Анализ текста на основе лексико-синтаксичеких шаблонов с сокращением многовариантности

Е.И. Большакова, А.А. Носков
Факультет ВМиК МГУ им. М.В.Ломоносова
Email: bolsh@cs.msu.su , alexey.noskov@gmail.com

Введение

В последнее время разрабатывается все больше различных приложений по автоматической обработке текста на естественном языке, решающих задачи извлечения определенной информации из текста – определений и связей терминов, именованных сущностей (имен персоналий, дат, названий фирм и т.п.) [1]. Извлечение информации из текста предполагает распознавание в текстах нужных языковых конструкций, что требует создания необходимых для этого программных средств. Как правило, распознавание выполняется на основе частичного синтаксического анализа текста, с использованием информации о составе и грамматических свойствах выделяемых конструкций, что отличает эту задачу от классической задачи синтаксического анализа, при которой последовательно по предложениям выполняется полный синтаксический разбор текста.

К программным системам, позволяющим автоматизировать выделение нужных языковых конструкций в текстах на естественном языке, относится известная система GATE [2] и подобные ей, например, Ellogon [3]. Эти системы достаточно универсальны и предлагают специальные языки (в системе GATE – язык Jape) для аннотирования фрагментов анализируемого текста и описания преобразований над аннотациями. Однако их использование требует определенной квалификации и опыта и усложняется тем, что в них нет средств описания специфичных лингвистических свойств, например, грамматического согласования, существенного для многих конструкций русского языка, в частности – именных словосочетаний.

В качестве средства задания языковых конструкций для их автоматического выделения, учитывающего особенности русского языка и имеющего существенно более низкий порог вхождения, был предложен язык LSPL [4]. В отличие от языков преобразования аннотаций, он создавался как декларативный язык спецификации выделяемых в текстах конструкций. Язык позволяет описывать конструкции в виде *лексико-синтаксических шаблонов*, определяющих входящие в конструкцию слова с учетом их морфологических характеристик и задающих условия их грамматического согласования.

Для языка LSPL был разработан метод автоматического выделения в тексте конструкций по их описанию в виде лексико-синтаксических шаблонов [5]. При выделении конструкции заданный шаблон последовательно накладывается на текст, образуя так называемые *варианты наложения*, соответствующие различным случаям вхождения в текст этой конструкции.

В настоящей работе кратко характеризуется язык шаблонов LSPL, описываются ключевые моменты разработанного для него метода выделения конструкций в тексте, а также рассматривается применяемая при выделении конструкций *группировка вариантов* наложения – способ сокращения множества возникающих при анализе текста синтаксических интерпретаций. Рассматриваемый способ представляет некоторое частное решение проблемы неоднозначности (многовариантности)

синтаксического разбора, неизбежно возникающей как при полном, так и при частичном синтаксическом анализе текста на естественном языке и являющейся одной из главных и не решенных в полной мере его проблем.

Лексико-синтаксические шаблоны

Шаблон языка LSPL описывает некоторую языковую конструкцию путем указания входящих в нее элементов – слов с их полностью или частично конкретизированными морфологическими характеристиками (часть речи, падеж, род, число и т.п.). Порядок следования элементов в шаблоне соответствует порядку элементов описываемой языковой конструкции. Например, шаблон `N V<t=past> Av` описывает конструкцию, состоящую из существительного (`N`), следующего за ним глагола (`V`) в прошедшем времени (`t=past`) и наречия (`Av`), например: *дети бежали быстро*. Для входящего в шаблон *элемента-слова* могут быть указаны часть речи, конкретная лексема, значения морфологических характеристик. К примеру, шаблон `A<синий,n=sing>` задает прилагательное (`A`) *синий* в любой из возможных форм единственного числа (`n=sing`): *синяя*, *синим, синему* и т.п.

Язык LSPL позволяет задавать *условия грамматического согласования*, указывающие равенство морфологических признаков некоторых элементов-слов описываемой конструкции. Например, шаблон `A N <A.g=N.g, A.n=N.n>` описывает прилагательное со следующим за ним существительным, согласованные по роду (`g`) и числу (`n`): *белое платье*, *синим туманом*, *горячего снега*. При необходимости можно задавать в шаблоне согласование слов по всем их общим морфологическим характеристикам, к примеру: `A N <A=N>`.

Для описания конкретных строк, встречающихся в конструкции, в шаблоне может быть использована запись вида `"строка"`. В частности, такой *элемент-строка* может быть использован для задания в шаблоне знаков пунктуации, например: `";"`.

