоритма. Поиск справа налево каждого следующего α-отрезка в S дает возможность начинать сборку α-сегментов с вложений максимальной глубины. Возможны две ситуации при рассмотрении найденного правостоящего β-отрезка:

1. β-отрезок находится непосредственно (контактно) справа от α-отрезка;
2. между α- и β-отрезками находятся один и более (число вложений не ограничено) полных (уже обработанных) α-сегментов.

Определяются грамматические условия для присоединения β-отрезка [Т. Кобзарева и др., 2001]:

а) синтаксическая неполнота α-отрезка (α-incompleteness);

б) синтаксическая неполнота β-отрезка (β-incompleteness);

в) в α-отрезке существует слово W_1, способное управлять словом W_2 в β-отрезке (α-manage);

г) слово или группа слов в α-отрезке образует сочинительную связь со словом или группой слов в β-отрезке (coordination);

В первой ситуации – контактного расположения β- относительно α-отрезка – допустимо использование только условия сочинения (coordination) для присоединения β-отрезка, при этом накладываются дополнительные грамматические ограничения (constraints) на поиск сочинения.

Представим множество исходных α- и β- отрезков на предложении S в виде вектора $V = \{Sg_1, Sg_2, .., Sg_n\}$, где, например, для S (“девочка, решив…, засмеялась”), заданного $V = \{Sg_1=\beta_1, Sg_2=\alpha_1, Sg_3=\alpha_2, Sg_4=\beta_2, Sg_5=\beta_3\}$, $V[3] = \alpha_2$, $V[5] = \beta_3$, etc. Тогда запишем алгоритм α-анализа псевдокодом [Т. Кормен и др., 2001, стр. 20]:

```
α-Analyse
1  for i ← length[V] downto 1
2      do if V[i] = α
3          then V[i] ← Montage(V, V[i], i, i+1)

Montage(V, α, i, j)
1  if j ≤ length[V]
2      then if V[j] ≠ β
3              then α ← Montage(V, α, i, j+1)
4              else if ( j = i + 1 and coordination(α, V[j], constraints) )
                       or (
                            j > i + 1
                            and (
                                    α-incompleteness(α)
                                    or (
                                         β-incompleteness(V[j])
                                         and (
                                                α-manage(α, V[j]) or  coordination(α, V[j])
                                         )
                                     )
                             )
5                      ) then α ← α ⊕ V[j]
6                              delete(V[j])
7                              α ← Montage(V, α, i, j)
8  return α
```

Учитывая рекурсивный характер задачи построения α-сегментов, в алгоритме α-анализа, записанного псевдокодом, для большей наглядности используется рекурсивный вызов функции Montage. Очевидно, что для эффективной программной реализации подобного типа рекурсию можно и необходимо переводить в итеративную форму, используя while-цикл [Н. Вирт, 2001]. Приведем результат работы процессора в ходе α-анализа сложного предложения, содержащего разрывные сегменты с α-вложениями, на рис.6:

“Когда, увидев в зеркале, принадлежавшем, как говорил брат, отцу, свое заплаканное лицо, Мария схватила письмо, лежавшее на столе, и зажгла свечу, в комнату вошел Иван.”

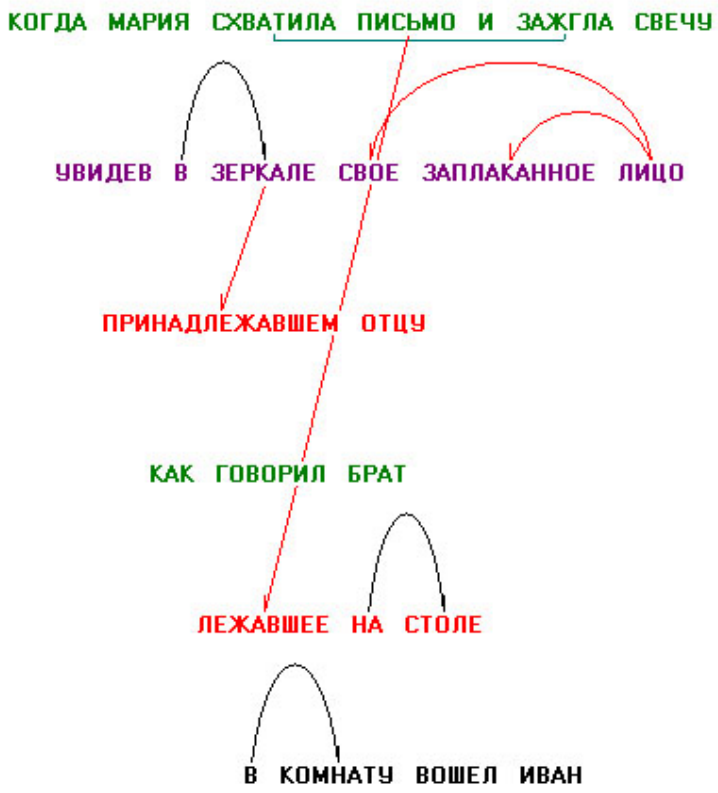


рис.6

После завершения α-анализа построенные полные α-сегменты “изымаются” из линейной последовательности S, вследствие чего на вход β-анализу поступает вектор V, содержащий последовательность β-отрезков, оставшихся непроанализированными или неприсоединенными к α-сегментам. Так, для S (“девочка, решив…, засмеялась”) вектор V на входе β-анализа принимает вид V = {$Sg_1=\beta_1$, $Sg_2=\beta_3$}. В упрощенном виде алгоритм β-анализа [Т. Кобзарева, 2002] можно представить в виде последовательности итераций: (а) процедура поиска неморфологического предиката (НМП) в β-отрезках; (б) установление границ между подгруппами последовательно расположенных β-отрезков, при условии фиксации НМП или постановки ‘;’ (в определенных случаях ‘:’) между отрезками; (в) объединение внутри подгрупп контактно расположенных β-отрезков, при условии синтаксической неполноты одного из двух β-отрезков и/или сочинения именных составляющих или предикатов двух β-отрезков. Приведем результат работы процессора в ходе β-анализа сложного предложения, содержащего два простых сегмента, разорванных α-вложениями, на рис.7:

“Едва уловимую особенность, отличавшую его сына от всех тех детей, которые по его мнению должны были стать людьми, ничем не замечательными, он

понимал как тайное волнение таланта, и, твердо помня, что покойный тесть был композитором, он в приятной мечте, похожей на литографию, спускался ночью со свечой в гостиную, где вундеркинд в белой рубашонке до пят играет на огромном черном рояле.” (В. Набоков)

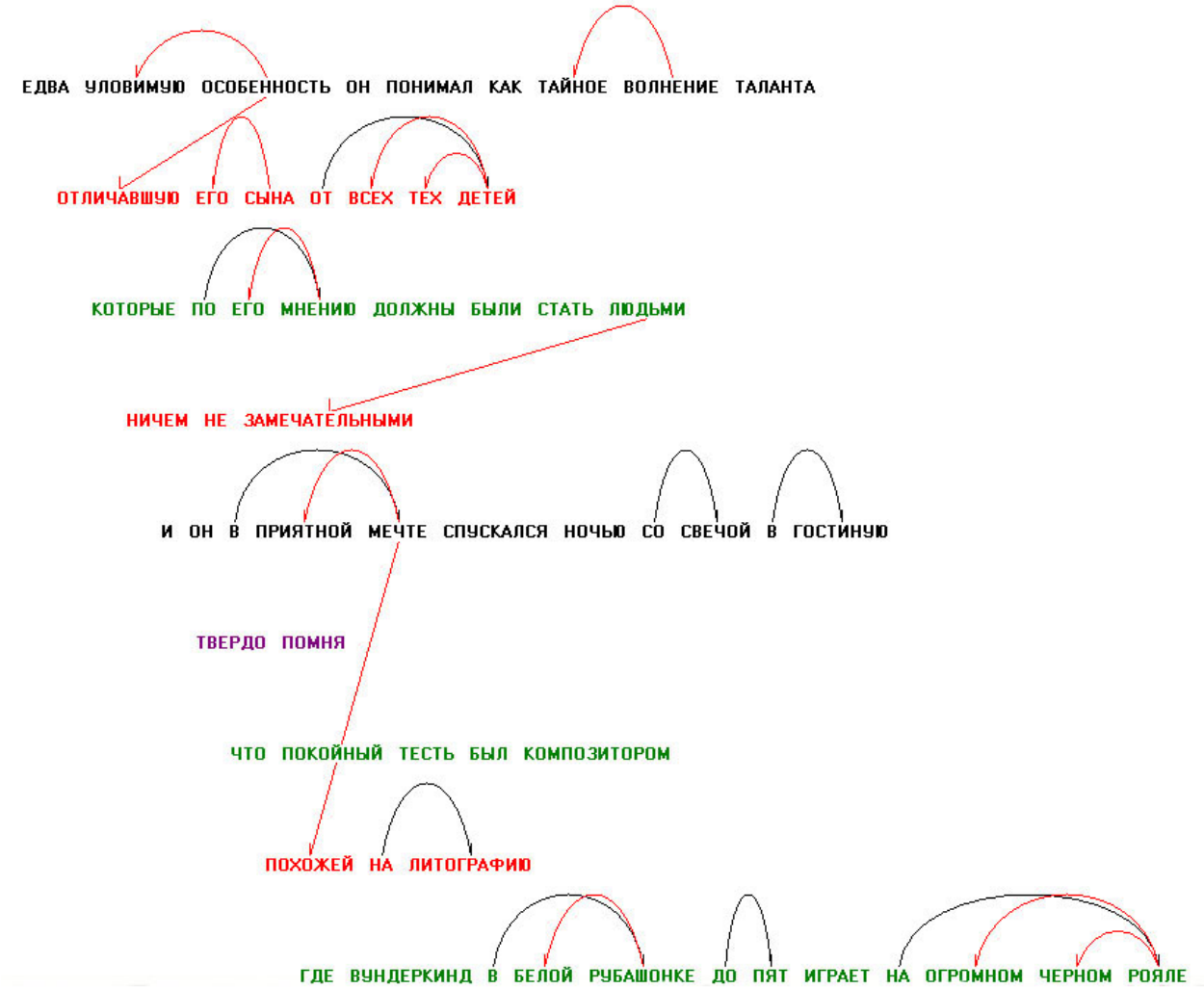


рис.7

Метод активизации омонимов

Активизация морфологического омонима, которая возникает в тех случаях, когда хотя бы один из омонимов словоформы $h \in W$ не отвечает проверяемому условию/ограничению или не способен образовать строящуюся синтагму, порождает отдельный граф синтагм G, состоящий из узлов новой смешанной цепочки типа S''. Интерпретация S'' на уровне синтагм всегда однозначна. В точке выбора порожденный граф наследует текущее состояние своего родителя, копируя ранее построенные синтагмы. Новый граф выделяется в отдельный поток, где процедура анализа продолжается из точки выбора. В

зависимости от прикладной системы, использующей модель сегментации, потоки могут работать параллельно или последовательно.

Активизация синтаксического омонима возникает на этапе построения α- и β- сегментов. В точке выбора порождается граф сегментов GS, что создает множественность интерпретаций для графа синтагм G на уровне сегментов. Как и в случае морфологической омонимии, порожденный граф наследует текущее состояние своего родителя, копируя ранее построенные сегменты, и новый граф выделяется в отдельный поток.

Метод активизации омонимов состоит из следующих понятий, определенных в процессоре:

- Условие/ограничение: в ходе работы алгоритмов-стратегий (PRN-NRA модуль, α-анализ, β-анализ, etc.) в пределах одного графа синтагм или сегментов проверяются условия/ограничения для некоторых элементов e_{ik} и e_{jm} цепочки S''типа: e_{ik} и e_{jm} согласованы или между e_{ik} и e_{jm} ∃ предикат (e_{xy} со значением части речи p, где p ∈ Предикат = [финитная ф. гл., кр. прил., кр. прич., предикатив]). Условия/ограничения, используемые стратегиями построения синтагм и сегментов, могут определять как морфологическую, так и синтаксическую омонимии. Подтвержденные в ходе анализа условия/ограничения, способные задавать синтаксическую омонимию, называются многозначными.
- Синтагма: построение синтаксического отношения $R(e_{ik}, e_{jm})$, каковым является NRA, PRN, LRN, etc. Синтагма, при определенных условиях/ограничениях (интерпозиция: "…сыном, осужденным отцом, …" R(сыном, осужденным) vs. R(отцом, осужденным)), определяет синтаксическую омонимию на графе синтагм.
- Событие: событие возникает в системе в результате подтверждения условия/ограничения или построения синтагмы.
- Точка выбора: событие в системе происходит в некоторой точке процедуры анализа, которая может быть определена в любом из модулей процессора; в случае порождения омонимичного графа такая точка выбора служит координатами места в процедуре анализа, с которого будет начат анализ нового графа.

