ЛЕКСИКО-СИНТАКСИЧЕСКИЕ ШАБЛОНЫ В ЗАДАЧАХ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТА*

LEXICOSYNTACTIC PATTERNS FOR AUTOMATIC TEXT PROCESSING

Большакова Е.И. (bolsh@cs.msu.su)

Баева Н.В. (baeva@recyclebin.ru)

Бордаченкова Е.А. (lenabord@mail.ru)

Васильева Н.Э. (nvasil@list.ru)

Морозов С.С. (sergej_morozov@rambler.ru)

Московский государственный университет им. М.В. Ломоносова, факультет ВМиК

Анализируются способы декларативного описания фрагментов текста на естественном языке, используемые для распознавания языковых конструкций в ходе поверхностного синтаксического анализа. Обсуждается понятие лексико-синтаксического шаблона языковых конструкций и характеризуется предложенный язык записи шаблонов, позволяющий задавать лексические и грамматические свойства входящих в него элементов.

Введение

Существенную поддержку в проведении лингвистических исследований оказывают программные средства, позволяющие автоматически находить в исследуемых текстах нужные единицы (словоформы, словосочетания и др.) и сохранять их конкорданс. Эту задачу отчасти решают пользовательские интерфейсы корпусов текстов – специальные программы, которые выполняют поиск текстовых единиц, используя сделанную заранее лингвистическую разметку текстов корпуса. Такое программное средство было создано для Национального корпуса русского языка (НКРЯ) [4]; оно позволяет искать в корпусе слова и их комбинации по некоторым грамматическим и лексико-семантическим характеристикам.

Однако любой размеченный корпус ограничен по предметным областям, жанрам и времени создания представленных в нем текстов. Неизбежно возникает потребность исследовать тексты, пока не включенные в корпусы, а для этого нужны уже другие программные средства. В них хотелось бы иметь возможность поиска более сложных единиц, чем словоформы, например, согласованных именных словосочетаний. Последнее требует реализации не только автоматического морфологического анализа слов предложений, но и выявления их синтаксических связей, т.е. решения задач, обычных для систем автоматической обработки текстов на основе поверхностного синтаксиса.

Из отечественных программных систем, используемых для поддержки лингвистических работ при построении специализированных систем автоматической обработки текстов, следует указать систему Alex [3]. Пользователем системы предполагается не лингвист, а эксперт в предметной области обрабатываемых документов, в чьи задачи входит описание лексики узкоспециализированных текстов – писем, объявлений и т.п. Система предоставляет довольно гибкие средства описания слов и словосочетаний в виде их шаблонов, которые используются затем для автоматического распознавания этих единиц в тексте. Alex расширяет возможности традиционных лексикографических систем, однако в его языке записи шаблонов нет встроенных средств для указания грамматических признаков распознаваемых лексических единиц.

Более универсальные средства описания свойств текстовых единиц реализованы в инструментальной системе GATE [7], также предназначенной для построения систем автоматической обработки текста, ориентированных на извлечение из текста информативных объектов (географических названий, персоналий и т.п.). Внутренний язык JAPE позволяет записывать правила распознавания текстовых объектов и определения их атрибутов (лексико-грамматических, семантических). Гибкость этой системы определяется принятой в ней атрибутно-объектной моделью текста, которая поддерживается объектно-ориентированным языком программирования JAVA. Однако в системе не предусмотрены средства для обработки высоко флективных языков (поскольку изначально она создавалась для английского), что требует программирования дополнительных модулей морфологического анализа и анализа словосочетаний. Именно так на основе GATE было построено семейство систем извлечения информации из деловых текстов [6].

Среда GATE и ее язык JAPE послужили прототипом при создании отечественного программного средства RCO Pattern Extractor [2], ориентированного на выделение специфических текстовых объектов (дат, адресов, имен юридических лиц и т.п.). В нем также использована атрибутно-объектная модель текста, а грамматика языка JAPE

* Работа выполняется при финансовой поддержке РФФИ (проект № 06-01-00571)

взята за основу и расширена рядом средств для работы с русскими текстами. Правило распознавания текстового объекта состоит из левой части (называемой образцом объекта) и правой части, в которой вычисляются нужные атрибуты распознанного объекта. Язык записи правил достаточно гибок, но больше похож на язык программирования, кроме того, его применение предполагает освоение атрибутно-объектной модели текста.

