УДК 519.688

СИСТЕМА ДЛЯ ПОИСКА И ВЫДЕЛЕНИЯ КОНСТРУКЦИЙ В ТЕКСТАХ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Е.И. Большакова (*bolsh@cs.msu.ru*)
А.А Носков (*alexey.noskov@gmail.com*)
МГУ им. М.В. Ломоносова, факультет ВМиК

Описывается система, разработанная для автоматического поиска и выделения в тексте на русском языке конструкций по их описанию в виде лексико-синтаксических шаблонов языка LSPL. Рассматривается применение системы для решения трех различных прикладных задач, требующих анализа ЕЯ-текстов.

Введение

Решение многих прикладных задач автоматической обработки текста на естественном языке, требующих проведения интеллектуальных операций над текстом (извлечение знаний [Хорошевский, 2004], реферирование и аннотирование, литературно-научное редактирование) предполагает распознавание в текстах определенных языковых конструкций. К программным системам, автоматизирующим выделение языковых конструкций в текстах относятся известная система GATE [Bontcheva et al., 2003] и подобные ей — Ellogon [Petasis et al., 2002] и RCO Pattern Extractor [Ермаков и др., 2003]. Они достаточно универсальны и предлагают формальные языки (в системе GATE – язык Jape) для задания информации о составе и грамматических свойствах выделяемых конструкций. Эти языки служат для аннотирования фрагментов анализируемого текста и описания преобразований над аннотациями, но в них нет средств явного задания лингвистических свойств, в том числе – грамматического согласования, типичного для многих конструкций русского языка (в первую очередь – именных словосочетаний).

В качестве способа формальной записи языковых конструкций для их представления в системе автоматической обработки текстов на русском языке был предложен язык LSPL [Большакова и др., 2007]. В отличие от языков преобразования аннотаций, он создавался как декларативный язык спецификации лингвистических свойств выделяемых в текстах конструкций с учетом особенностей русского языка. Язык позволяет описывать конструкции в виде *лексико-синтаксических шаблонов*,

определяющих входящие в конструкцию слова с учетом их морфологических характеристик и условий грамматического согласования. Шаблоны конструкций использовались в ряде работ по автоматической обработке текстов, например, лексические шаблоны — в отечественной системе Alex [Жигалов и др., 2002]. Но в отличии от последних, в шаблонах языка LSPL задается грамматическая информация.

Для языка LSPL был разработан метод автоматического выделения в тексте конструкций по их шаблонам [Носков, 2009] и построена программная система, включающая среду для просмотра и анализа текстов. При выделении конструкций заданные шаблоны последовательно накладываются на текст, образуя так называемые *варианты наложения*, соответствующие различным случаям вхождения в текст этих конструкций.

В данной работе кратко характеризуются выразительные средства языка LSPL и разработанный для него метод выделения конструкций по их шаблонам. Описывается построенная программная система поиска и выделения конструкций и рассматривается применение системы для решения трех различных прикладных задач, требующих анализа ЕЯ-текстов: терминологический анализ научно-технического текста, обработка запросов пользователя в вопросно-ответной системе, генерация программных тестов по комментариям программного кода.

1. Язык лексико-синтаксических шаблонов

При создании языка LSPL в него включались выразительные средства, позволяющие достаточно гибко записывать лексикографические единицы (строки, словоформы, лексемы) и их грамматические характеристики с учетом того, что они должны быть понятны не только лингвистам, но и другим специалистам, участвующим в разработке шаблонов.

Шаблон языка LSPL определяет последовательность элементов, из которых состоит описываемая языковая конструкция – слов с их полностью или частично конкретизированными морфологическими характеристиками (часть речи, падеж, род, число и т.п.), и позволяет задавать условия их грамматического согласования. Например, шаблон `N V<t=past> Av<N=V>` описывает конструкцию, состоящую из существительного (`N`), следующего за ним глагола (`V`) в прошедшем времени (`t=past`) и наречия (`Av`), причем существительное и глагол грамматически согласованы; например: *операторы работали быстро*.