- Состояние: текущем состоянием графа называется множество его узлов и связей, состояние графа фиксируется в точке выбора.
- Класс эквивалентности: класс эквивалентности $[h]_p$, где p – отношение эквивалентности на множестве омонимов словоформы $W \in S$, h – омоним словоформы W, состоящий из пары значений pos – часть речи и GR – множество граммем, тогда $p = \{(h_i, h_j): h_i, h_j \in W; h_i=\{pos_i, GR_i\}; h_j=\{pos_j, GR_j\}; h_i: (pos_i \in X) \& (Y \cap GR_i \vee Y = \varnothing)$ и $h_i \equiv h_j$, если и только если для h_j справедливо $(pos_j \in X) \& (Y \cap GR_j \vee Y = \varnothing)\}$; X и Y – множества, заданные инициировавшим событие условием/ограничением. Пример 1: между элементами e_{ik} ($e_{ik} := h_k \in W_i$) и e_{jm} ($e_{jm} := h_m \in W_j$) установлено согласование по грамматическому числу и падежу; известно, что для e_{jm} допустимы следующие значения селективных признаков POS = {существительное, местоимение} и согласование определено по одному из возможных значений граммем GR = {{мн., им.}, {мн., вн.}}, тогда условием/ограничением, подтвердившим согласование, формируется множество X = POS и Y = GR для отношения эквивалентности p; предположим, словоформа W_j состоит из трех омонимов $W_j = \{h_1=\{$сущ., {мн., им.}}, $h_2=\{$сущ., {мн., вн.}}, $h_3=\{$гл., $GR_3\}\}$; пусть для h_m m = 2, тогда $f_p(h_m) = [h_m]_p = \{h_1, h_2\}$. Пример 2: допустим, что в цепочке S'' найден предикат e_{jm} ($e_{jm} := h_m \in W_j$), тогда условием/ограничением, осуществлявшим поиск предиката, формируется множество X = Предикат и $Y = \varnothing$ для отношения эквивалентности p; предположим, словоформа W_j состоит из четырех омонимов $W_j = \{h_1=\{$ кр. прил., $GR_1\}$, $h_2=\{$наречие, $GR_2\}$, $h_3=\{$предикатив, $GR_3\}$, $h_4=\{$частица, $GR_4\}\}$; пусть для h_m m = 3, тогда $f_p(h_m) = [h_m]_p = \{h_1, h_3\}$.
- Функция разбиения: аргументом функции является класс эквивалентности $[h]_p$ и множество W; если $B = W \setminus [h]_p$ и $B \neq \varnothing$, то вызвать функцию клонирования, иначе завершить обработку события; $[h]_p$ будем также называть множеством W', а B – множеством W''.
- Функция клонирования: аргументом функции служат точка выбора, в которой возникло событие, и состояние графа в данной точке; функция клонирования изменяет множество узлов или связей исходного графа и порождает омонимичный вариант исходного графа, который добавляется в

конец списка L или L’ для графа синтагм или сегментов соответственно. В точке выбора отличие между исходным и клонированным вариантами графа - незначительное, но по окончании процедуры анализа для каждого из вариантов различие в узлах и связях, как правило, становится существенным. Так, морфологическая омонимия на графе синтагм может повлиять на узлы зависящего от него графа сегментов, т.е. на границы построенных сегментов.

Таким образом, принцип ленивых вычислений, алгоритмически мотивированный условиями/ограничениями, реализуется в системе через событие и обработку каждого такого события. В процессоре зафиксировано три сценария обработки события:

(1) Событие[$G=(S'', E)$, условие/ограничение] $\Rightarrow$ Функция разбиения($[h]_p$, W) $\Rightarrow$ Функция клонирования(точка выбора, состояние $G=(S'', E)$) == $G=(S_x'', E)$ и $G'=(S_y'', E_y)$, где $W' \subset S_x''$, $W'' \subset S_y''$ и $E = E_y$.

(2) Событие[$G=(S'', E)$, синтагма syn] $\Rightarrow$ Функция клонирования(точка выбора, состояние $G=(S'', E)$) == $G=(S'', E_x)$ и $G'=(S_y'', E_y)$, где $syn \in E_x$, $syn \notin E_y$ и $S'' = S_y''$.

(3) Событие[GS=(ST, SE), многозначное условие/ограничение] $\Rightarrow$ Функция клонирования(точка выбора, состояние GS=(ST, SE)) == $GS=(ST_x, SE)$ и $GS'=(ST_y, SE_y)$, где $ST_x \neq ST_y$ и $SE = SE_y$. Синтаксическая омонимия на графе сегментов возникает при неоднозначности присоединения некоторого β-отрезка. Например, для исходного графа GS узел $(\alpha \oplus \beta) = \alpha'$ и $\alpha' \in ST_x$, а в порожденном GS’ узлы $\alpha, \beta \in ST_y$ (ситуация синтаксической омонимии на границах сегментов была показана на рис.2 и 3).

Основная идея изложенного метода активизации омонимов состоит в том, чтобы избежать (в тех случаях, когда это возможно) полного декартова произведения морфологических омонимов ($W_1 \times \ldots \times W_n$) и повторного построения общих для омонимичных графов синтагм и сегментов.

Общая схема реализации анализатора

В приведенной ниже схеме (рис. 8) отражено взаимодействие модулей в программной реализации автоматической синтаксической сегментации. Каждый модуль на схеме реализует лингвистическую стратегию или механизм

управления анализом в процессоре. Механизм управления отвечает за построение графов синтагм и сегментов, применение метода активизации омонимов и формирование потоков в процессе, также в механизм управления включены программные библиотеки, реализующие общие для всех лингвистических модулей функции (проверка полного и частичного согласования или грамматического управления, проверка общих структурных ограничений и т.д) и сценарии обработки события. Стрелка вида X → Y на схеме означает, что модуль Y может быть вызван модулем X в качестве подпрограммы.

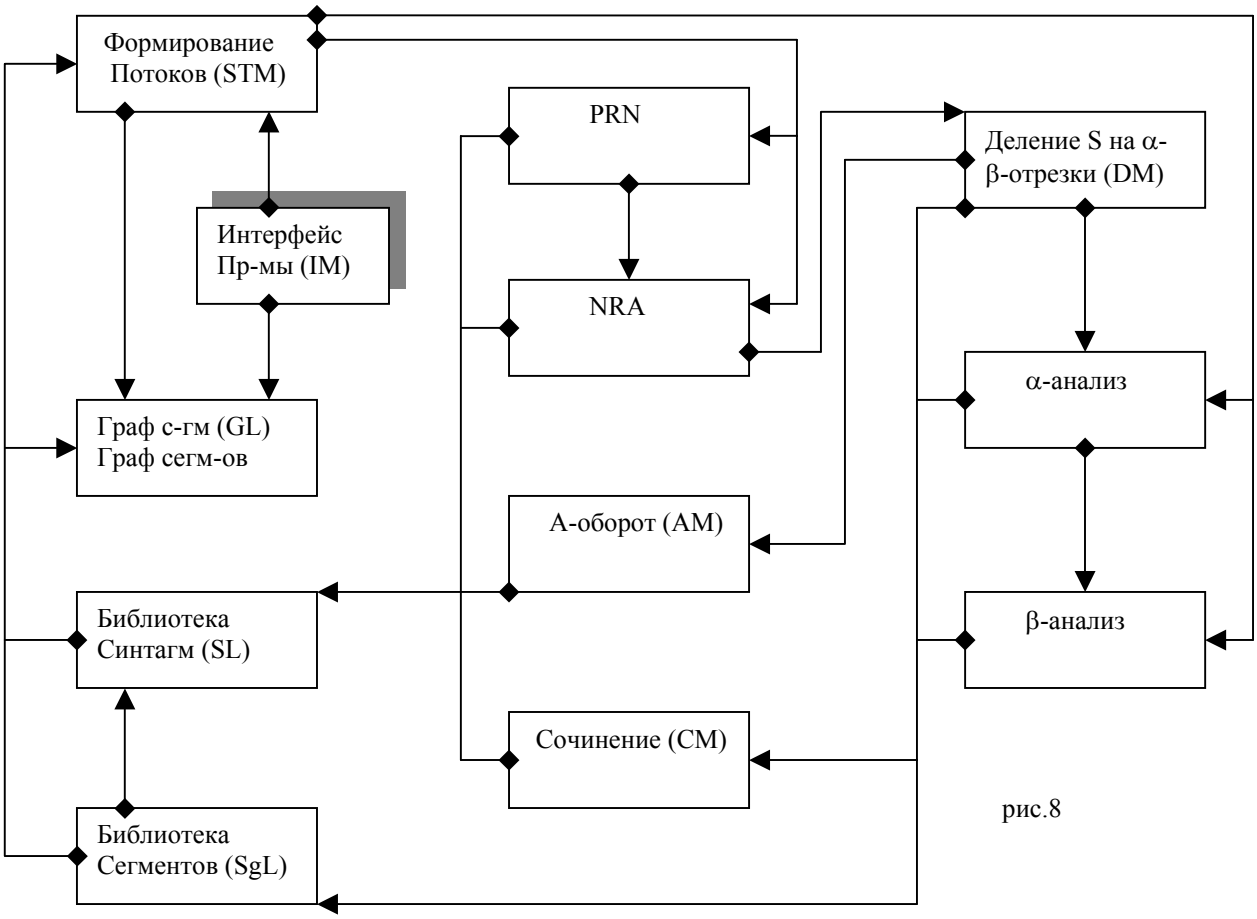


рис.8

Входными данными программы (модуля IM) является текст. Каждое законченное предложение текста, выделенное знаками пунктуации, проходит предварительную графематическую и морфологическую обработку. IM для каждого отдельного предложения инициализирует граф синтагм, заполняя исходными терминальными единицами (результатами обработки) его внутренние структуры данных для хранения узлов графа, после чего открывает

центральный поток (модуль STM). Построение графа синтагм и сегментов анализируемого предложения осуществляется в последовательности PRN→NRA→DM→α-анализ→β-анализ, базовых грамматических стратегий анализа. В процессе анализа внутри центрального потока через библиотеки SL и SgL формируются новые потоки в STM. Каждый открывающийся поток является результатом успешного сценария обработки события и принимает при инициализации клонированный граф синтагм или сегментов. Каждый поток в системе использует общую процедуру анализа, в соответствии со схемой, и может создавать неограниченное число новых потоков в процессе работы. Стрелки STM→PRN, STM→NRA, STM→α-анализ и STM→β-анализ на схеме демонстрируют с точностью до модуля местоположение точек выбора, с которых может начинаться процедура анализа в очередном потоке. Дадим функциональные характеристики модулей механизма управления: STM – формирование и инициализация потоков; GL – объектно-ориентированная библиотека классов, позволяющая строить графы синтагм и сегментов и предоставляющая набор методов (процедур и функций) для работы с ними; SL реализует общие лингвистические функции (проверка согласования, управления), а также 1 и 2 сценарии обработки события; SgL реализует структурные ограничения на сегментации и 3 сценарий обработки события. Дадим функциональные характеристики лингвистических модулей [Т. Кобзарева и др., 2000]: PRN и NRA проводят синтагматический анализ в процессоре; AM осуществляет анализ обособленных согласованных оборотов в предложении; CM – поиск грамматического сочинения; DM – деление предложения на первоначальные α- и β-отрезки и классификация α-отрезков; α-анализ и β-анализ реализуют синтаксическую “сборку” α- и β-сегментов из первоначальных отрезков. Использование потоков повышает устойчивость работы программы и позволяет эффективно распределять вычисления на серверах с многопроцессорной архитектурой.

Для отладки работы и развития анализатора используется корпус тестов, состоящий из 230 сложных предложений. Такой корпус позволяет контролировать влияние вносимых в процессор изменений на результаты анализа. Приведем значения характеристик такого контроля за системой на примере пяти правильно построенных предложений (т.е. предложений, для

которых программой были построены только синтаксически допустимые варианты синтагматических и сегментационных интерпретаций).

Предложение	Кол-во слов (включая знаки препинания)	Кол-во сегментационных вариантов (мотивированное синтаксической омонимией)	Кол-во синтагматических вариантов (мотивированное морфологической омонимией)	Кол-во возможных вариантов (декартово произведение омонимов)	Время анализа (в секундах)
Нелепая провинциальная дама, которая раздражала друзей утверждением, что паровозы, пароходы и прочие новшества изобретены ее сыном, приводила всех в неистовство, деликатно намекая, что он сочинитель каждого прочитанного ею романа.	36	2	4	48	0,3
И потом до самого разъезда мы ни о чем не потолковали, не сговаривались насчет будущих**, в** даль тронувшихся пятнадцати дорожных лет, нагруженных частями наших несобранных встреч, и следя за ней в лабиринте жестов и теней жестов, из которых состоял вечер, я был поражен ее невниманием ко мне, чистосердечнейшей естественностью этого невнимания, ибо я еще тогда не знал, что, если бы сказал я два слова, оно сменилось бы тотчас чудной окраской чувств, веселым, добрым, по возможности деятельным участием, точно женская любовь была родниковой водой, содержащей целебные соли, которой она из своего ковшика поила всякого, если только напомнить.	115	1	8	96	1,3
Железнодорожная проза, как дамская сумочка этого предсмертного мужичка, полна инструментами сцепщика, бредовыми частичками, скобяными предлогами, которым место на столе судебных улик, развязана от всякой заботы о красоте.	34	1	2	4	0,2
Девочка, решив уже, когда ее позвали, задачу, засмеялась.	13	1	1	4	0,001
Участники российских финансовых рынков, продавая рубли, старались минимизировать возможные негативные последствия углубления финансового кризиса, которые, как свидетельствует мировой опыт, проявляются в резком обесценении национальной валюты.	31	1	1	1	0,001

Заключение

Метод монтажа и метод активизации омонимов лингвистически адекватны и универсальны, т.е. независимы от анализируемого естественного языка. Адекватность понимается как соответствие модели процессора трем сформулированным принципам: описательному, объяснительному и эмулирующему.

Программная реализация процессора выполнена на языке Object Pascal с использованием C библиотек, система анализа протестирована на пятистах сложных предложениях. Взаимодействие между морфологическим и синтаксическим модулями в программе организовано через текстовый файл заданного формата, выходные данные сегментационного процессора также представляются в виде текстового файла. Настоящий процессор, в первую очередь, рассматривается как экспериментальное пространство для создания промышленных систем синтаксического анализа.