Кроме элементов-слов и элементов-строк шаблоны могут содержать и более сложные элементы, к которым относятся *повторения* (записываются в фигурных скобках). К примеру, запись `{N<c=gen>}<1,5>` обозначает цепочку от одного до пяти существительных в родительном падеже (`c=gen`). Частным случаем повторения является *опциональный элемент* (записывается в квадратных скобках). К примеру, шаблон `[Pa] N <Pa =N>` задает в общем случае причастие (`Pa`) и согласованное с ним (по всем общим морфологическим характеристикам) существительное, но причастие может быть опущено.

Язык позволяет давать имена шаблонам и использовать уже определенные шаблоны для задания шаблонов более сложных конструкций. Например, шаблон с именем `NP`

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

определяет именную группу из нескольких прилагательных, согласованного с ними существительного и опционального существительного в родительном падеже (*белые шапки гор, теплый летний дождь, синий туман*). Поскольку в этот шаблон входят в общем случае два разных существительных (`N1` и `N2`), в их записи используются числовые индексы. Указанный шаблон можно использовать для описания конструкции, состоящей из описанной именной группы и глагола в прошедшем времени: `NP V<t=past>` (*пушистый кот спал*). Таким образом, в

качестве элементов шаблона язык допускает *экземпляры* (употребления) других (вспомогательных) шаблонов, а для описания выделяемой в тексте конструкции часто используется несколько взаимосвязанных шаблонов.

LSPL-шаблон может иметь *параметры*, задающие морфологические характеристики описываемой шаблоном конструкции (указываются в конце шаблона, в скобках), например, `N V (N.c,N.g)` – шаблон, параметрами которого являются падеж (`c`) и род (`g`) входящего в шаблон существительного. Для сокращения может быть использована запись вида `N V (N)` – в таком шаблоне параметрами являются все морфологические характеристики существительного.

Параметры шаблона особенно ценны при использовании шаблонов в других шаблонах. В рассмотренном выше шаблоне `NP` в качестве параметров могли быть установлены морфологические характеристики первого существительного `N1`:

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

Тогда допустимо использовать их в условиях согласования: шаблон `NP V <NP=V>` задает конструкцию из именной группы `NG` и согласованного с ней глагола `V` (примером является словосочетание *белый кот спал*, но не *белый кот спала*, поскольку последнее не согласовано.

Язык LSPL позволяет также записывать в шаблоне несколько альтернатив, соответствующих различным вариантам описываемой языковой конструкции. Например, шаблон `AP = A|Pa` описывает адъектив – прилагательное или причастие.

В целом, язык шаблонов является достаточно гибким и мощным средством задания лексических и грамматических свойств выделяемых в тексте конструкций.

Внутреннее представление текста и вариантов наложения

При наложении LSPL-шаблона на текст, т.е. при поиске и выделении в тексте языковой конструкции, свойства которой заданы этим шаблоном, получаются *варианты наложения*. Каждый вариант наложения – это отрезок текста (непрерывная последовательность символов текста), соответствующий выделенной конструкции, вместе с набором конкретных значений морфологических характеристик слов, входящих в этот отрезок. Будем называть такой набор *синтаксической интерпретацией* отрезка текста. В случае, когда отрезок состоит из одного слова, точнее, словоформы, синтаксической интерпретацией является набор ее морфологических характеристик.

Отрезок текста, представляющий вариант наложения шаблона, включает подотрезки, соответствующие разным элементам этого шаблона, например, словам или повторениям. Кроме них, в отрезке есть символы, *незначимые* с точки зрения производимого анализа. Незначимыми обычно считаются пробельные и управляющие символы, но к ним могут быть отнесены также знаки пунктуации и другие знаки. В общем, любой анализируемый с помощью шаблонов текст разбивается на непересекающиеся отрезки: *значимые* и *незначимые*. Это разбиение используется для построения внутреннего представления текста в виде графа, предложенного для эффективного наложения шаблонов.

Вершины графа текста соответствуют незначимым отрезкам текста, а ребрами являются различные синтаксические интерпретации лежащих между ними значимых отрезков текста.

При построении внутреннего представления текста сначала осуществляется его разбиение на значимые и незначимые отрезки, при этом идущие подряд незначимые отрезки склеиваются и образуют вершины графа. Построенные вершины нумеруются, начиная с 0, в направлении от начала к концу текста. Затем выполняется морфологический анализ слов, входящих в значимые отрезки, и построение ребер графа между соседними вершинами – эти ребра соответствуют установленным в ходе анализа морфологическим интерпретациям этих слов. Граф считается ориентированным: все его ребра направлены в сторону вершин с большим номером.