В общем случае для входящего в шаблон элемента-слова могут быть указаны часть речи, конкретная лексема, значения морфологических характеристик. Например, шаблон `A<верный,n=plur>` задает прилагательное (`A`) *верный* в любой из возможных форм множественного

числа: *верные*, *верных, верным* и т.п. Условия согласования задают равенство морфологических признаков некоторых элементов-слов.

Для описания конкретных строк, встречающихся в описываемых конструкциях, например, знаков пунктуации, в шаблоне может быть использована запись вида `“;”`.

В шаблон могут включаться и более сложные элементы – повторения и опциональные элементы (записываются соответственно в фигурных и квадратных скобках). К примеру, запись `{N<c=gen>}` обозначает цепочку из нескольких существительных в родительном падеже (*вывод записи файла* и т.п.), а элемент шаблона `["не"]` указывает необязательность вхождения частицы *не* в описываемую конструкцию.

Лексико-синтаксический шаблон может иметь имя и параметры, которые записываются в круглых скобках и фиксируют морфологические характеристики описываемой конструкции. Например, шаблон

```
NP = {A} N1 <A=N1> [N2<c=gen>] (N1)
```

определяет именную группу `NP` из нескольких прилагательных, согласованного с ними существительного и опционального существительного в родительном падеже (*короткий интервал времени, известная теорема, удаленный банковский терминал*). В этот шаблон входят два разных существительных (`N1` и `N2`) и для их записи используются числовые индексы. Параметрами шаблона являются морфологические характеристики первого существительного.

Язык позволяет применять уже определенные шаблоны для задания шаблонов более сложных языковых конструкций. Важно, что при этом можно использовать параметры известных шаблонов для конкретизации элементов последних. К примеру, указанный выше шаблон именной группы можно применить для описания конструкции, включающей эту группу и согласованный с ней глагол в прошедшем времени: `NP V<t=past> <NP=V>` (*внутренний файл проверялся*).

В целом, язык LSPL является достаточно гибким и мощным средством задания лексических и грамматических свойств выделяемых в тексте языковых конструкций, которое позволяет описывать лингвистические свойства сложных конструкций в виде нескольких взаимосвязанных шаблонов.

2. Метод выделения языковых конструкций по шаблонам

Метод выделения основан на представлении текста в виде графа, где ребра соответствуют различным синтаксическим интерпретациям фрагментов текста, состоящих из слов, а вершины соответствуют фрагментам, не содержащим слов (в частности, пробельным символам).

При построении внутреннего представления текста сначала осуществляется его разбиение на фрагменты — слова, знаки препинания и

пробельные символы, и выполняется морфологический анализ слов, морфологические интерпретации которых образуют ребра графа. При этом в общем случае пара соседних вершин графа соединена несколькими ребрами вследствие морфологической омонимии (например, именительного и винительного падежа существительных).

Любая выявленная в дальнейшем конструкция представляется в графе дополнительным ребром. Поскольку LSPL-шаблоны могут иметь параметры, в качестве которых выступают морфологические характеристики входящих в них элементов, то и дополнительным ребрам приписываются значения этих параметров.

Такое графовое представление текста удобно для учета различных сочетаний морфологических интерпретаций входящих в текст слов и позволяет при наложении шаблонов единообразно обрабатывать как элементы-слова, так и вспомогательные шаблоны.

Выделение конструкции представляет из себя поиск в графе текста и включает три основных этапа:

1. Определяется множество ребер, с которых может быть начат поиск в графе. Для этого применяются индексы: слов, шаблонов и частей речи.
2. В графе ищутся пути, начинающиеся с найденных ребер и соответствующие последовательности элементов шаблона. Поиск этих путей представляет из себя обход графа в глубину с откатом назад, при этом на каждом шаге рассматриваются все допустимые продолжения пути, соответствующие текущему элементу шаблона и установленным для него синтаксическим ограничениям (конкретизация падежа, рода и т.п.). На этом же этапе осуществляется проверка условий согласования входящих в шаблон элементов.
3. Найденные пути группируются и образуют варианты наложения, добавляемые в граф в виде новых ребер.