Все рассмотренные в настоящей работе процессоры созданы в течение последних 10-12 лет. Можно выделить три доминирующих подхода к проектированию моделей синтаксического анализа естественного языка: лексикализм (HPSG), контекстно-свободные грамматики (LinkParser) и алгоритмический подход. Последний характеризуется разделением на уровни лингвистического анализа и модульностью системы. Алгоритмический подход состоит из двух направлений: основу первого составляют правила (процессор Диалинг), а второго – грамматические стратегии (анализатор ОИС). STP скорее относится к алгоритмическому процессору смешанного типа. Если основным критерием для построения и оценки правильности синтаксической структуры предложения в лексикализме и CFG служит связность графа, то алгоритмический подход, возвращаясь к шахматной аналогии, оперирует «фокусным пространством», выделяя «куски-ситуации», соответствующие сегментам, каждый из которых содержит явный или скрытый предикат. Модели типа LinkParser подразумевают жесткий порядок слов в предложении и морфологическую простоту анализируемого языка. Унифицирующие грамматики целиком зависят от полноты лексикона и выверенности каждой из его лексических статей. Модульные анализаторы, стараясь использовать наиболее общие синтаксические законы языка, дают возможность снизить

зависимость анализа от словаря и значительно сократить затраты на разработку лингвистического обеспечения. Теряя, в определенном смысле, прозрачность архитектуры процессора и его программной реализации, алгоритмическая модель часто позволяет избежать избыточности вычислений при построении синтаксической структуры и лучше поддается контролю за принятием решений в процессе анализа, что отвечает принципам проектирования систем ИИ.

ГЛАВА 4. ПРИКЛАДНЫЕ ВОЗМОЖНОСТИ СИНТАКСИЧЕСКИХ ПРОЦЕССОРОВ В СИСТЕМАХ МАШИННОГО ПЕРЕВОДА И АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТОВ

В настоящей главе приводятся технические характеристики и оценка качества методик, разработанных для систем морфологического и предсинтаксического анализа и для процессоров синтаксической сегментации; дается краткое описание прикладных систем АОТ и МП, в которых были внедрены и опробованы предложенные методики; рассматриваются дальнейшие перспективы использования процессоров синтаксической сегментации в системах АОТ.

Качество методики морфологического анализа без словаря, разработанного в НТЦ «Система», оценивалось по следующим показателям :

- отношение количества ошибочно построенных гипотез к общему количеству полученных основ;
- минимальное количество словоформ одной лексемы, достаточное для гарантированного формирования правильной основы.

Общая скорость работы системы автоматической индексации БД характеризуется двумя показателями:

- скорость обработки текстов морфологическим анализом в процессе автоматического построения грамматического словаря основ;
- скорость индексации текстов с использованием построенного словаря основ.

Результаты испытаний методики морфологического анализа НТЦ «Система»:

Характеристика	Значение
Отношение количества ошибочно	5

построенных гипотез к общему количеству полученных основ, [%]	
минимальное количество словоформ одной лексемы, достаточное для гарантированного формирования правильной основы[13]	3
Скорость обработки в процессе построения словаря основ [Мб/ч]	6
Скорость индексации текстов с использованием построенного словаря основ [Мб/ч]	100

Полученное в результате тестирования отношение количества ошибочно построенных гипотез к общему количеству полученных основ показывает приемлемую погрешность метода. Заметим, что погрешность метода уменьшается при накоплении информации, т.е. увеличении выборки (количества словоформ для одной лексемы), которое зависит от объема «прочитанных» системой текстов. Скорость обработки существенно зависит от программной реализации методики. Имеющаяся программная реализация метода может быть значительно оптимизирована для дальнейшего промышленного использования. Морфологический анализатор без словаря был внедрен в первую версию ИПС законодательной БД, созданной на платформе Oracle 7.3 в НТЦ «Системы».

Качество морфологического анализа с использованием лексикона зависит от двух параметров:

объем словаря – число включенных в словарь лексем;

морфологическое покрытие – процент найденных слов в лексиконе в процессе анализа произвольного корпуса текстов.

Для морфологического компонента проекта Диалинг объем словаря составляет 165 тысяч лексем (в это число входят имена собственные и географические названия), покрытие – 98%. Скорость анализа достигает 200 Мб/ч.

Процессор LxPlatform 3.5 состоит из трех основных модулей: stemmer – морфанализ словоформы; tagger – снятие морфологической омонимии; np-grouper – выделение NP составляющих из текста. Приведем показатели оценки качества для модулей LxPlatform и скоростные характеристики:

Модуль	Характеристика	Значение
Stemmer	Объем словаря, [тыс. лексем]	150

[13] В 80% случаев достаточно 2 словоформ одной лексемы.

Stemmer	Морфологическое покрытие, [%]	96
Stemmer	Скорость обработки текста, [Мб/ч]	450
Tagger	Отношение количества правильно выбранных для омонимичных словоформ лемм к общему количеству омонимичных словоформ, [%]	99
Tagger	Отношение количества правильно выбранных для омонимичных словоформ усеченных грамматических помет к общему количеству омонимичных словоформ, [%]	94,5
Tagger	Скорость обработки текста, [Мб/ч]	430
np-grouper	Отношение количества правильно построенных NP к общему количеству выделенных NP из текста, [%]	98,3
np-grouper	Скорость обработки текста, [Мб/ч]	400

Технология LxPlatform 3.5 была доведена до промышленного использования и успешно внедрена в системы АОТ отдела разработки Inxight.

Система АОТ Inxight	Функционал системы
Murax	Система, позволяющая расширить интеллектуальные возможности поисковой машины (АИПС), состоит из двух частей: (1) Concept Linker строит дерево концептов для множества индексируемых документов, где каждый концепт является частотной синтаксической составляющей NP (таким образом, выделение именных групп является центральной частью концептуального анализа документов в Murax); (2) Similarity Search позволяет находить для исходного текста родственные (похожие) документы.
Categorizer	Проводит автоматическую классификацию входящего информационного потока (документов) по определенной заранее таксономии. Классификатор необходимо обучать на таксономии заданной предметной области. Для полноценного обучения в среднем требуется 30 документов на каждую категорию.
Smart Discovery	Позволяет создавать в полуавтоматическом режиме таксономию предметной области на заданном массиве документов.

Каждое из вышеприведенных приложений использует LxPlatform в качестве ядра программы, подвергая результаты обработки текстов, полученных модулями tagger и np-grouper, дополнительному вероятностно-статистическому анализу.

Оценка качества работы синтаксического процессора определяется парой «точность (уровень ошибок в построенных синтаксических структурах предложений), полнота (степень покрытия текста синтаксическими связями, или связность графа предложения)».

Приведем показатели оценки качества и скорости для синтаксического анализа группы Диалинг[14]:

Характеристика	Значение
Отношение количества правильно сегментированных сложных предложений к общему количеству сложных предложений в тексте, [%]	78
Отношение количества правильно построенных синтаксических групп к общему количеству построенных групп в предложениях текста, [%]	97
Отношение количества правильно выбранных с использованием системы весов МИ сегмента к общему количеству выбранных МИ сегментов в предложениях текста, [%]	95
Покрытие: отношение количества слов, вошедших в состав синтаксических групп, к общему количеству слов в тексте, [%]	79
Скорость синтаксического анализа, [слов/сек.]	350

Синтаксический процессор является компонентой системы МП Диалинг. Результаты синтаксического анализа поступают на вход семантического модуля [А. Сокирко, 2001] системы, использующий в процессе работы русский общесемантический словарь (РОСС). Семантика берет от синтаксического компонента границы построенных сегментов, лучшие МИ сегментов, получившие максимальный вес, и только те синтаксические группы, которые имеют высокую точность построения и не требуют дополнительной семантико-синтаксической проверки через РОСС. Таким образом, центральной функцией синтаксического анализатора в составе системы МП Диалинг является сегментация сложного предложения и выбор МИ внутри сегментов (т.е. снятие грамматической омонимии).

Синтаксический процессор группы Диалинг используется компанией «ВААЛ» (www.vaal.ru) для создания психологических методик анализа предвыборных, социологических и политических текстов [И. Ножов, 2002].

В Исследовательском центре искусственного интеллекта ИПС РАН в Переславле-Залесском синтаксический анализ группы Диалинг интегрирован как компонент лингвистического процессора в систему автоматического извлечения информации из текстов на русском языке [Д. Кормалев, Е. Куршев и др., 2002].

Приведем показатели оценки качества и скорости для программной реализации синтаксической сегментации группы ОИС РГГУ:

Характеристика	Значение
Отношение количества правильно сегментированных сложных предложений к общему количеству сложных предложений в тексте, [%]	85
Отношение количества правильно построенных синтагм к общему количеству построенных синтагм в предложениях текста, [%]	98
Скорость анализа, [слов/сек.]	450

[14] Все значения характеристик приводятся для состояния проекта Диалинг июня 2001г.

Необходимо отметить, что данный анализатор не рассчитан на промышленное использование и, следовательно, имеет ряд ограничений в процессе анализа. Так, в сегментацию не включена обработка вводных и уточняющих оборотов и приложений, так как анализ подобных конструкций является второстепенной и решаемой задачей, решение которой не влияет на общий механизм алгоритмов сегментации. Имеющаяся программная реализация синтаксической сегментации может быть значительно оптимизирована для дальнейшего промышленного использования. Оптимизация программы и реализация метода активизации омонимов на языке C++ способны существенно увеличить скорость анализа (в 7-10 раз). Являясь внутриуниверситетским проектом, настоящая программная реализация синтаксической сегментации служит доказательством работоспособности лингвистических алгоритмов, разработанных Т.Ю. Кобзаревой, и предложенных методов монтажа и активизации омонимов.

Процессор синтаксической сегментации группы ОИС используется в учебном процессе Института Лингвистики РГГУ для проведения лабораторных и семинарских занятий.

Создание эффективной системы сегментации сложного предложения позволит осуществлять поиск заданного образца в пределах одного сегмента, что существенно повысит качество работы интеллектуальных ИПС. Сегментационный анализатор, способный выделить сегмент простого предложения в составе сложного, может стать центральным звеном в программах автоматического реферирования текстов. Сегментация предложения - компонента полного синтаксического анализа, без которой представляется невозможным полноценное решение задач извлечения информации из текстов, автоматической кластеризации информационного потока и машинного перевода.

ЗАКЛЮЧЕНИЕ

Сформулируем основные результаты исследования:

- проведено сравнение существующих подходов к проектированию синтаксического анализа языка: опирающейся на лексикализм унифицирующей грамматики (HPSG), конекстно-свободной грамматики (LinkParser) и модели поверхностного текстового процессора (STP);

- разработаны два метода автоматического синтаксического анализа предложения: метод активизации омонимов и рекурсивный метод монтажа сегментов. Метод активизации омонимов реализует в системе синтаксической сегментации принцип ленивых вычислений и позволяет избежать полного декартова произведения морфологических омонимов и повторного построения общих для омонимичных графов синтагм и сегментов, что существенно снижает число рассматриваемых структурных интерпретаций предложения. Метод монтажа, опирающийся на свойство сегментной проективности в естественном языке, является базовым механизмом для выделения и классификации сегментов в составе сложного предложения;
- метод монтажа и метод активизации омонимов лингвистически адекватны и универсальны (независимы от анализируемого ЕЯ). Адекватность понимается как соответствие модели процессора трем сформулированным принципам: описательному, объяснительному и эмулирующему;
- автоматическая синтаксическая система ОИС представляет собой не законченный промышленный продукт, а экспериментальную систему для отработки лингвистических решений;
- реализация системы ОИС позволила создать автоматическую синтаксическую сегментацию русского предложения без искусственных ограничений на анализ;
- разработан оригинальный метод прикладного морфологического анализа без использования словаря, опирающийся на построение леса деревьев морфологических гипотез и сравнение (корреляция) через матрицы инцидентности полученных деревьев для дальнейшей унификации грамматических гипотез;
- экспериментально доказана возможность применение скрытых моделей Маркова для задачи снятия морфологической омонимии в русскоязычном тексте.

ЛИТЕРАТУРА

[D.Grune, C.Jacobs, 1990] D.Grune, C.Jacobs. Parsing Techniques. A practical guide. – Vrije Universiteit, Amsterdam, 1990.

[Г. Буч, 2000] Г. Буч. Объектно-ориентированный анализ и проектирование. – М.: «Издательство Бином», 2000.

[I. Sag, T. Wasow, 1999] Ivan A. Sag, Thomas Wasow. Syntactic Theory: A Formal Introduction. – Stanford University, 1999

[С. Бурлак, С. Старостин, 2001] С. А. Бурлак, С. А. Старостин. Введение в лингвистическую компаративистику. – Эдиториал УРСС, М., 2001.

[M. Boden, 1990] M. Boden. Artificial intelligence and images of man. // Perspectives From Artificial Intelligence, 1990.

[Xerox, 1999] Examples of Networks and Regular Expressions. // www.xrce.xerox.com/research

[И. Мельчук, 1999] И. А. Мельчук. Опыт теории лингвистических моделей «Смысл ⇔ Текст». - М., 1999.

[Ф. де Соссюр, 1999] Ф. де Соссюр. Курс общей лингвистики. – М., 1999.

[S. Oepen, K. Netter, 1997] S. Oepen, K. Netter. Test Suites for Natural Language Processing. // Linguistic Databases, CSLI Lecture Notes #77.

[D. Sleator, D. Temperley, 1991] D. Sleator, D. Temperley. Parsing English with a Link Grammar. – CMU-CS-91-196, School of Computer Science, Carnegie Mellon University, Pittsburg, 1991.

[D. Grinberg, J. Lafferty, 1995] D. Grinberg, J. Lafferty. A robust parsing algorithm for Link Grammars. - CMU-CS-95-125, School of Computer Science, Carnegie Mellon University, Pittsburg, 1995.

[XRCE MLTT, 1995] Application of Finite-State Networks. // www.xrce.xerox.com/research

[Н. Леонтьева, 1995] Н. Н. Леонтьева. «Политекст»: информационный анализ политических текстов. // НТИ, Сер.2, 1995, №4.

[Н. Сущанская, 1989] Н. Ф. Сущанская. Программный препроцессор для естественноязыковых интерфейсов. - Автореф. дисс. к.т.н. – К.: РИО ИК, 1989.