В общем случае между парой соседних вершин графа проходит несколько ребер — они соответствуют различным синтаксическим (морфологическим) интерпретациям слов. На Рис. 1 приведен пример графа текста, и над ним показано разбиение текста на значимые и незначимые отрезки (светлые и темные участки соответственно).

Как видно из рисунка, количество синтаксических интерпретаций может оказаться достаточно велико. Например, у слова *маленький* их две (одна соответствует именительному падежу, другая – винительному), а у слова *высокого* – три (мужской род, именительный или винительный падеж, родительный падеж среднего рода).

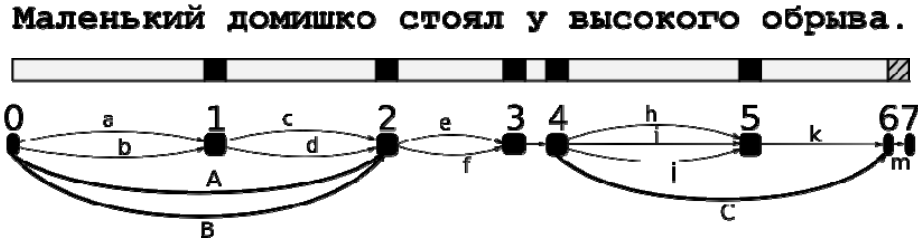

Рис. 1. Граф текста с вариантами наложения шаблона A N <A=N> (N)

Любая пара вершин в этом графе внутреннего представления однозначно определяет некоторый отрезок текста, фиксируя его начало и конец, а различные пути в графе между этими вершинами задают возможные комбинации синтаксических интерпретаций входящих в этот отрезок значимых подотрезков. Таким образом, построение графа позволяет рассматривать задачу выделения конструкций в тексте, как поиск путей в графе, удовлетворяющих синтаксическим требованиям, накладываемым шаблоном.

В графе текста сохраняются также промежуточные результаты его анализа по шаблонам: любая выявленная в тексте в результате наложения некоторого шаблона конструкция представляется в графе дополнительным ребром, соединяющими вершины – конечные точки соответствующего отрезка текста. Поскольку применяемые шаблоны могут иметь параметры, в качестве которых выступают морфологические характеристики входящих в них элементов, то и дополнительным ребрам приписываются значения этих параметров.

В общем случае одному и тому же отрезку может соответствовать несколько вариантов наложения, отличающихся только значениями морфологических характеристик. В графе на Рис. 1 жирно выделены два ребра A и B между вершинами 0 и 2, которые отображают два варианта наложения на отрезок текста *маленький домишко* шаблона A N <A=N> (N), описывающего согласованную пару из прилагательного и существительного. Первый вариант соответствует именительному, а второй – винительному падежу слов шаблона. Другое жирное ребро C между

вершинами 4 и 6 соответствует наложению этого же шаблона на отрезок *высокого обрыва*.

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

Стратегия наложения шаблона на текст

Наложение шаблона на текст рассматривается как поиск в графе текста и включает три основных этапа:

1. Определение множества ребер, с которых может быть начат поиск в графе;
2. Поиск путей в графе, соответствующих шаблону, начиная с найденных ребер;
3. Группировка вариантов наложения – найденные пути группируются и образуют варианты наложения, добавляемые в граф в виде новых ребер.

Цель первого этапа – максимально сузить множество ребер, с которых могут начинаться языковые конструкции, описанные шаблоном. Для этого применяются *индексы*, позволяющие быстро определять множество ребер графа, с которых есть смысл начинать наложение шаблона. Используются три типа индексов: индекс слов текста, индекс частей речи и индекс наложенных шаблонов. Индексы строятся одновременно с построением графа текста и обеспечивают доступ соответственно к словоформам заданного слова, словам нужной части речи и вариантам наложения заданного шаблона.

На втором этапе рассматривается множество всех путей в графе, начинающихся с заданного ребра. В этом множестве ищутся те пути, которые соответствуют последовательности элементов шаблона. Поиск нужных путей представляет из себя обход графа в глубину с откатом назад, при этом на каждом шаге рассматриваются все допустимые продолжения пути, которые соответствуют текущему элементу шаблона (например, слову определенной части речи или варианту наложения заданного шаблона) и установленным для него синтаксическим ограничениям (конкретизация падежа, рода и других морфологических характеристик).