Из-за морфологической омонимии в общем случае возможно несколько вариантов наложения шаблона на один и тот же отрезок текста (несколько путей в графе). Указанная многовариантность создает сложности как для эффективной работы метода, так и для последующего анализа результатов. Для сокращения многовариантности в ходе выделения конструкций по их шаблону предполагается, что важны только те характеристики элементов, которые вошли в число параметров этого шаблона. Варианты наложения, имеющие одинаковые значения параметров, группируются, что позволяет ограничить рост количества наложений и значительно увеличить эффективность анализа.

3. Функции и структура программной системы

Основная функция системы — для заданного текста на естественном языке и набора LSPL-шаблонов поиск в тексте фрагментов, содержащих

варианты их наложения. Найденные варианты наложения представляют вхождения описанных языковых конструкций в текст.

Поскольку языковые конструкции, выделенные при анализе текста, часто требуют некоторой дальнейшей обработки при решении прикладных задач, система предоставляет механизм задания преобразований найденных вариантов наложения. Такие преобразования могут включать нормализацию слов выделенной конструкции, генерацию новых шаблонов из элементов конструкции, перевод выделенных конструкций в структуру данных, требуемую прикладной задачей.

При анализе текстов часто нужны какие-либо внешние словари (например, словарь синонимов). Поэтому, в систему была добавлена возможность подключения словарей. Словари рассматриваются в форме отображения из множества всех словосочетаний в {True,False}, то есть в форме характеристической функции этого множества. Этот подход удобен с той точки зрения, что позволяет использовать "неявные" словари, например, словарь всех слов, встречающихся в индексах Google.

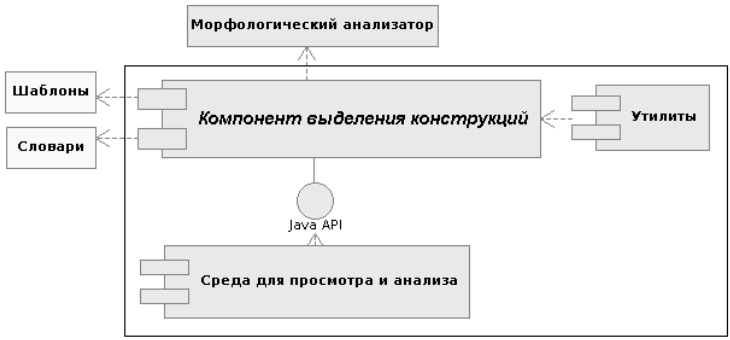

Рис. 1. Архитектура системы

В настоящий момент система включает следующие программные компоненты (см. Рис. 1):

- Ядро, реализующее выделение языковых конструкций по LSPL-шаблонам и их преобразование;
- Консольные утилиты, предоставляющие доступ к функциям ядра;
- Адаптер для использования ядра в языке программирования Java;
- Среда просмотра и анализа текстов, позволяющая пользователю выполнять поиск в тексте нужных конструкций по заданным шаблонам.

Среда реализована на языке программирования Java с использованием адаптера. Для осуществления графематического и морфологического

анализа в программном комплексе предусмотрены подключаемые модули (в настоящий момент используются анализаторы АОТ [Сокирко, 2004]).

4. Среда просмотра и анализа текстов

Входящая в состав системы среда просмотра и анализа текстов, позволяет задавать шаблоны конструкций и визуализировать результаты поиска по заданным шаблонам. Основными пользователями среды являются специалисты, участвующие в разработке LSPL-шаблонов, которым необходимо анализировать тексты и тестировать создаваемые шаблоны. Исходя из этого, основные возможности среды включают:

- Загрузку текстов различных форматов; поддерживаются форматы Microsoft Office, XML, HTML, а также простой неразмеченный текст;
- Определение новых шаблонов или загрузка уже существующих из файла, при этом происходит проверка корректности шаблонов и вывод сообщений о найденных синтаксических ошибках;
- Поиск и выделение в загруженном тексте конструкций по заданным шаблонам; подсчет статистики употреблений конструкций;
- Просмотр морфологических характеристик выделенных конструкций;
- Просмотр вспомогательной информации о представлении текста в виде графа.