[С. Гладунов, О. Федяев, 2002] С. А. Гладунов, О. И. Федяев. Распознавание речи на основе нейросетевой аппроксимации фонем. // КИИ-2002. Труды конференции, т.1 – М., Физматлит, 2002.

[Я. Тестелец, 2001] Я. Г. Тестелец. Введение в общий синтаксис. – М., РГГУ, 2001.

[С. Эйзенштейн, 2000] С. М. Эйзенштейн. Монтаж. – М., 2000.

[Х. Дрейфус, С. Дрейфус, 1998] Дрейфус Х., Дрейфус С. Создание сознания vs моделирование мозга. //Аналитическая философия: Становление и развитие. М., 1998.

[Д. Серл, 1998] Серл Д. Мозг, сознание и программы. //Аналитическая философия: Становление и развитие. М., 1998.

[Х. Патнэм, 1999] Патнэм Х. Философия сознания. //М., 1999.

[ВИНИТИ, 1990] Итоги науки и техники: физические и математические модели нейронных сетей, том 1, М., изд. ВИНИТИ, 1990.

[А. Кибрик, 2001] Кибрик А.Е. Очерки по общим и прикладным вопросам языкознания. – УРСС, М., 2001.

[Э. Сепир, 1993] Э. Сепир. Избранные труды по языкознанию и культорологии. //М., 1993.

[В. Ингве, 1965] В. Ингве. Гипотеза глубины. //Новое в лингвистике. Вып. 4, М., 1965 – с. 126.

[Б. Страуструп, 1999] Б. Страуструп. Язык программирования С++. – М., 1999.

[Н.А.Еськова, И.Г.Бидер и др.] Н.А.Еськова, И.Г.Бидер и др. Формальная модель русской морфологии.

[С.О.Шереметьева, С.Ниренбург, 1996] Эмпирическое моделирование в вычислительной морфологии. //НТИ, №7, 1996.

[Г.Г.Белоногов, 1984] Г.Г.Белоногов. Итоги науки и техники. Серия “Информатика”, т.№8,1984г.

[J. Goldsmith, 1999] J. Goldsmith. Unsupervised Learning of the Morphology of a Natural Language. //University of Chicago, 1998.

[И. Ножов, 2000] Ножов И.М. Прикладной морфологический анализ без словаря. // КИИ-2000. Труды конференции – М.: Физматлит, 2000. Т.1. С. 424-429

[И. Ножов, 2000] Ножов И.М. Процессор автоматизированного морфологического анализа без словаря. Деревья и корреляция. //Диалог'2000. Труды конференции - Протвино, 2000. Т.2. С. 284-290.

[А.Зализняк, 1980] Зализняк А.А. Грамматический словарь русского языка - М.: Русский язык, 1980 г.

[Ф.Харари, 1973] Ф.Харари. Теория графов. - М., 1973.

[Lauri Karttunen, 1993] Lauri Karttunen. Finite-State Lexicon Compiler. //Technical Report. ISTL-NLTT, Xerox Palo Alto Research Center, Palo Alto, California, 1993.

[Lauri Karttunen, 1994] Lauri Karttunen. Constructing Lexical Transducers. //15th International Conference on Computational Linguistics. Coling 94, I, pages 406-411. August 5-9, 1994. Kyoto, Japan.

[Finite-State Network, 1995] Finite-State Network. // Xerox Research Center, Grenoble, www.xrce.xerox.com/research

[Ж.Г.Аношкина, 1995] Ж.Г.Аношкина. Морфологический процессор русского языка. //Альманах «Говор», Сыктывкар, 1995, с.17-23.

[Ч. Хоккетт, 1961] Ч. Хоккетт. Грамматика для слушающего. // Новое в лингвистике. Вып. 4, М., 1965 – с. 139.

[S. Oepen, J. Caroll, 2000] S. Oepen, J. Caroll. Parser engineering and performance profiling. // Journal of Natural Language Engineering # 6 (1), 2000.

[Т. Кормен и др., 2001] Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритмы, построение и анализ. – М., МЦНМО, 2001.

[G. Neumann, J. Piskorski, 2001] G. Neumann, J. Piskorski. A Shallow Text Processing Core Engine. – DFKI, Saarbruecken, 2001.

[Дж. Фридл, 2001] Дж. Фридл. Регулярные выражения. – СПб., 2001.

[А. Сокирко, 2001] А. В. Сокирко. Семантические словари в автоматической обработке текста (по материалам системы Диалинг). - Автореф. дисс. к.т.н. – М., 2001.

[Д. Панкратов и др., 2000] Д. В. Панкратов, Л. М. Гершензон, И. М. Ножов. Описание фрагментации и синтаксического анализа в системе Диалинг. // Техническая документация, www.aot.ru, 2000.

[Т.Ю.Кобзарева, 2002] Т.Ю. Кобзарева. Некоторые аспекты анализа сочинения при сегментации русского предложения (неоднозначности при появлении

«матрешек»). // КИИ-2002. Труды конференции – М.: Физматлит, 2002. Т.1. С.192-198.

[И. Ножов, 2002] И.М. Ножов. Проектирование сегментационного анализатора русского предложения. // КИИ-2002. Труды конференции – М.: Физматлит, 2002. Т.1. С. 212-222.

[А. Белоусов, С. Ткачев, 2001] А. И. Белоусов, С. Б. Ткачев. Дискретная математика. - т.19, М.,2001.

[Т. Кобзарева и др., 2000] Т.Ю. Кобзарева, Д.Г. Лахути, И.М. Ножов. Сегментация русского предложения. // КИИ-2000. Труды конференции – М.: Физматлит, 2000. Т.1. С. 339-344.

[Т. Кобзарева и др., 2001] Т.Ю. Кобзарева, Д.Г. Лахути, И.М. Ножов. Модель сегментации русского предложения. // Диалог'2001. Труды конференции – Аксаково, 2001. Т.2. С. 185-194.

[Н. Вирт, 2001] Н. Вирт. Алгоритмы и структуры данных. – СПб., 2001.

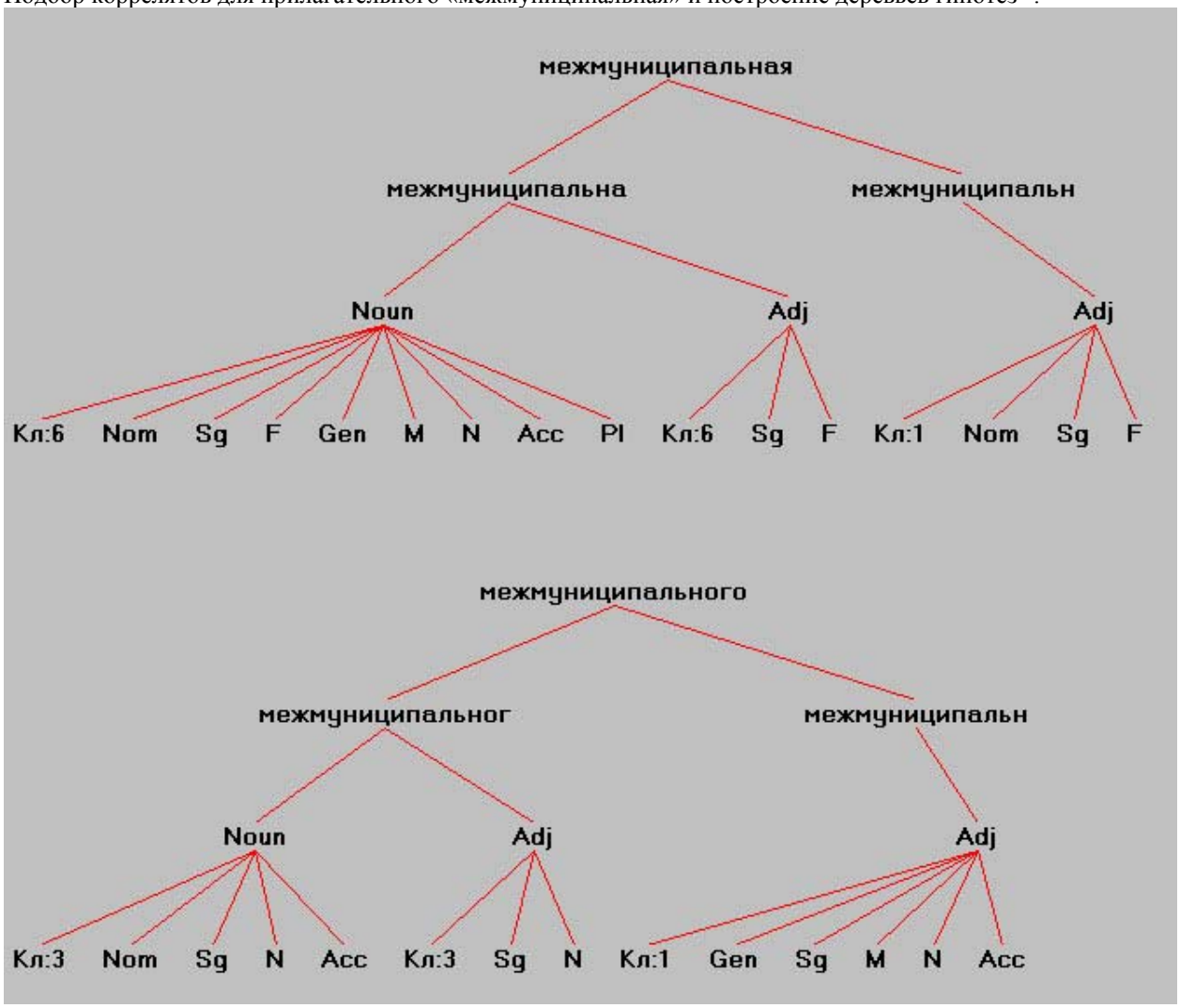
[И. Ножов, 2002] Ножов И.М. Синтаксический анализ. //Компьютерра, № 21 (446), 2002.

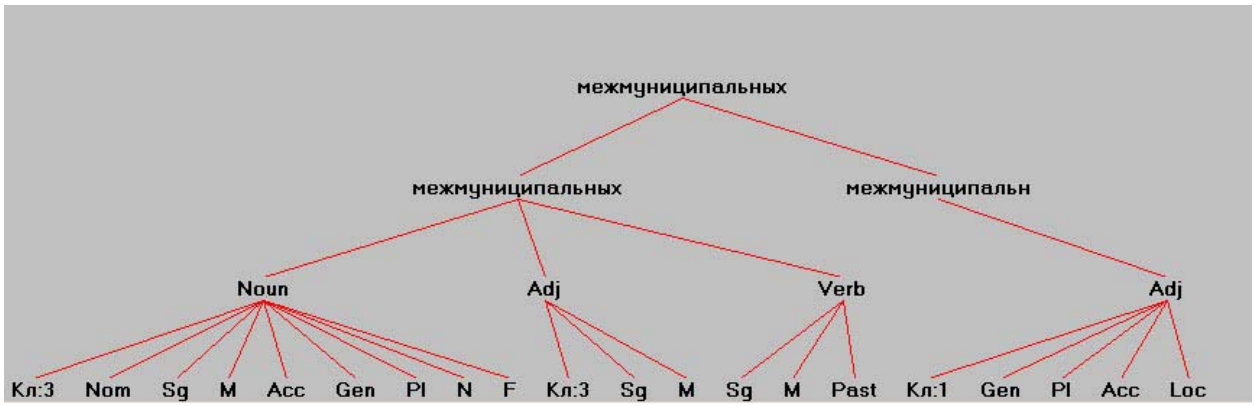
[Д. Кормалев, Е. Куршев и др., 2002] Кормалев Д.А., Куршев Е.П., Сулейманова Е.А., Трофимов И.В. Извлечение данных из текста. Анализ ситуаций ньюсмейкинга. // КИИ-2002. Труды конференции, т.1 – М., Физматлит, 2002.

[И. Мельчук, 1997] Курс общей морфологии - Т.№1, М., 1997.

