УДК 519.688

МЕТОД ВЫДЕЛЕНИЯ В ТЕКСТЕ КОНСТРУКЦИЙ ПО ИХ ЛЕКСИКО-СИНТАКСИЧЕСКИМ ШАБЛОНАМ

© 2009 г. А. А. Носков

alexey.noskov@gmail.com

Кафедра Алгоритмических Языков

1 Введение

В настоящее время в различных областях компьютерной лингвистики и искусственного интеллекта существует потребность в средствах автоматизации выделения в тексте на естественном языке определенных языковых конструкций, в частности, согласованных именных словосочетаний (*усталое осеннее солнце*, *уходящий поезд*), глагольных групп (*шел по тротуару*, *писать стихи*), а также более сложных конструкций, характерных для текстов конкретного, например, научно-технического стиля (*под A будем понимать B*, *предположим, что C*) и т.п. До сих пор задача такого выделения обычно решалась каждый раз заново в условиях конкретного приложения по автоматической обработке текста, и для конкретных типов языковых конструкций. В данной работе предлагается новый метод, позволяющий осуществлять выделение в тексте достаточно широкого круга языковых конструкций, описанных в виде шаблонов языка LSPL[1].

В отличие от давно изучающейся задачи полного синтаксического анализа, при которой последовательно по предложениям распознается вся синтаксическая структура текста, решаемая нами задача предполагает распознавание в тексте только заданных конструкций по описанию их свойств. Безусловно, она может быть решена и на основе полного синтаксического анализа, однако полный анализ сам по себе является более сложной и высокозатратной задачей, избыточной для целей выделения только необходимых конструкций. Фактически, можно считать задачу выделения в тексте языковых конструкций, как разновидность поверхностного синтаксического анализа.

Если рассматривать возможность использования уже существующих инструментов для решения указанной задачи, то стоит отметить, что она быть решена достаточно просто при наличии подготовленных корпусов текстов, таких как НКРЯ (Национальный корпус русского языка) или ХАНКО (Хельсинский аннотированный корпус) [4] – за счет использования морфологической и синтаксической разметки текстов корпуса, подготовленной вручную лингвистами. Однако корпуса ограничены по набору текстов, в то время как часто возникает необходимость исследовать тексты, не включенные в корпус. Кроме того, в корпусах размечаются только те языковые конструкции, которые были признаны значимыми для задач, решаемых корпусом, в то время как нередко требуется выделять в текстах другие конструкции, для которых разметка в корпусе отсутствует.

К инструментам, позволяющим автоматизировать выделение нужных типов языковых конструкций, следует отнести такие инструментальные системы для построения приложений обработки естественного языка, как GATE (General Architecture for Text Engineering) [5] и Ellogon [6]. Эти системы предлагают языковые средства для описания единиц текста (например, язык Japе в системе GATE) и программные средства для автоматического выделения фрагментов текста по такому описанию. Однако несмотря на универсальность этих систем их применение для русского языка представляет некоторые сложности. Большинство подобных систем разрабатывались для английского языка, который по ряду свойств отличается от русского языка, который является более флективным и имеет менее строгий порядок слов, что приводит к необходимости использовать при анализе русскоязычных текстов широкий набор морфологических характеристик слов. Поскольку указанные системы разрабатывались для английского языка, в них также нет специальных средств задания условия грамматического согласования, крайне важного для автоматического выделения конструкций русского языка.

Если же рассматривать отечественные системы, ориентированные на анализ текстов на русском языке, то среди них стоит отметить систему Alex [3], позволяющую описывать в виде лексических шаблонов слова, словосочетания и их сокращения, и затем автоматически выделять их в тексте. Основным недостатком системы является ограниченность языка шаблонов — в нем нет возможности использования морфологических характеристик слов описываемых языковых конструкций, что значительно сужает возможности системы.

В качестве более гибкого средства задания конструкций естественного языка для их автоматического выделения, был предложен язык LSPL [1], учитывающий особенности русского языка. Он позволяет в удобной форме специфицировать конструкции русского языка из достаточно широкого класса в виде так называемых лексико-синтаксических шаблонов, определяющих входящие в конструкцию слова с учетом их морфологических характеристик и задающих условия их грамматического согласования. При автоматическом выделении некоторой конструкции шаблон последовательно накладывается на текст, образуя так называемые варианты наложения, соответствующие различным случаям вхождения в текст этой конструкции. Рассматриваемый в настоящей работе метод выделения языковых конструкций в тексте разработан именно для случая их описания в виде LSPL-шаблонов.