Важный момент второго этапа – проверка условий согласования входящих в шаблон элементов. Для оптимизации поиска она выполняется сразу, как только становятся конкретизированы все морфологические характеристики, входящие в условие согласования, что обеспечивает более раннее отсечение несогласованных вариантов наложения.

Чтобы реализовать это, во-первых, при переводе шаблона во внутреннее представление условия согласования сдвигаются максимально влево, до позиции последнего элемента, участвующего в согласовании. Во-вторых, в процессе поиска в графе поддерживается так называемый *контекст наложения*, отражающий состояние процесса наложения шаблона. Как только какой-либо элемент шаблона ставится в соответствие ребру графа, в контекст наложения добавляется пара «элемент — ребро». Впоследствии, если встречается условие согласования, то ребра, соответствующие участвующим в согласовании элементам, извлекаются из контекста и их характеристики сравниваются.

Другой немаловажный момент поиска – обработка сложных элементов шаблона: повторений и экземпляров шаблонов. Таким элементам могут быть поставлены в соответствие последовательности из нескольких ребер, в отличие от простых элементов, которые накладываются на одно ребро. Для того, чтобы не нарушать

единого принципа представления информации о наложениях (каждому элементу шаблона ставится в соответствие одно ребро), наложение любого сложного элемента включает добавление специального группирующего ребра над всей соответствующей этому элементу последовательностью ребер.

На третьем, заключительном этапе наложения шаблона на текст найденные пути в графе (варианты наложения шаблона) должны быть внесены в граф текста, однако перед этим происходит их группировка, существенно сокращающая их число.

Группировка вариантов наложения

В общем случае каждый значимый отрезок в тексте имеет несколько синтаксических интерпретаций, и, следовательно, представляется несколькими кратными ребрами в графе. Поскольку при наложении шаблона выполняется поиск всех допустимых путей в графе, то наличие в графе кратных ребер означает существование различных путей между двумя вершинами, и, значит, возможность нескольких вариантов наложения шаблона на один и тот же отрезок текста. Эти варианты возникают вследствие морфологической омонимии слов текста. Количество получающихся вариантов для шаблонов общего вида (не конкретизирующих лексемы и их грамматические характеристики) может быть весьма велико. Часть этих вариантов отсеивается при проверке условий согласования (если таковые записаны в шаблоне).

Рассмотрим, к примеру, анализ текста *большой зал внезапно наполнился мягким светом* с помощью шаблона

```
S = NG1 Av V<t=past> NG2
```

и использованного в нем вспомогательного шаблона именной группы `NG = A N`. Поскольку в шаблоне `NG` нет условий согласования, а количество морфологических интерпретаций словоформ *большой* и *зал* равно соответственно шести (мужской род, именительный или винительный падеж; женский род, родительный, дательный, творительный или предложный падеж) и трем (мужской род, единственное число, именительный или винительный падеж; женский род, множественное число, творительный падеж), то получается 18 вариантов наложения на отрезок *большой зал*. С учетом трех вариантов наложения этого шаблона на отрезок *мягким светом*, получаем 18×3 вариантов наложения шаблона `S` на всё рассматриваемое предложение. Введение же условий согласования в шаблон именной группы `NG`: `NG = A N <A=N>` позволяет сократить количество вариантов до 4 (при этом вполне вероятно, что не все они действительно необходимы).

Неконтролируемый рост числа вариантов наложения может возникать в случае использования в шаблоне повторения элементов, если при этом не используются условия согласования. Рассмотрим, к примеру, шаблон `P = A {N} '.'`, который выделяет прилагательное и следующие за ним существительные до первой точки. Если этот шаблон накладывается на отрезок текста из 7 слов, каждое из которых имеет по две морфологические интерпретации, то получается 128 различных вариантов наложения, соответствующих всевозможным комбинациям интерпретаций слов. Заметим, однако, что интерпретации входящих в повторение существительных, по существу, не нужны и их можно сгруппировать и считать неразличимыми.

Ясно, что количество вариантов наложения очень быстро растет как с увеличением сложности шаблона, так и с увеличением длины анализируемого текста, что приводит к увеличению расхода памяти на поддержание графа и значительному

снижению эффективности поиска. Кроме того, указанная многовариантность нежелательна и для человека, изучающего результаты выделения нужной конструкции, поскольку она выглядит как множество практически неотличимых друг от друга вариантов наложений шаблона на один и тот же отрезок текста.