ПРИЛОЖЕНИЕ 1. ПРИМЕРЫ РАБОТЫ МОРФОЛОГИЧЕСКИХ И ПРЕДСИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ

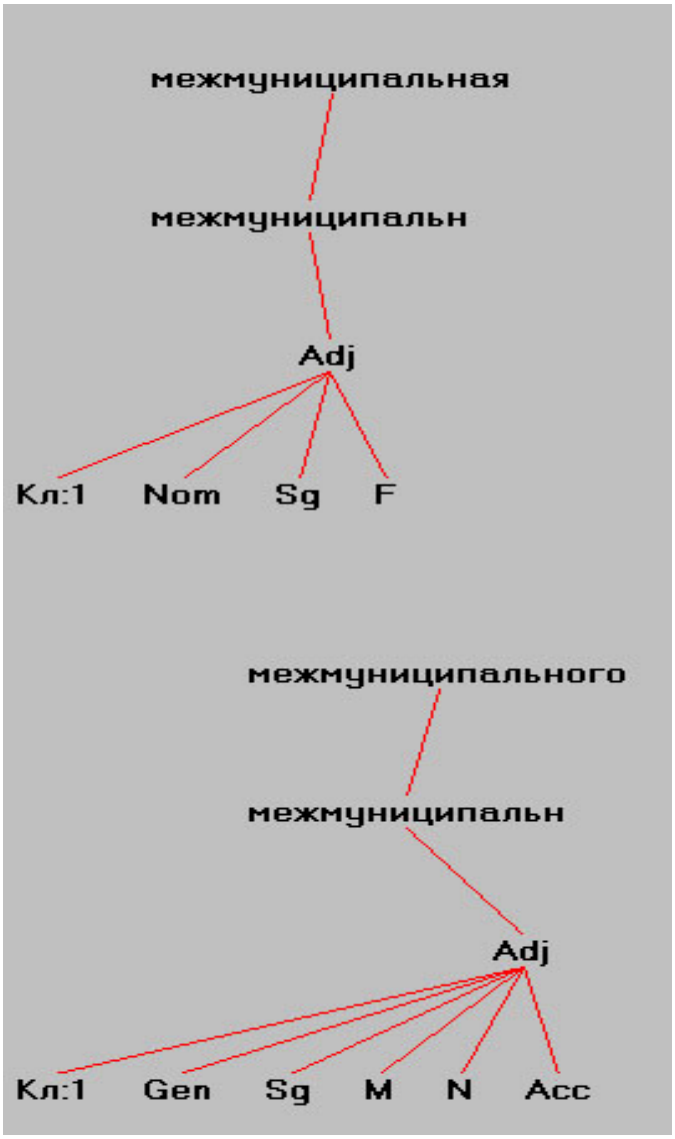
Морфологический анализ без словаря (НТЦ «Система»).
Подбор коррелятов для прилагательного «межмуниципальная» и построение деревьев гипотез[15]:

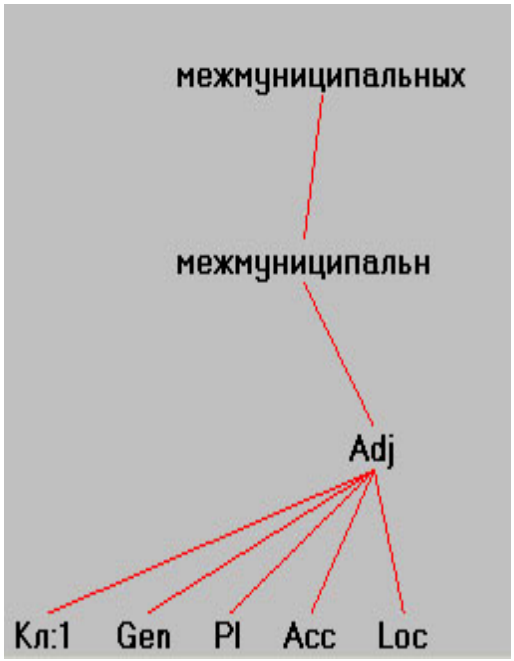




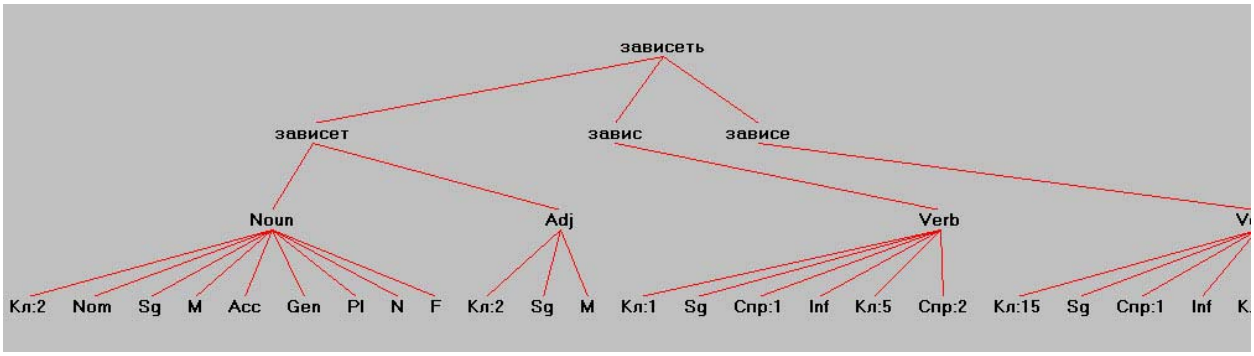
Результат корреляции (унификация гипотезы):

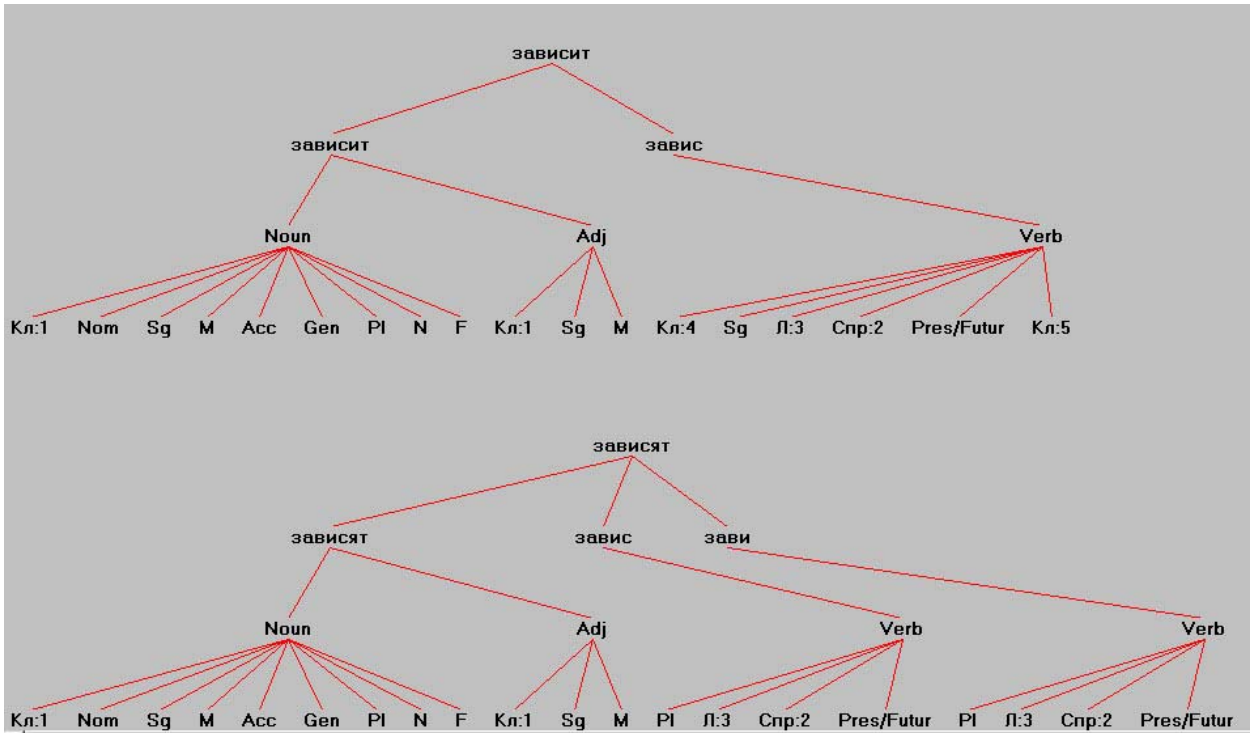
[15] ‘Кл:’ обозначает номер парадигматического класса.



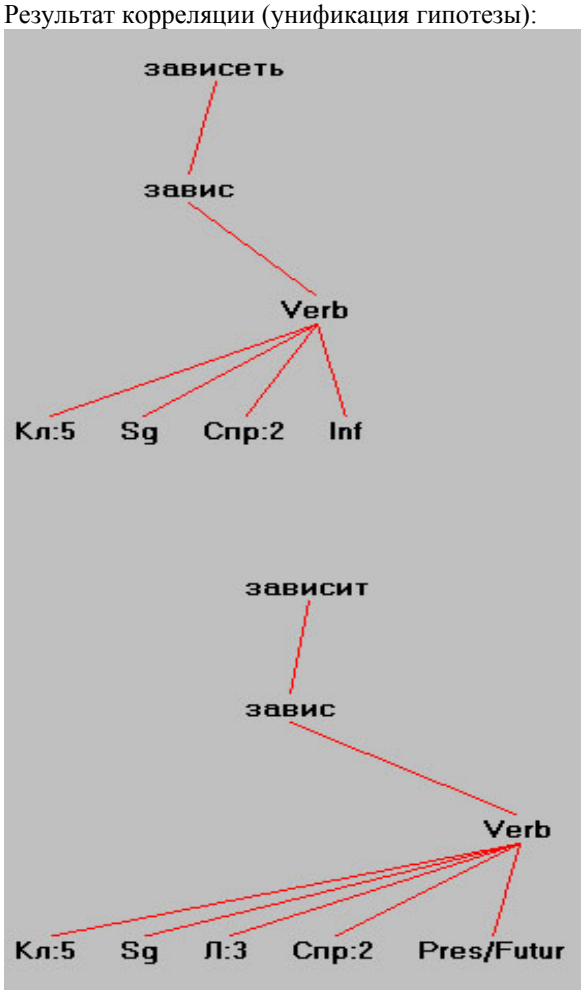


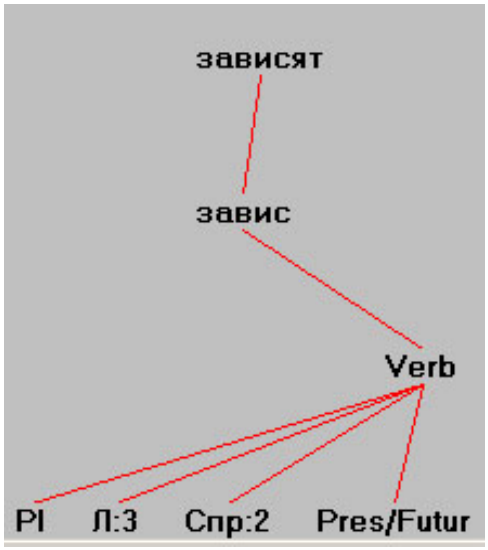
Подбор коррелятов для глагола «зависеть» и построение деревьев гипотез:



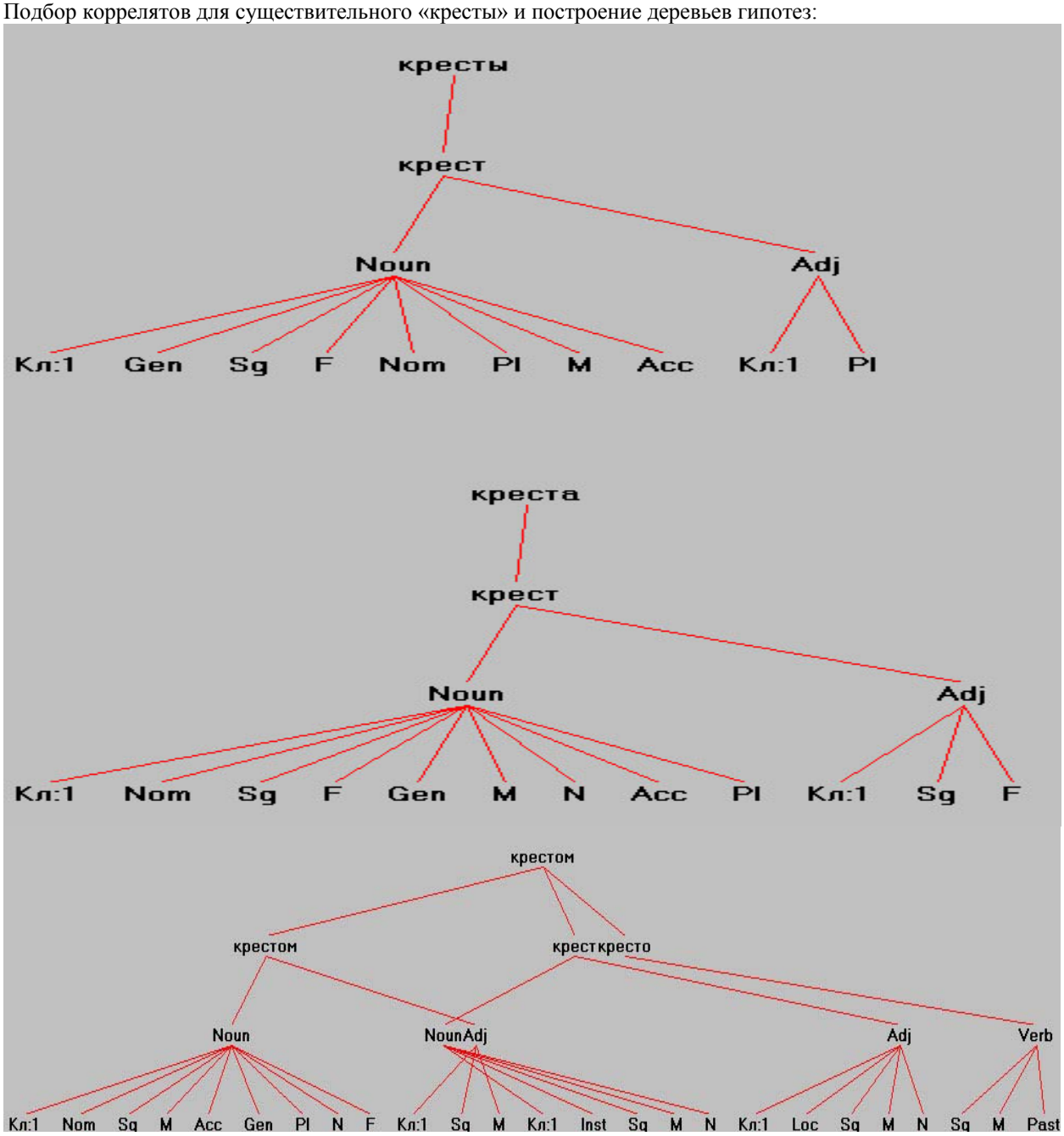


Результат корреляции (унификация гипотезы):