Рассмотренные программные системы различаются по своим функциям, что не могло не отразиться на наборе используемых в них средств декларативного описания текстовых единиц. В настоящей работе приводятся результаты сравнительного анализа средств данных систем и обсуждается концепция лексико-синтаксического шаблона языковых конструкций, предложенная для записи их лексических и грамматических особенностей. Описываются основные возможности разрабатываемого в настоящее время языка записи лексико-синтаксических шаблонов, называемого далее LSPL (LexicoSyntactic Pattern Language). Эти возможности покрывают основные потребности поиска текстовых единиц на основе поверхностного синтаксического анализа текста и сопоставимы со средствами других систем. В то же время язык предлагает новую концепцию записи лексико-синтаксических свойств, в том числе – контекстно-зависимых.

Способы описания фрагментов текста

Распознавание текстовых единиц по их лексическим и грамматическим особенностям невозможно без реализации морфологического анализа слов и частичного синтаксического анализа предложений, что неизбежно делает соответствующие системы языковозависимыми. По этой причине мы сравниваем возможности декларативного описания текстовых единиц только для отечественных систем, созданных в последние годы и предназначенных для обработки русских текстов – систему Alex, RCO Pattern Extractor и пользовательский интерфейс Национального корпуса русского языка.

В этих системах для выделения фрагментов текста описываются их свойства, относящиеся к различным уровням анализа текста – графематическому, морфологическому, лексическому, синтаксическому. Системы отличаются в основном степенью охвата этих уровней и возможностью комбинирования (при помощи логических операций НЕ, ИЛИ, И) разных свойств. Возможности описания нужных единиц текста в сравниваемых системах представлены в Таблице 1.

Выразительные возможности		НКРЯ	RCO	Alex	LSPL
лексико-графические единицы	символьная строка	–	+	+	+
	словоформа	+	+	+	+
	лексема	+	+	–	+
морфо-синтаксические условия	часть речи	+	+	–	+
	морфологические характеристики	+	+	–	+
	согласование	–	–	–	+
логические операции	лексико-графические единицы	+	+	+	–
	морфо-синтаксические условия	+	+	–	–
запись конструкций	именование	–	+	+	+
	альтернативы	–	+	+	+
	повторение	–	+	+	+

Таблица 1. Сравнение средств описания фрагментов текста

Конкретная словоформа легко задается и находится во всех рассматриваемых системах. Простейшая же возможность искать произвольную символьную строку, включающую как буквенные, так и небуквенные символы (имеющаяся во всех редакторах текстов) реализована в полной мере уже не везде – НКРЯ не производит поиск строк со знаками препинания. Более сложное, но нужное средство – поиск любой допустимой словоформы данной лексемы – есть в НКРЯ и RCO, но не доступно (без предварительного описания шаблона всех словоформ) в системе Alex.

Возможность использовать при выделении текстовых единиц значения их морфологических характеристик (часть речи, падеж, число, род, лицо, время) есть в НКРЯ и RCO. Но грамматическое согласование нескольких единиц, необходимое для однозначного выделения некоторых конструкций (например, именных групп), нельзя непосредственно записывать ни в одной из трех указанных систем.

Все три системы допускают логическое комбинирование условий: Alex – только лексических, а RCO и НКРЯ – как лексических, так и морфо-синтаксических (например, в НКРЯ возможен поиск существительного в именительном ИЛИ в винительном падеже). Дополнительно, НКРЯ разрешает указание лексико-семантических свойств искомых слов (разряд, тематический класс и т.п.) и комбинирование таких условий с другими (такой поиск возможен за счет разметки корпуса).

Важным выразительным средством являются допускаемые в Alex и RCO операции регулярных выражений, служащие для записи альтернативных вариантов и повторяющихся конструкций (в Таблице 1 они указаны как альтернативы и повторения). К примеру, в Alex синонимические варианты слова проспект могут быть записаны альтернативами: `проспект V пр-кт V пр-т V просп.`. В RCO операция регулярного повторения * позволяет

описать, например, строку из произвольного количества букв А: `"A*"`; при помощи этой же операции выражение `({!Token.Text == ""})*` задает повторение любой лексемы произвольное число раз или вообще ее отсутствие.

Полезной возможностью является именование конструкций: в Alex – шаблонов, в RCO – образцов объектов. Именование позволяет определять на основе уже известных шаблонов (образцов) шаблоны (образцы) все более сложных объектов и тем самым постепенно наращивать их мощность.

К дополнительным возможностям отнесем запись контекстных условий, позволяющих искать нужный объект по его контексту. В явном виде это есть только в системе Alex, в RCO же эта возможность реализуется использованием меток, приписываемых частям образца и используемых потом в правой части правил.