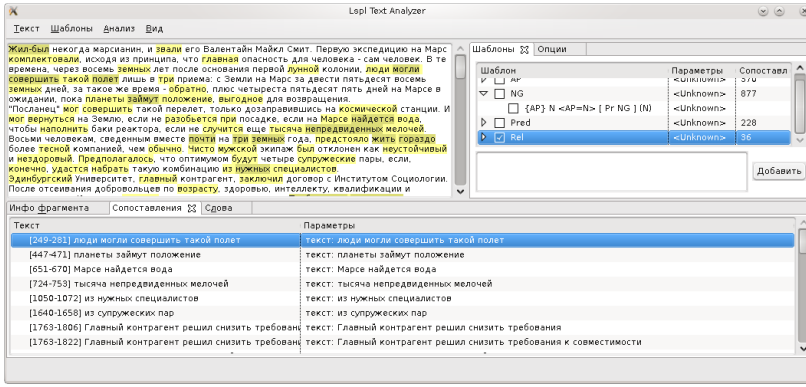

Рис. 2. Среда просмотра и анализа текстов

Пользовательский интерфейс среды состоит из трех основных областей — см. Рис. 2. Левая верхняя область предназначена для редактирования текста и визуализации выделенных конструкций. В правой верхней области добавляются новые шаблоны, и происходит управление визуализацией выделенных конструкций в тексте. В нижней области выводится информация о найденных вариантах наложения выбранного

шаблона, информация о всех конструкциях в тексте под курсором, а также список всех слов текста с фильтрацией по частям речи.

5. Применения программной системы

Описанная система была опробована при решении нескольких различных задач, связанных с обработкой естественного языка: терминологического анализа текстов, генерации кода тестов для программного кода на основе ЕЯ-комментариев и построения вопросно-ответной системы.

Терминологический анализ

Рассматривался как один из этапов в рамках работ по автоматизации обработки научно-технических текстов. Язык LSPL использовался для формального описания [Ефремова и др., 2010]:

- синтаксических образцов научно-технических терминологических словосочетаний (разных видов именных групп *технология накачки* и т.п.);
- регулярно используемых в научно-технических текстах фраз-определений новых (авторских) терминов и их синонимов (*эту операцию будем называть правилом генерализации…; под прерыванием понимается…* и т.п.);
- правила образования лексико-синтаксических вариантов терминов (*библиотека стандартных программ – библиотека программ, шина адреса – адресная шина* и т.п.);
- правила образования в тексте соединений нескольких терминологических словосочетаний (*разрядность регистра, внутренний регистр – разрядность внутреннего регистра* и т.п.).

Для каждой группы полученных LSPL-шаблонов на базе программных компонентов системы были построены и экспериментально исследованы процедуры выявления термиоупотреблений в тексте, что позволило сформулировать стратегию их совместного применения, позволяющую улучшить показатели точности и полноты распознавания терминов в тесте.

Генерация программных тестов на основе комментариев

Основана на том предположении, что написанные на естественном языке комментарии к программному коду часто содержат полуформальное описание требуемого поведения соответствующих объектов кода (функций, методов, классов и т. п.). Это описание может быть использовано для построения процедур автоматизированного тестирования.

На базе описанной программной системы было разработано приложение (в виде плагина для IDE Eclipse), выполняющее генерацию Junit-тестов (процедур тестирования) по Javadoc-комментариям.

Генерация процедур тестирования проходит в три этапа:

1. С помощью шаблонов именных групп извлекаются возможные именования элементов программного кода. Семантика именований определяется из разметки комментариев для системы автоматической генерации документации Javadoc.