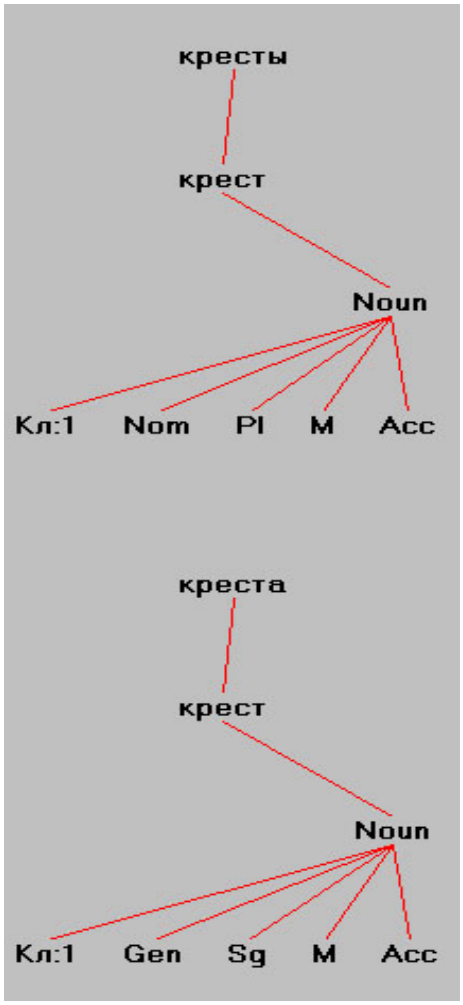


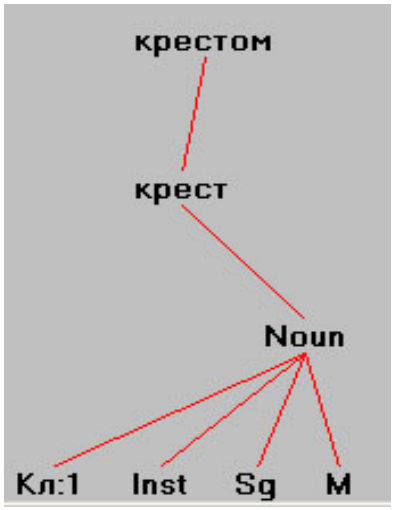


Подбор коррелятов для существительного «кресты» и построение деревьев гипотез:



Результат корреляции (унификация гипотезы):





Результаты анализа технического и финансового текста русскоязычной версией процессора LinguistX Platform:

Исходный текст:

Тип обслуживания используется для обозначения требуемой услуги. Тип обслуживания - это абстрактный или обобщенный набор параметров, который характеризует набор услуг, предоставляемых сетями, и составляющих собственно протокол Internet. Этот способ обозначения услуг должен использоваться шлюзами для выбора рабочих параметров передачи в конкретной сети, для выбора сети, используемой при следующем переходе датаграммы, для выбора следующего шлюза при маршрутизации сетевой датаграммы.

В 1960-х годах исследователи начали эксперименты по соединению компьютеров друг с другом с помощью телефонных линий, используя фонды Агентства Перспективных Проектов Исследований Министерства Обороны США.

Предыдущие попытки объединения компьютеров в сеть требовали наличия линии между двумя компьютерами сети, нечто вроде железнодорожной одноколейки. Пакетная система позволила создавать "шоссейные магистрали" для данных, по которым много машин движутся фактически в одном и том же ряду. Каждому пакету выдается компьютерный эквивалент карты и расписания, так что его можно направить в желательное место назначения, где все такие пакеты снова соберут в сообщение, пригодное для использования человеком или компьютером.

По мере того, как эта система, названная ARPANet, росла, несколько предприимчивых студентов колледжа разработали способ ее использования для проведения электронных конференций. Они начались как научные дискуссии, но скоро от них отпочковались

конференции практически по всем аспектам жизни, как только люди осознали возможность разговаривать с тысячами людей по всей стране.

2 июля в рамках программы, направленной на повышение информационной открытости, нефтяная компания подписала договор с американской компанией на проведение независимой оценки запасов нефти и газа. Завершение аудита запасов намечено на конец текущего - начало будущего года.

Результат анализа модуля tagger для первого абзаца исходного текста:

paragraph:

Тип	[Nn-Nom]	тип
обслуживания	[Nn-Gen]	обслуживание
используется	[Verb-Fin]	использовать
для	[Prep-Gen]	для
обозначения	[Nn-Gen]	обозначение
требуемой	[Verb-Gen]	требовать
услуги	[Nn-Gen]	услуга
.	[Punct-Sent]	.
Тип	[Nn-Acc]	тип
обслуживания	[Nn-Gen]	обслуживание
-	[Punct]	-
это	[Pron-Nom]	это
абстрактный	[Adj-Nom]	абстрактный
или	[Conj]	или
обобщенный	[Adj-Nom]	обобщенный
набор	[Nn-Nom]	набор
параметров	[Nn-Gen]	параметр
,	[Punct-Comma]	,
который	[Det-Nom]	который
характеризует	[Verb-Fin]	характеризовать
набор	[Nn-Acc]	набор
услуг	[Nn-Gen]	услуга
,	[Punct-Comma]	,
предоставляемых	[Verb-Gen]	предоставлять
сетями	[Nn-Obl]	сеть
,	[Punct-Comma]	,
и	[Conj]	и
составляющих	[Verb-Obl]	составлять
собственно	[Adv]	собственно
протокол	[Nn-Nom]	протокол
Internet	[Prop]	Internet
.	[Punct-Sent]	.
Этот	[Det-Acc]	этот
способ	[Nn-Acc]	способ
обозначения	[Nn-Gen]	обозначение
услуг	[Nn-Gen]	услуга
должен	[Adj-Brf]	должен | должный
использоваться	[Verb-Inf]	использовать
шлюзами	[Nn-Obl]	шлюз
для	[Prep-Gen]	для
выбора	[Nn-Gen]	выбор
рабочих	[Adj-Gen]	рабочий
параметров	[Nn-Gen]	параметр
передачи	[Nn-Gen]	передача
в	[Prep-Obl]	в
конкретной	[Adj-Obl]	конкретный
сети	[Nn-Obl]	сеть
,	[Punct-Comma]	,
для	[Prep-Gen]	для
выбора	[Nn-Gen]	выбор
сети	[Nn-Gen]	сеть

, [Punct-Comma] ,
используемой [Verb-Obl] использовать
при [Prep-Obl] при
следующем [Adj-Obl] следующий
переходе [Nn-Obl] переход
датаграммы [Nn-Gen] датаграмма
, [Punct-Comma] ,
для [Prep-Gen] для
выбора [Nn-Gen] выбор
следующего [Adj-Gen] следующий
шлюза [Nn-Gen] шлюз
при [Prep-Obl] при
маршрутизации [Nn-Obl] маршрутизация
сетевой [Adj-Gen] сетевой
датаграммы [Nn-Gen] датаграмма
. [Punct-Sent] .

Результат анализа исходного текста модулем np-grouper:

paragraph:
np: Тип(Nn-Nom) обслуживания(Nn-Gen)
np: обозначения(Nn-Gen) требуемой(Verb-Gen) услуги(Nn-Gen)
np: Тип(Nn-Acc) обслуживания(Nn-Gen)
np: абстрактный(Adj-Nom) или(Conj) обобщенный(Adj-Nom) набор(Nn-Nom) параметров(Nn-Gen)
np: набор(Nn-Acc) услуг(Nn-Gen)
np: сетями(Nn-Obl)
np: протокол(Nn-Nom)
np: способ(Nn-Nom) обозначения(Nn-Gen) услуг(Nn-Gen)
np: шлюзами(Nn-Obl)
np: выбора(Nn-Gen) рабочих(Adj-Gen) параметров(Nn-Gen) передачи(Nn-Gen)
np: конкретной(Adj-Obl) сети(Nn-Obl)
np: выбора(Nn-Gen) сети(Nn-Gen)
np: следующем(Adj-Obl) переходе(Nn-Obl) датаграммы(Nn-Gen)
np: выбора(Nn-Gen) следующего(Adj-Gen) шлюза(Nn-Gen)
np: маршрутизации(Nn-Obl) сетевой (Adj-Gen) датаграммы(Nn-Gen)

paragraph:
np: годах(Nn-Obl)
np: исследователи(Nn-Nom)
np: эксперименты(Nn-Acc)
np: соединению(Nn-Obl) компьютеров(Nn-Gen)
np: друг(Nn-Nom)
np: другом(Nn-Obl)
np: телефонных(Adj-Gen) линий(Nn-Gen)
np: фонды(Nn-Acc) Агентства(Nn-Gen) Перспективных(Adj-Gen) Проектов(Nn-Gen) Исследований(Nn-Gen) Министерства(Nn-Gen) Обороны(Nn-Gen) США(Prop-Gen)

paragraph:
heading: start=7167 end=7681
np: Предыдущие(Adj-Nom) попытки(Nn-Nom) объединения(Nn-Gen) компьютеров(Nn-Gen)
np: сеть(Nn-Acc)
np: наличия(Nn-Acc) линии(Nn-Gen)
np: компьютерами(Nn-Obl) сети(Nn-Gen)
np: железнодорожной(Adj-Gen) одноколейки(Nn-Gen)
np: Пакетная(Adj-Nom) система(Nn-Nom)
np: шоссейные(Adj-Nom) магистрали(Nn-Nom)
np: данных(Nn-Gen)
np: машин(Nn-Gen)
np: ряду(Nn-Obl)
np: Каждому(Adj-Obl) пакету(Nn-Obl)

np: компьютерный(Adj-Acc) эквивалент(Nn-Acc) карты(Nn-Gen) и(Conj) расписания(Nn-Gen)
np: желательное(Adj-Acc) место(Nn-Acc) назначения(Nn-Gen)
np: такие(Adj-Nom) пакеты(Nn-Nom)
np: сообщение(Nn-Acc)
np: использования(Nn-Gen)
np: человеком(Nn-Obl) или(Conj) компьютером(Nn-Obl)

paragraph:
np: система(Nn-Nom)
np: несколько(Num) предприимчивых(Adj-Gen) студентов(Nn-Gen) колледжа(Nn-Gen)
np: способ(Nn-Acc)
np: использования(Nn-Gen)
np: проведения(Nn-Gen) электронных(Adj-Gen) конференций(Nn-Gen)
np: научные(Adj-Nom) дискуссии(Nn-Nom)
np: конференции(Nn-Nom)
np: аспектам(Nn-Obl) жизни(Nn-Gen)
np: люди(Nn-Nom)
np: возможность(Nn-Acc)
np: тысячами(Num) людей(Nn-Gen)
np: стране(Nn-Obl)

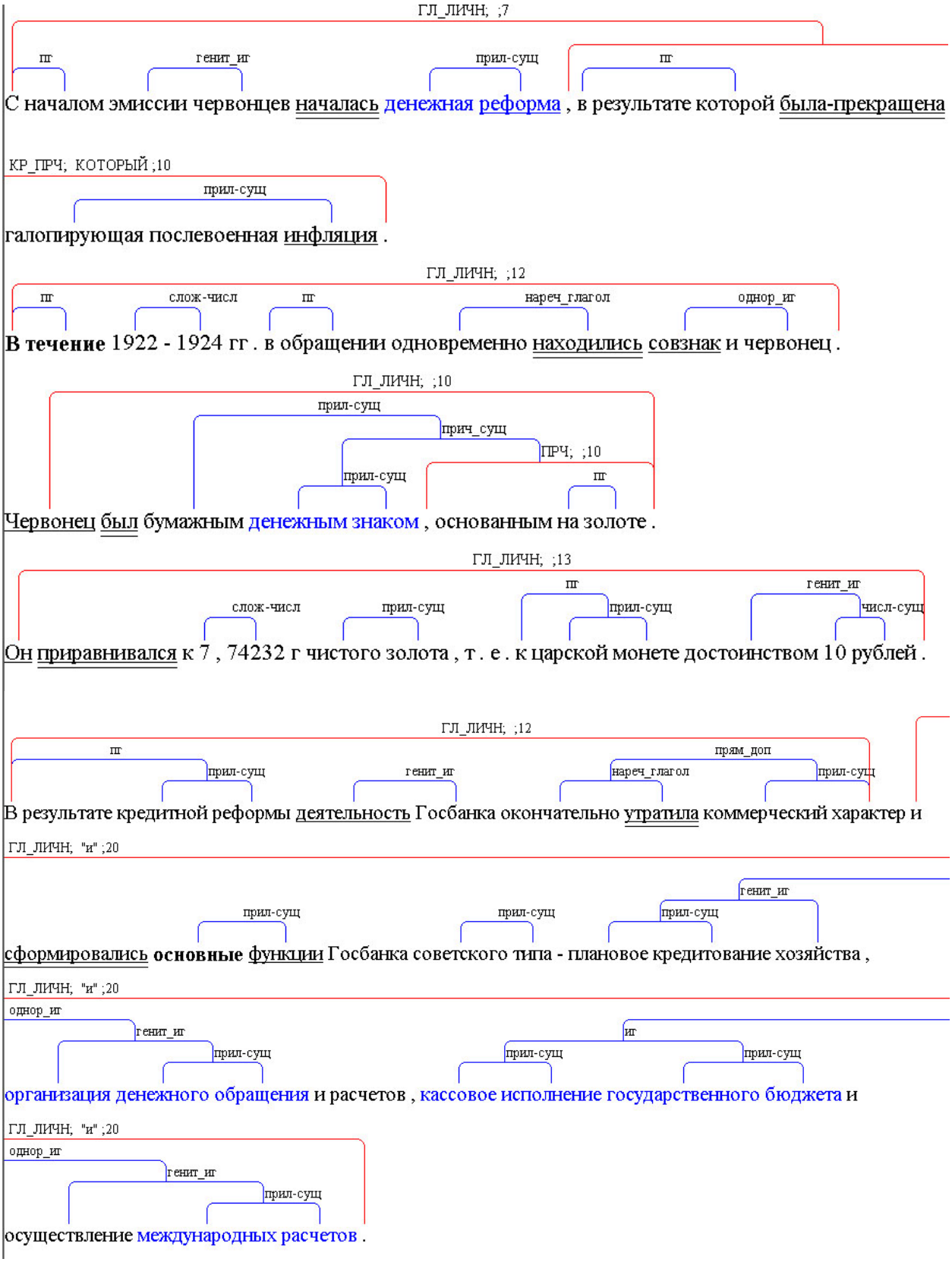
paragraph:
np: 2(Dig) июля(Nn-Gen)
np: рамках(Nn-Obl) программы(Nn-Gen)
np: повышение(Nn-Acc) информационной(Adj-Gen) открытости(Nn-Gen)
np: нефтяная(Adj-Nom) компания(Nn-Nom)
np: договор(Nn-Acc)
np: американской(Adj-Obl) компанией(Nn-Obl)
np: проведение(Nn-Acc) независимой(Adj-Gen) оценки(Nn-Gen) запасов(Nn-Gen) нефти(Nn-Gen) и(Conj) газа(Nn-Gen)
np: Завершение(Nn-Acc) аудита(Nn-Gen) запасов(Nn-Gen)
np: конец(Nn-Acc)
np: начало(Nn-Acc) будущего(Adj-Gen) года(Nn-Gen)