В последнем столбце Таблицы 1 охарактеризованы возможности разрабатываемого нами языка LSPL, основанного на понятии лексико-синтаксического шаблона. Как видим, он проигрывает рассмотренным системам лишь в средствах логического комбинирования, а несомненное его преимущество – возможность записи условий грамматического согласования текстовых единиц.

Концепция лексико-синтаксического шаблона

Наша концепция лексико-синтаксического шаблона сформировалась в ходе изучения терминологических и дискурсивных особенностей научно-технической прозы с целью отобразить их в словарных компонентах системы автоматической обработки научно-технических текстов [1]. В процессе этой работы возникла потребность формализовать и представить в словаре системы характерные для научной прозы конструкции (например, *Под T будем понимать D, Далее докажем P, Допустим, что S*). Формализация подобной конструкции предполагает определение множества входящих в нее лексем и их возможных грамматических форм, а также выявление необходимых синтаксических условий (например, условия согласования грамматических характеристик лексем). Эту информацию можно зафиксировать в виде некоторой декларативной структуры, которую мы назвали лексико-синтаксическим шаблоном языковой конструкции. По сути, лексико-синтаксический шаблон – это структурный образец языковой конструкции, который отображает ее лексические и поверхностно-синтаксические свойства.

Для формирования набора шаблонов характерных конструкций научной прозы требовалось провести дополнительное исследование совокупности именно русских научно-технических текстов (скудно представленных пока в размеченных корпусах русских текстов) с целью уточнения возможных вариантов, состава и грамматических особенностей конструкций. Проведение такого лингвистического исследования – достаточно трудоемкая задача и желательно автоматизировать ее решение, например, с помощью программного средства, позволяющего находить в исследуемых текстах нужные конструкции по некоторым их лексико-синтаксическим свойствам, т.е. по их шаблонам.

Таким образом возникла задача разработки языка записи лексико-синтаксических шаблонов, который был бы пригоден равно как

- способ формальной записи специфических языковых конструкций для их представления в системе автоматической обработки научно-технических текстов;
- язык записи запросов на поиск исследуемых конструкций в текстах, формулируемых на основе их словарного состава и несложных грамматических условий.

При решении этой задачи мы старались отобрать в язык шаблонов выразительные средства, позволяющие достаточно гибко записывать лексико-графические единицы (символьные строки, словоформы, лексемы) и их грамматические характеристики, причем в более простой и явной форме – так, чтобы результирующее описание свойств некоторой языковой конструкции соответствовало интуитивному понятию образца этой конструкции, а включенные в язык грамматические характеристики (например, часть речи) были доступны для понимания не только лингвисту. Для этого мы учли удачные стороны функционального языка программирования Рефал [5], разработанного для синтаксического анализа программных конструкций и предложившего свою оригинальную концепцию выражения-образца.

Отметим, что потребность в распознавании специфических конструкций и записи их в виде лексико-грамматических шаблонов возникала неоднократно, в частности, при решении задачи автоматического выявления лексико-семантических связей между словами для их представления в системе WordNet [8]. Языковые конструкции, сигнализирующие такую связь, записывались в виде лексико-синтаксических образцов. Наша концепция лексико-синтаксического шаблона идейно близка к рассмотренной в [8], однако нами она детально проработана с учетом специфики русского языка и воплощена в формальном языке шаблонов.

Выразительные возможности LSPL-шаблонов

Основным элементом лексико-синтаксического шаблона языка LSPL (и простейшим шаблоном) является *элемент-слово*, соответствующий отдельному слову текста и описывающий конкретную словоформу, множество словоформ конкретной лексемы, обладающих фиксированными морфологическими характеристиками, или же произвольное слово заданной части речи, стоящее в определенной грамматической форме.

В элементе-слове латинской буквой указывается часть речи слова, а в угловых скобках записываются нужная лексема и указываются нужные значения грамматических признаков, соответствующих данной части речи. При этом используются символьные обозначения: для частей речи – `N` (существительное), `V` (глагол), `A` (прилагательное) и т.д., для грамматических характеристик и их значений – `c` (падеж), `n` (число), `g` (род), `t` (время), `p` (лицо) и др.. Например,

шаблон A<важный; c=nom, g=fem> описывает слова *важная*, *важна*, поскольку в нем указан именительный падеж и женский род, а признаки формы прилагательного (полное/краткое) не фиксированы.

Для задания словоформы требуется задать морфологические характеристики полностью: шаблон N<теорема; c=ins, n=plur> соответствует словоформе *теоремами*. Заметим, что конкретную словоформу язык позволяет задать и в виде символьной строки так: "теоремами", однако такое задание исключает возможность последующего согласования с этой словоформой других элементов шаблона.