Для сокращения многовариантности в ходе выделения конструкций по их LSPL-шаблонам используется следующий принцип, основанный на использовании информации о составе применяемых шаблонов – их параметрах, условиях, элементах. Предполагается, что при анализе по шаблонам важны только те характеристики элементов, которые будут использованы в объемлющей конструкции или вошли в число параметров самого шаблона. Все синтаксические интерпретации элементов, различающиеся только остальными характеристиками, могут быть сгруппированы и сделаны неразличимыми для дальнейшего поиска на графе.

Группировка согласно указанному принципу применяется в нескольких случаях.

В случае наложения шаблона важны только те морфологические характеристики, которые вошли в параметры шаблона – эти характеристики не могут быть опущены при дальнейшем анализе результатов человеком и при наложении других шаблонов. Например, для шаблона `NG = A N <A=N> (N)` важными считаются только морфологические характеристики существительного (но не прилагательного). Это позволяет осуществить группировку вариантов наложения по различным наборам значений параметров шаблона, т.е. заменить одним вариантом все варианты наложения, различающиеся только характеристиками прилагательного, что приводит к значительному уменьшению количества рассматриваемых в дальнейшем вариантов наложения. Результат проводимого в таких случаях процесса группировки представляет собой группу вариантов наложения, неразличимых с точки зрения выходных морфологических характеристик.

Для приведенного выше примера текста и шаблона `NG` использование такого подхода сокращает количество вариантов наложения для отрезка *большой зал* до двух, соответствующих именительному и винительному падежу слова *зал*, в то же время шаблон `A N <A=N>`, отличающийся от предыдущего только тем, что не имеет параметров, после группировки (по синтаксическим интерпретациям как прилагательного, так и существительного) имеет уже только один вариант наложения на тот же отрезок.

В случае повторения элементов шаблона применяемый нами метод производит группировку всех синтаксических интерпретаций соседних отрезков, на которые наложены повторяющиеся элементы, если участвующие в повторении элементы (или их параметры) не вынесены в параметры шаблона и не участвуют в условиях согласования. Для рассмотренного выше примера применения шаблона `P = A {N} '.'` вместо 128 вариантов наложения после такой группировки останется всего лишь два варианта, соответствующих двум интерпретациям первого слова (прилагательного).

Подчеркнем, что во всех случаях группировка производится только для интерпретаций одного отрезка текста.

Важно, что группировка вариантов наложения не означает потери информации о найденных синтаксических интерпретациях – эта информация сохраняется, но в «запакованном» виде, и при необходимости может быть распакована и использована для более подробного анализа соответствующих отрезков текста. По этой причине

словосочетание «группировка вариантов» более точно отражает суть применяемого нами метода (его третьего этапа), чем сочетание «сокращение вариантов».

Также заметим, что меняя набор выходных параметров шаблона, можно управлять количеством возможных вариантов его наложения на текст.

Заключение

Представленный метод анализа текста позволяет осуществлять выделение в тексте на русском языке различных языковых конструкций, описанных в виде лексико-синтаксических шаблонов языка LSPL. Для сокращения возникающей при анализе текста многовариантности предложен способ группировки вариантов наложения шаблонов.

Разработан программный комплекс, включающий:

- Ядро, написанное на языке C++ и реализующее предложенный метод;
- Набор консольных утилит для интеграции ядра с различными скриптами;
- Прикладной интерфейс для языка Java;
- Графический пользовательский интерфейс для анализа текста лингвистом.

Комплекс был опробован для решения ряда задач терминологического анализа русскоязычного текста и показал неплохие результаты по производительности и устойчивости работы.

В программном комплексе дополнительно реализованы средства подключения дополнительных компонентов, осуществляющих некоторые операции над найденными вариантами наложения шаблона: подсчет статистики выделенных конструкций, трансляцию их в логические формулы, извлечение составных частей конструкций (например, терминологических вариантов из конструкций определений терминов), синтез новых шаблонов на основе множества найденных вариантов наложения.

Список литературы

1. 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 т. М.: Физматлит, 2004, т. 2, с. 573-581.
2. Bontcheva K., et al. Developing Reusable and Robust Language Processing Components for Information Systems using GATE. In: Proceedings of the 13th Int. Workshop on Database and Expert Systems Applications, DEXA. Washington, 2002, p. 223-227.
3. Petasis G., et al. Ellogon: A New Text Engineering Platform. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002). Las Palmas, 2002, p. 72-78.
4. 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.
5. Носков А.А. Метод выделения в тексте конструкций по их лексико-синтаксическим шаблонам // Сборник статей молодых ученых факультета ВМиК МГУ- М.: Издательский отдел фак-та ВМиК МГУ им. М.В. Ломоносова; МАКС Пресс, 2009, Выпуск 6, с. 136-145.