ПРИЛОЖЕНИЕ 2. ПРИМЕРЫ АНАЛИЗА СИНТАКСИЧЕСКИХ ПРОЦЕССОРОВ

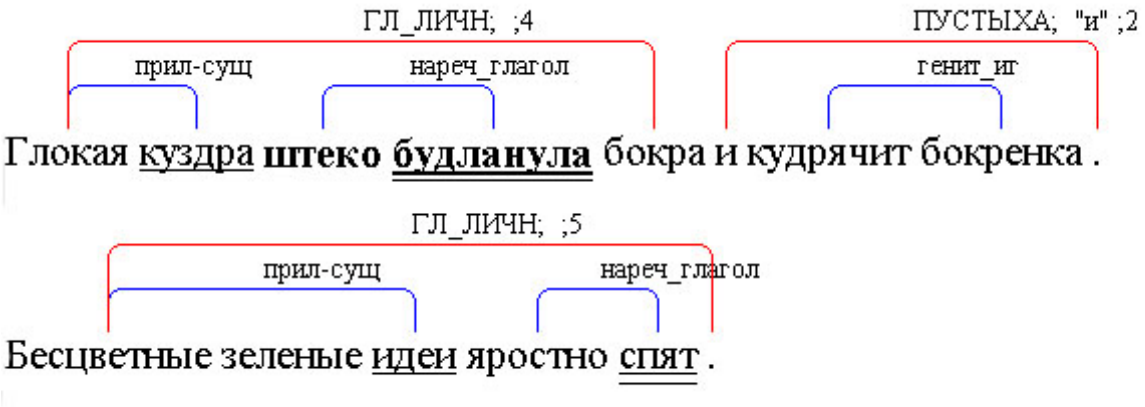
Результат анализа технического текста синтаксическим процессором Диалинг[16]:

[16] Голубым цветом в тексте выделены найденные в тезаурусах термины.

Результат анализа финансового текста синтаксическим процессором Диалинг:


ГЛ_ЛИЧН; ;7
пг
генит_иг
прил-сущ
пг
С началом эмиссии червонцев началась денежная реформа , в результате которой была-прекращена
КР_ПРЧ; КОТОРЫЙ ;10
прил-сущ
галопирующая послевоенная инфляция .
ГЛ_ЛИЧН; ;12
пг
слож-числ
пг
нареч_глагол
однор_иг
В течение 1922 - 1924 гг . в обращении одновременно находились совзнак и червонец .
ГЛ_ЛИЧН; ;10
прил-сущ
прич_сущ
ПРЧ; ;10
прил-сущ
пг
Червонец был бумажным денежным знаком , основанным на золоте .
ГЛ_ЛИЧН; ;13
пг
генит_иг
слож-числ
прил-сущ
прил-сущ
числ-сущ
Он приравнивался к 7 , 74232 г чистого золота , т . е . к царской монете достоинством 10 рублей .
ГЛ_ЛИЧН; ;12
пг
прям_доп
прил-сущ
генит_иг
нареч_глагол
прил-сущ
В результате кредитной реформы деятельность Госбанка окончательно утратила коммерческий характер и
ГЛ_ЛИЧН; "и" ;20
генит_иг
прил-сущ
прил-сущ
прил-сущ
сформировались основные функции Госбанка советского типа - плановое кредитование хозяйства ,
ГЛ_ЛИЧН; "и" ;20
однор_иг
генит_иг
прил-сущ
иг
прил-сущ
прил-сущ
организация денежного обращения и расчетов , кассовое исполнение государственного бюджета и
ГЛ_ЛИЧН; "и" ;20
однор_иг
генит_иг
прил-сущ
осуществление международных расчетов .

Ниже приводятся два классических примера разбора, демонстрирующих тезис о независимости синтаксической структуры предложения от смысла высказывания:



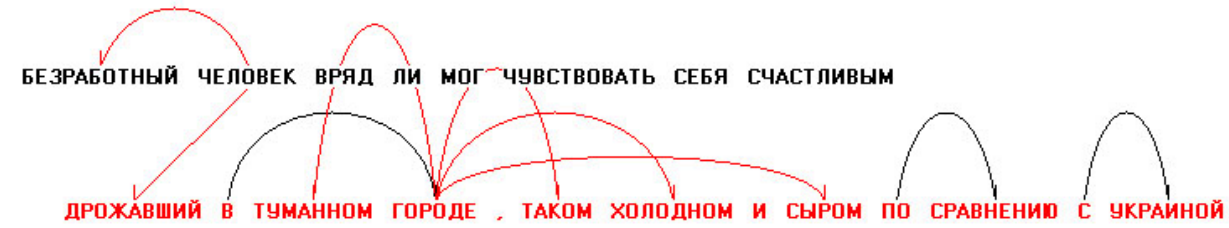
Примеры построены автоматически процессором Диалинг, с использованием процедуры морфологического предсказания (морфологический компонент Диалинг) для не найденных в словаре слов.

Приведем результаты анализа нескольких сложных предложений, содержащих различные грамматические трудности для построения структуры сегментов. Все приведенные ниже результаты получены экспериментальной системой ОИС[17].

Исходное предложение:

Безработный человек, дрожавший в туманном городе, таком холодном и сыром по сравнению с Украиной, вряд ли мог чувствовать себя счастливым.

Вариант 1:



Вариант 2:

Пример демонстрирует влияние морфологической омонимии как на граф синтагм, так и на граф сегментов: вариант 1 – ‘безработный’ прилагательное, вариант 2 – ‘безработный’ существительное, что приводит к появлению неморфологического предиката в сегменте “безработный человек”.

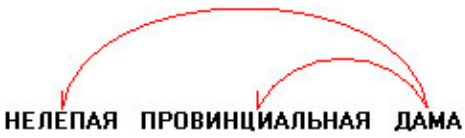
Исходное предложение:

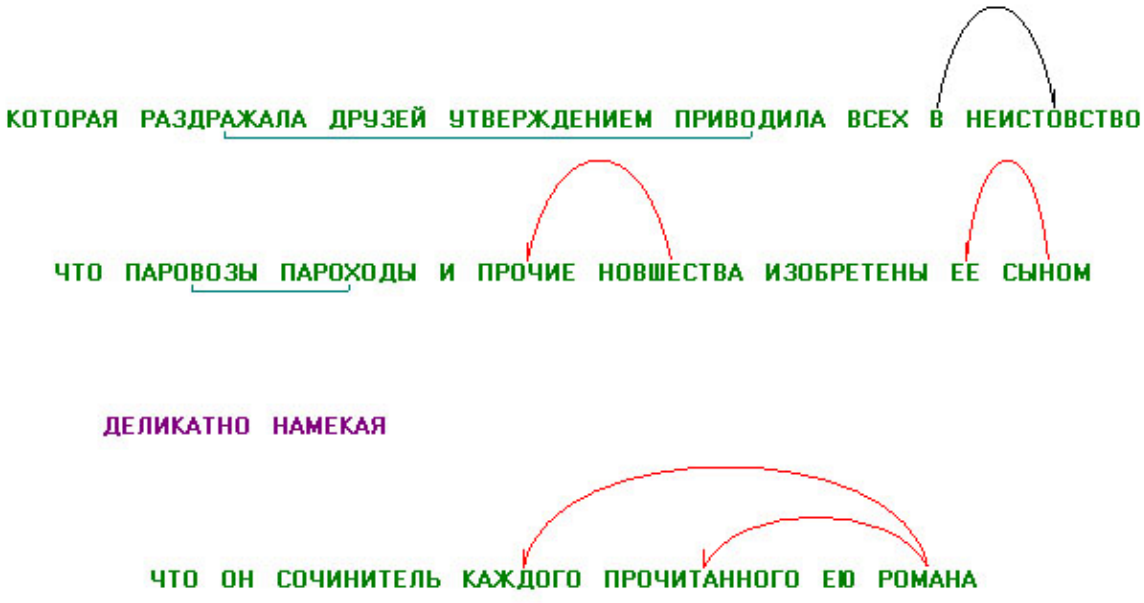
Нелепая провинциальная дама, которая раздражала друзей утверждением, что паровозы, пароходы и прочие новшества изобретены ее сыном, приводила всех в неистовство, деликатно намекая, что он сочинитель каждого прочитанного ею романа.

[17] Цвет сегмента маркирует его тип: черный цвет - β-сегмент; зеленый – SubS; фиолетовый – DvS; красный – AS; желтый – PS; серый – PrtS. Направление стрелки задает направление синтаксической связи – от главного к зависимому. Цвет стрелки маркирует тип синтагмы: черный – PRN; красный – NRA; синий – управление; голубой (нижняя скобка) – сочинение; желтый – предикат-субъект; зеленый – генитивное определение в постпозиции.

Вариант 1:

Вариант 2:

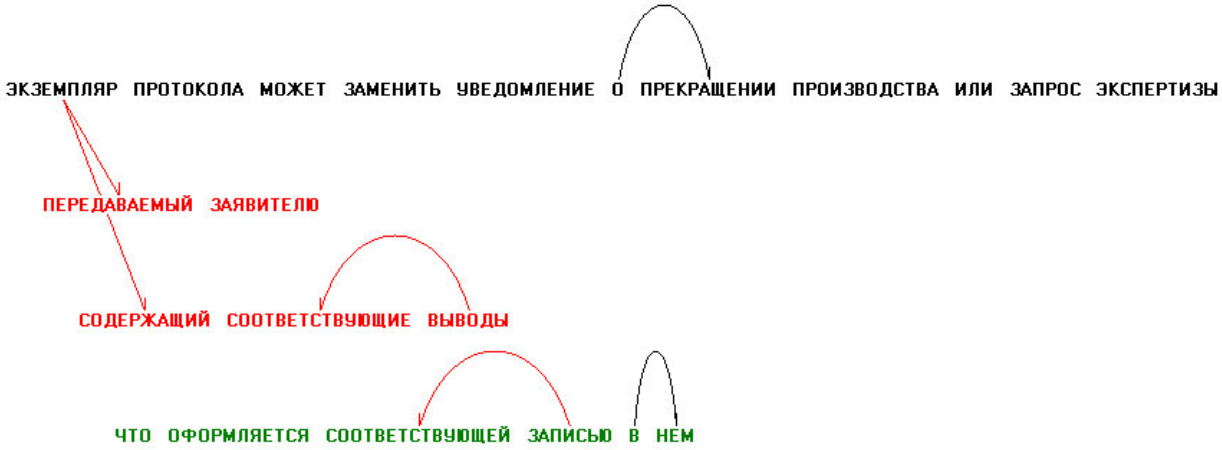




Пример демонстрирует случай синтаксической омонимии (второй вариант является синтаксически допустимой интерпретацией исходного предложения).

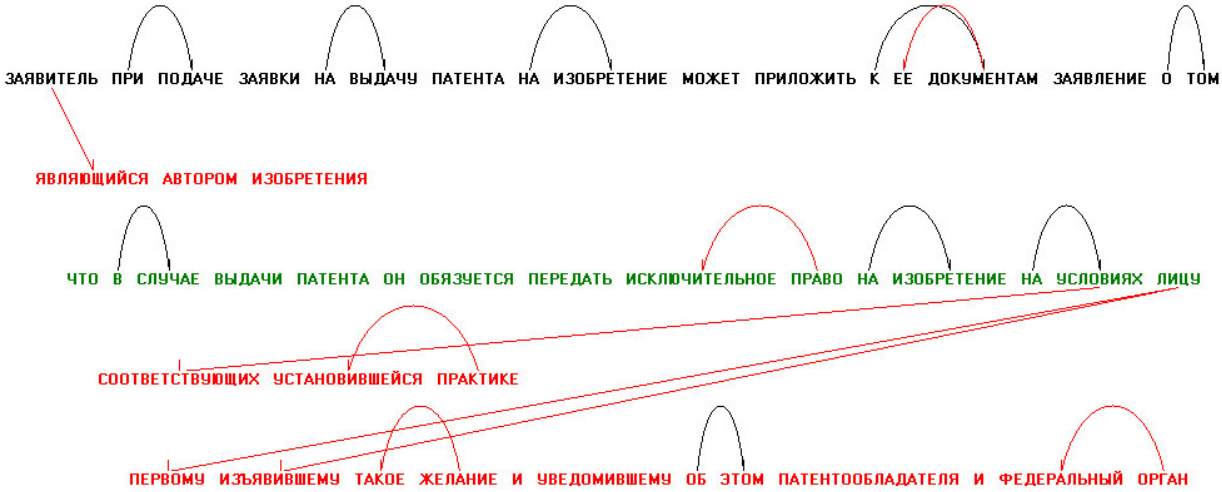
Исходное предложение:

Экземпляр протокола, передаваемый заявителю, содержащий соответствующие выводы, может заменить уведомление о прекращении производства или запрос экспертизы, что оформляется соответствующей записью в нем.