2. Для каждого именования элемента кода формируется набор LSPL-шаблонов для выделения употреблений этого именования в тексте (при этом используется предоставляемый системой механизм преобразований выделенных конструкций). На основе сформированных шаблонов именований строятся шаблоны конструкций, описывающих различные аспекты поведения кода (взаимосвязь параметров функции и ее результата, условия возникновения исключений и т.п.).

3. Построенные шаблоны используются для выделения из текста комментариев набора ограничений, описывающих поведение кода, а из них — набора проверяемых в тестах условий, по которым генерируется код процедур тестирования.

Вопросно-ответная система

Является еще одним примером приложения, разработанного с использованием LSPL-шаблонов и программных средств их поддержки. Вопросно-ответная система анализирует утверждения и вопросы, вводимые пользователем в виде предложений русского языка и переводит их в формулы логики первого порядка. Для синтаксического анализа предложений используются LSPL-шаблоны, а для перевода в формулы применяется механизм преобразований. Отметим, что использование этого механизма позволяет описывать правила преобразований непосредственно в шаблонах и отделить их от программного кода вопросно-ответной системы.

Формулы, полученные из ЕЯ-утверждений, рассматриваются как аксиомы и образуют базу знаний системы, а полученные из вопросов формулы — как подлежащие доказательству теоремы. В результате доказательства система может выдать положительный ответ (теорема доказана), отрицательный (доказана обратная теорема) или же сообщить, что ответ неизвестен (ни одна из двух теорем не может быть доказана). Для доказательства применяется метод резолюций.

Заключение

В работе описана программная система для поиска и выделения конструкций естественного языка по их формальному описанию в виде

лексико-синтаксических шаблонов языка LSPL. Успешное применение системы для трех различных прикладных задач показывает, что на основе системы могут быть решены многие задачи, требующие извлечения и анализа информации из текстов на естественном языке.

Список литературы

[Bontcheva et al., 2003] Bontcheva K., Maynard D., Tablan V., and Cunningham H. GATE: A Unicode-based infrastructure supporting multilingual information extraction. In proceedings of Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL'03), Borovets, 2003.

[Petasis et al., 2002] Petasis G., Karkaletsis V., Paliouras G., Androutsopoulos I. and Spyropoulos C. D. Ellogon: A New Text Engineering Platform // Proc. of the Third International Conference on Language Resources and Evaluation (LREC 2002). Las Palmas, 2002, p. 72-78.

[Большакова и др., 2007] Большакова Е.И., Баева Н.В., Бордаченкова Е.А., Васильева Н.Э., Морозов С.С. Лексико-синтаксические шаблоны в задачах автоматической обработки текстов // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции Диалог '2007. – М.: Издательский центр РГГУ, 2007, с.70-75.

[Ермаков и др., 2003] 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 – Москва, 2003.

[Ефремова и др., 2010] Ефремова Н.Э., Большакова Е.И., Носков А.А., Антонов В.Ю. Терминологический анализ текста на основе лексико-синтаксических шаблонов // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Межд. конференции «Диалог» (Бекасово, 26-30 мая 2010 г.) Вып. 9(16). – М.: Изд-во РГГУ, 2010, с. 124-129.

[Жигалов и др., 2002] 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.

[Носков, 2009] Носков А.А. Метод выделения в тексте конструкций по их лексико-синтаксическим шаблонам // Сборник статей молодых ученых факультета ВМиК МГУ- М.: МАКС Пресс, 2009, Выпуск 6, с. 136-145.

[Сокирко, 2004] Сокирко А. В. Морфологические модули на сайте www.aot.ru // Труды международной конференции «Диалог-2004. Компьютерная лингвистика и интеллектуальные технологии». М.: Наука, 2004. С. 559.

[Хорошевский, 2004] Хорошевский В. Ф., Управление знаниями и обработка ЕЯ-текстов // Девятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2004: Труды конференции. В 3-х т. М.: Физматлит, 2004, т. 2, стр. 565-572.