Опустив все грамматические характеристики в элементе-слове, мы получим шаблон, соответствующий любой словоформе лексемы: например, A<важный>.

Элемент-слово может также описывать произвольное слово определенной части речи, обладающее нужными морфологическими характеристиками, например, глагол настоящего времени третьего лица множественного числа задается шаблоном так: V<t=pres, p=3, n=plur>. Если же не нужны конкретизации характеристик, то можно записать V<> или просто V.

В общем случае в шаблон могут входить как несколько элементов-слов разных частей речи, так и несколько разных слов одной части речи. Для различения элементов-слов одной части речи используются индексы, например, шаблон N1 N2 описывает два стоящих рядом существительных. Чаще всего индексы нужны при задании условий согласования.

Для описания последовательностей одинаковых элементов шаблона служит конструкция повторения – фигурные скобки, например: шаблон {A}N описывает последовательность прилагательных (возможно, пустую), за которой идет существительное (например, *большие красные полосатые листья*). Если известны ограничения на количество однотипных элементов, то их можно указать в шаблоне непосредственно за конструкцией повторения в угловых скобках: шаблон {A}<1,3> N задает последовательность из одного, двух или трех прилагательных и существительного.

Элементы, входящие в описываемую конструкцию опционально, указываются в шаблоне в квадратных скобках. В шаблоне {A} N ["не"] V частица *не* перед глаголом указана как факультативная.

Можно задавать альтернативные варианты некоторой конструкции, для этого используется символ |, например, шаблон AP = A|Pa описывает понятие адъектива, т.е. прилагательного (A) или причастия (Pa).

Важная особенность нашего языка – возможность задавать грамматическое согласование элементов шаблона. Условия согласования относятся ко всему шаблону в целом и поэтому они указываются после всех элементов шаблона в виде равенства значений согласуемых морфологических признаков (в угловых скобках). Например, в шаблоне A<тяжелый> N <A.g=N.g, A.n=N.n, A.c=N.c> записано условие, что слово *тяжелый* и следующее за ним существительное согласованы в роде, числе и падеже. Этот шаблон описывает комбинации *тяжелым вечером*, *тяжелых камней*, *тяжелое тело* и многие другие. Если нужно указать согласование по всем общим морфологическим признакам, то его можно записать короче: A<тяжелый> N <A=N>.

При создании шаблона сложного фрагмента текста имеет смысл выделить его составные части и описывать их по очередности в виде шаблонов, давая этим шаблонам буквенные имена. Например, грамматически согласованную именную группу, состоящую из прилагательных, существительного (главного слова) и нескольких зависящих от него существительных в родительном падеже, можно задать так:

```
NNG = {A} N1 {N2<c=gen>} <A=N1> (N1)
```

Имя шаблона NNG записывается слева и отделяется от самого шаблона знаком равенства. Указание в конце шаблона элемента N1 в скобках означает, что именная группа NNG наследует весь набор морфологических характеристик главного существительного N1, и в дальнейшем его можно использовать в условиях согласования. Помещенные в круглые скобки характеристики мы называем *параметрами шаблона*. В общем случае в качестве параметров шаблона можно указывать несколько отдельных морфологических характеристик входящих в шаблон элементов, например:

```
NGpn = {A}N1 {N2<c=gen>} <A=N1> (N1.p, N1.n)
```

В качестве дополнительных примеров приведем:

1) LSPL-шаблон, описывающий один из наиболее распространенных синтаксических образцов научно-технических терминов (согласованную именную группу из нескольких адъективов и существительного):

```
AN = {AP} N <AP=N> (N)
```

где AP – ранее описанный шаблон адъектива; параметрами шаблона AN устанавливаются грамматические характеристики входящего в него существительного.

2) Шаблон ACT = AN V <AN=V>

использует шаблон AN и его параметры для описания конструкции, состоящей из именной группы и следующего за ней глагола, согласованного в роде и числе, как во фразе *Построенный программный комплекс удовлетворяет…*.