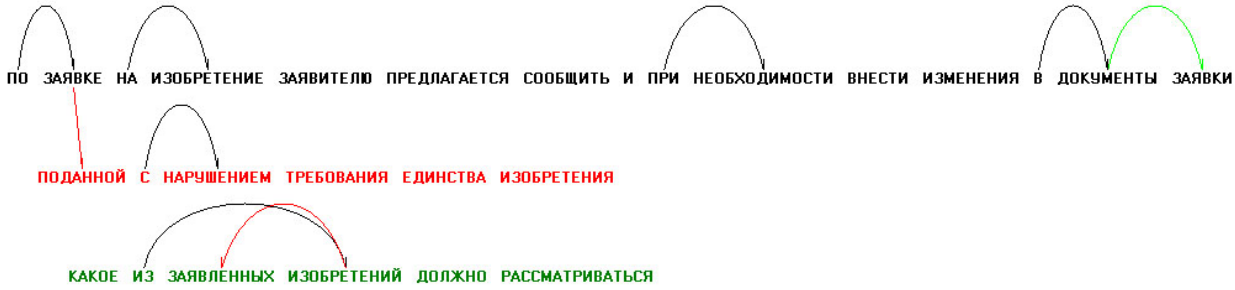
Исходное предложение:

Заявитель, являющийся автором изобретения, при подаче заявки на выдачу патента на изобретение может приложить к ее документам заявление о том, что в случае выдачи патента он обязуется передать исключительное право на изобретение на условиях, соответствующих установившейся практике, лицу, первому изъявившему такое желание и уведомившему об этом патентообладателя и федеральный орган.



Исходное предложение:

По заявке на изобретение, поданной с нарушением требования единства изобретения, заявителю предлагается сообщить, какое из заявленных изобретений должно рассматриваться, и при необходимости внести изменения в документы заявки.

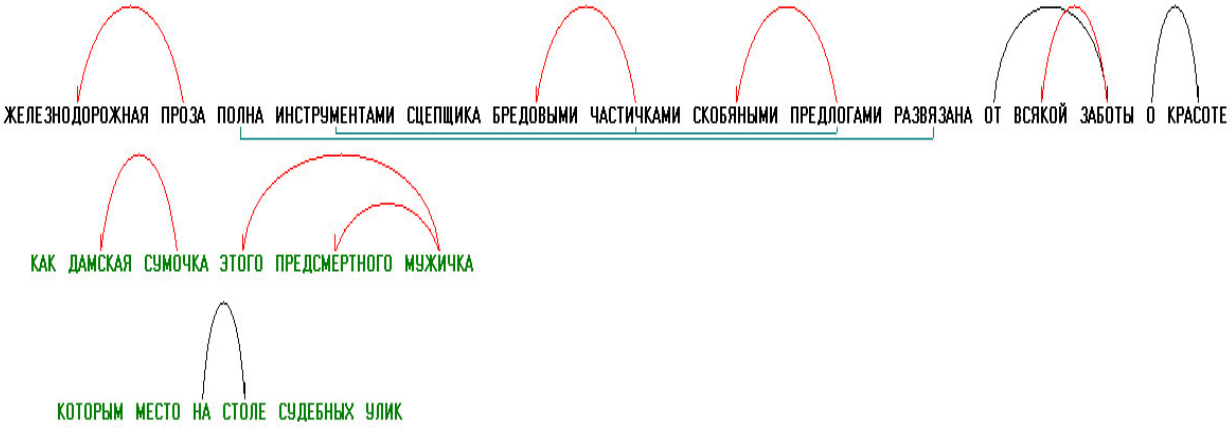


Исходное предложение:

Участники российских финансовых рынков, продавая рубли, старались минимизировать возможные негативные последствия углубления финансового кризиса, которые, как свидетельствует мировой опыт, проявляются в резком обесценении национальной валюты.

Исходное предложение:

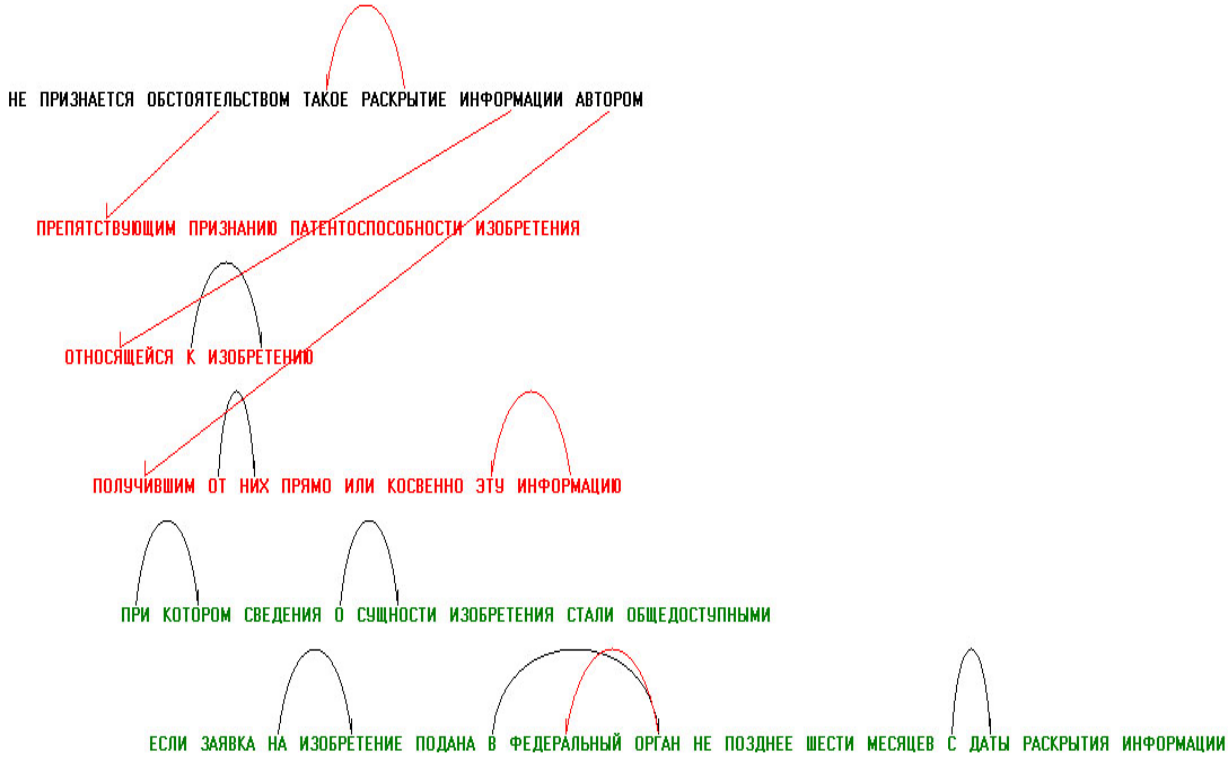
Железнодорожная проза, как дамская сумочка этого предсмертного мужичка, полна инструментами сцепщика, бредовыми частичками, скобяными предлогами, которым место на столе судебных улик, развязана от всякой заботы о красоте.



Пример демонстрирует случай вложенного сочинения.

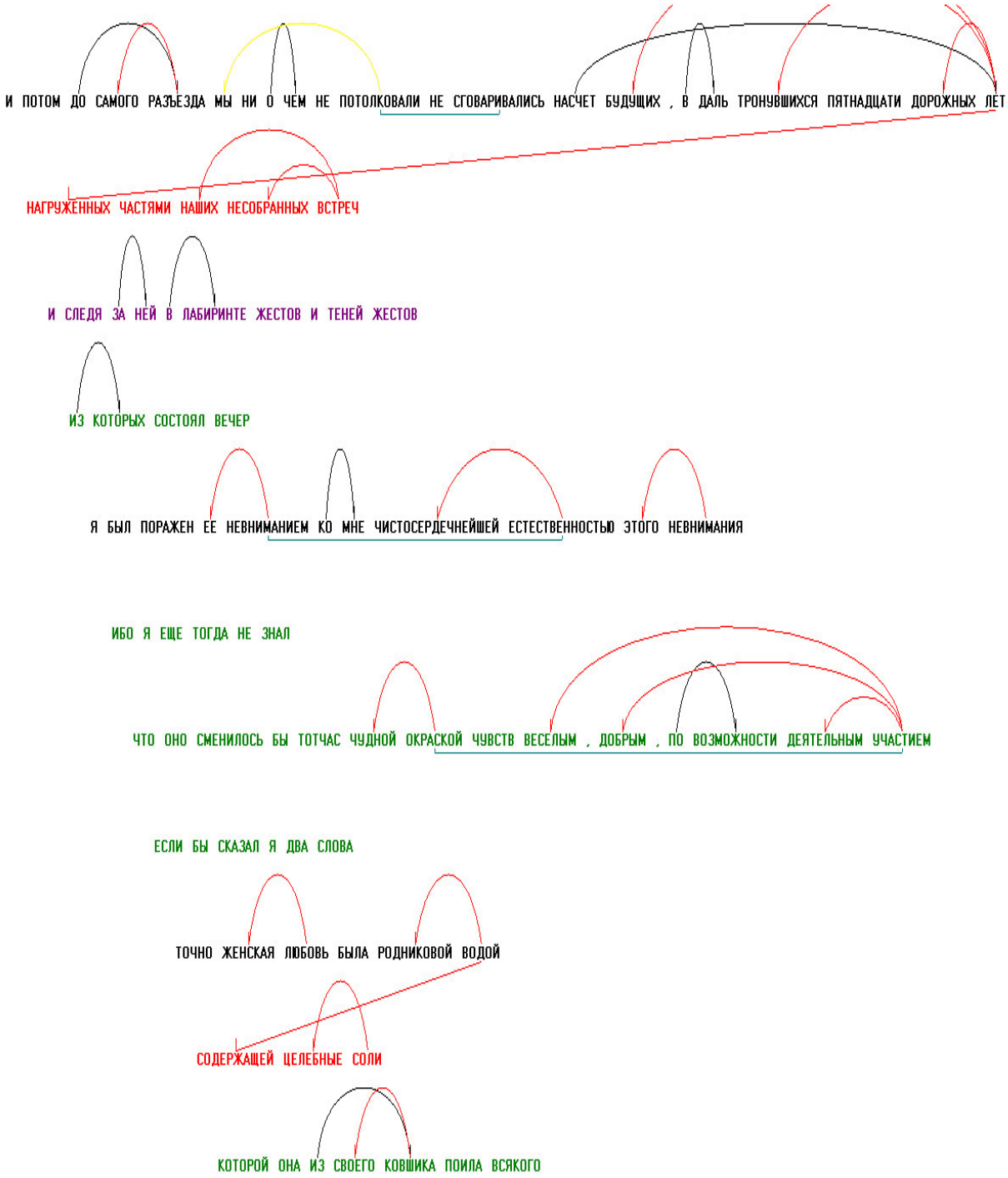
Исходное предложение:

Не признается обстоятельством, препятствующим признанию патентоспособности изобретения, такое раскрытие информации, относящейся к изобретению, автором, получившим от них прямо или косвенно эту информацию, при котором сведения о сущности изобретения стали общедоступными, если заявка на изобретение подана в федеральный орган не позднее шести месяцев с даты раскрытия информации.



Исходное предложение:

*И потом до самого разъезда мы ни о чем не потолковали, не сговаривались насчет будущих**,** в даль тронувшихся пятнадцати дорожных лет, нагруженных частями наших несобранных встреч, и следя за ней в лабиринте жестов и теней жестов, из которых состоял вечер, я был поражен ее невниманием ко мне, чистосердечнейшей естественностью этого невнимания, ибо я еще тогда не знал, что, если бы сказал я два слова, оно сменилось бы тотчас чудной окраской чувств, веселым, добрым, по возможности деятельным участием, точно женская любовь была родниковой водой, содержащей целебные соли, которой она из своего ковшика поила всякого, если только напомнить.*


И ПОТОМ ДО САМОГО РАЗЪЕЗДА МЫ НИ О ЧЕМ НЕ ПОТОЛКОВАЛИ НЕ СГОВАРИВАЛИСЬ НАСЧЕТ БУДУЩИХ , В ДАЛЬ ТРОНУВШИХСЯ ПЯТНАДЦАТИ ДОРОЖНЫХ ЛЕТ
НАГРУЖЕННЫХ ЧАСТЯМИ НАШИХ НЕСОБРАННЫХ ВСТРЕЧ
И СЛЕДЯ ЗА НЕЙ В ЛАБИРИНТЕ ЖЕСТОВ И ТЕНЕЙ ЖЕСТОВ
ИЗ КОТОРЫХ СОСТОЯЛ ВЕЧЕР
Я БЫЛ ПОРАЖЕН ЕЕ НЕВНИМАНИЕМ КО МНЕ ЧИСТОСЕРДЕЧНЕЙШЕЙ ЕСТЕСТВЕННОСТЬЮ ЭТОГО НЕВНИМАНИЯ
ИБО Я ЕЩЕ ТОГДА НЕ ЗНАЛ
ЧТО ОНО СМЕНИЛОСЬ БЫ ТОТЧАС ЧУДНОЙ ОКРАСКОЙ ЧУВСТВ ВЕСЕЛЫМ , ДОБРЫМ , ПО ВОЗМОЖНОСТИ ДЕЯТЕЛЬНЫМ УЧАСТИЕМ
ЕСЛИ БЫ СКАЗАЛ Я ДВА СЛОВА
ТОЧНО ЖЕНСКАЯ ЛЮБОВЬ БЫЛА РОДНИКОВОЙ ВОДОЙ
СОДЕРЖАЩЕЙ ЦЕЛЕБНЫЕ СОЛИ
КОТОРОЙ ОНА ИЗ СВОЕГО КОВШИКА ПОИЛА ВСЯКОГО
ЕСЛИ ТОЛЬКО НАПОМНИТЬ