3) Шаблон для выделения однородных членов, которыми могут быть именные группы:

```
SNG = AN1 {“,” AN2}<1> [“и” AN3] <AN1.c=AN2.c=AN3.c> (AN1)
```

с помощью этого шаблона и шаблона ACT фразу *Дама сдавала в багаж диван, чемодан, саквояж, карт ину, корзину, карт онку и маленькую собачонку* можно задать следующим шаблоном

```
ACT “в багаж” SNG<c=acc>
```

4) LSPL-шаблон перечислений вида *Мышка за кошку, кошка за Жучку, Жучка за внучку, внучка за бабку, бабка за дедку, дедка за репку…*

```
SN = N1 “за” N2<c=acc> {“,” N3 “за” N4<c=acc>}<1> <N1.c=N3.c>
```

5) Шаблон типичной для деловой и научно-технической прозы определяющей конструкции, которая состоит из адъективов, согласованных с ними существительных и подчиненных существительных (вместе с адъективами) в родительном падеже; при этом как вспомогательный используется шаблон AN более простой конструкции из согласованных адъективов и существительного):

```
NP = AN1 {AN2<c=gen>} (AN1)
```

6) Шаблон для одной из характерных конструкций определения новых терминов в научно-технических текстах:

```
DT = NP1<c=acc> ["мы"] "назовем" NP2<c=ins> <NP1.n = NP2.n>
```

Этот шаблон описывает, в частности, фразу *Указанную операцию* ***назовем*** *операцией поиска примеров*, предложение *Поддержку динамичности изменения доступного информационного пространства* ***мы назовем*** *динамичностью информационной модели*, а также и другие фразы подобной структуры и лексического состава.

Заключение

Охарактеризованы основные возможности языка LSPL, предназначенного для описания лексико-синтаксических шаблонов языковых конструкций с целью их распознавания в ходе поверхностного синтаксического анализа текста на русском языке. Хотя разработка языка была инициирована решением задачи автоматизации обработки научно-технических текстов, язык допускает спецификацию достаточно сложных конструкций, присутствующих в текстах любого функционального стиля.

Поскольку разработка любого языка предполагает создание реализующего его программного модуля, соответствующий модуль разрабатывается и для описанной версии LSPL. Модуль опирается на уже существующую утилиту морфологического анализа, созданную Г. Сидоровым на основе словаря А.А. Зализняка, и реализует определенную стратегию распознавания в заданном тексте фрагментов, соответствующих заданному лексико-синтаксическому шаблону. В результате последовательного просмотра текста находятся все возможные фрагменты, соответствующие заданному шаблону (причем эти фрагменты могут пересекаться). Заметим, что для шаблонов общего вида (не конкретизирующих лексемы и их грамматические характеристики) с учетом морфологической омонимии число найденных подходящих фрагментов может быть весьма велико.

Нами изучаются также следующие возможности развития языка LSPL:

- усиление его выразительности за счет введения средств логического комбинирования условий, а также записи условий грамматического управления (пока условия управления можно учесть только для конкретных лексем);
- введение операций над выделенными фрагментами; наиболее значимыми являются операции подсчета статистики, извлечения составных конструкций и синтеза нового шаблона (последнее может применяться, например, для распознавания в научных текстах определений новых терминов и их извлечения).

Литература

1. Большакова Е.И., Васильева Н.Э., Морозов С.С. Лексико-синтаксические шаблоны для автоматического анализа научно-технических текстов // Десятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2006. Труды конференции в 3-х томах. М.: Физматлит, 2006. Т. 2. С.506-524.
2. Ермаков А.Е., Плешко В.В., Митюнин В.А. RCO Pattern Extractor: компонент выделения особых объектов в тексте // Информатизация и информационная безопасность правоохранительных органов: XI Международная научная конференция. Сборник трудов. М.: 2003.
3. Жигалов В.А., Жигалов Д.В., Жуков А.А., Кононенко И.С., Соколова Е.Г., Толдова С.Ю. Система Alex как средство для многоцелевой автоматизированной обработки текстов // Труды международного семинара Диалог’2002 "Компьютерная лингвистика и интеллектуальные технологии". М.: Наука, 2002. Т.2. С.192-208.
4. *Национальный Корпус Русского Языка.* http://www.ruscorpora.ru
5. Романенко С.А. Метаалгоритмический язык Рефал и тенденции его развития // Искусственный интеллект: В 3-х кн. Кн. 3. Программные и аппаратные средства: Справочник. М.: Радио и связь, 1990. С.48-56.
6. Хорошевский В.Ф. OntosMiner: семейство систем извлечения информации из мультиязычных коллекций документов // Девятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2004. Труды конференции в 3-х томах. М.: Физматлит, 2004. Т.2. С.573-581.
7. General Architecture for Text Engineering. http://www.gate.ac.uk/
8. Hearst M.A. Automated Discovery of WordNet Relations. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database. Cambridge.: MIT Press, 1998. P.131-151.