2 Язык LSPL и задача выделения конструкций

Основным средством языка LSPL является шаблон, описывающий некоторую языковую конструкцию. Каждый шаблон в общем случае состоит из имени и набора альтернатив, описывающих различные варианты языковой конструкции. Например, если шаблон описывает пару слов, соединенных союзом «*или*» или «*либо*», то он содержит две соответствующие альтернативы, разделенные символом «|»: `N1 «или» N2 | N1 «либо» N2`.

Элементами шаблона языка LSPL являются входящие в описываемую языковую конструкцию слова с их полностью или частично конкретизированными морфологическими характеристиками (часть речи, падеж, род, число и т.п.). При этом порядок следования элементов в шаблоне соответствует порядку элементов описываемой языковой конструкции. Например, шаблон `N V<t=past>` описывает конструкцию из существительного (`N`) и следующего за ним глагола (`V`) в прошедшем времени (`t=past`). Элемент шаблона может описывать любую из допустимых словоформ некоторого слова, в этом случае в шаблоне записывается начальная форма этого слова (в частности, для существительных – именительный падеж единственного числа), например: `A<белый>` описывает прилагательное «белый», находящееся в конструкции в любой из возможных форм.

Язык LSPL позволяет задавать условия грамматического согласования, указывающие равенство морфологических признаков у различных 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`).

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

Кроме указанных простых элементов шаблоны могут содержать и более сложные элементы, в частности, повторения, которые записываются в фигурных скобках. Например, запись `A<1,3> N <A=N>` обозначает последовательность, состоящую из одного, двух или трех прилагательных, за которыми следует согласованное с ними (по всем общим морфологическим характеристикам) существительное. Частным случаем повторения является опциональный элемент: к примеру, запись `[A] N <A=N>` описывает в общем случае прилагательное вместе со согласованным с ним существительным, но прилагательное может быть опущено.

Достаточно часто удобно выделить некоторую стандартную конструкцию или ее часть в отдельный шаблон и дать ему имя, что допускается языком LSPL. Например, шаблон с именем `NameGroup`:

```
NameGroup = {A} N1 <A=N1>
```

описывает именную группу, состоящую из последовательности прилагательных и согласованного с ними существительного (например, *белый снег* или *теплый летний дождь*, но не *белый снега*). После такого описания можно использовать этот шаблон в других шаблонах, к примеру шаблон `NameGroup V` задает последовательность из согласованной именной группы и следующего за ней глагола (*пушистый кот спал*, но не *пушистый кошка спала*).

LSPL-шаблон может иметь параметры, задающие морфологические характеристики описываемой шаблоном конструкции. Например, запись `NG = A N (N.c, N.g)` — шаблон с именем `NG`, параметрами которого являются падеж (`c`) и род (`g`) входящего в шаблон существительного. Для сокращения может быть использована запись вида `NG = A N (N)` — в таком шаблоне параметрами являются все морфологические характеристики существительного. Параметры шаблона особенно ценны при использовании шаблонов в других шаблонах. Например, можно описать именную группу как:

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

Такая именная группа задает последовательность прилагательных с согласованным с ними существительным и опциональным существительным в родительном падеже (к примеру, *белые шапки гор*). В дальнейшем для экземпляров такого шаблона можно использовать условие согласования: конструкция `NG1 V <NG1=V>` соответствует именной группе `NG`, согласованной по всем морфологическим характеристикам с глаголом `V` и описывает словосочетания вида *белый кот спал*, но не *белый кот спала* (последнее не согласовано).

Кроме использования уже описанных шаблонов в других шаблонах язык LSPL позволяет строить рекурсивные определения шаблонов:

```
NG = {A} N1 <A=N1> [NG2<c=gen>] (N1)
```

Этот шаблон именной группы отличается от предыдущего тем, что опциональная часть может быть представлено не только одиночным существительным, но и произвольной именной группой. К примеру, этот шаблон описывает такие конструкции, как *тоненькая струйка дыма далекого пожара*.

Таким образом, язык LSPL является достаточно мощным средством для описания различных языковых конструкций. Для заданного текста и LSPL-шаблона задача выделения в тексте на русском языке конструкций по шаблону может быть уточнена так: найти все фрагменты текста, представляющие собой языковую конструкцию, свойства которой описаны в виде LSPL-шаблона. При автоматическом выделении конструкции шаблоны накладываются на текст, образуя так называемые **варианты наложения** – непрерывные последовательности символов текста, или **отрезки текста**, соответствующие выделенной конструкции.

LSPL-шаблон может содержать входящие в описываемую языковую конструкцию знаки пунктуации (в виде **элементов-строк**), однако конструкции могут быть описаны и без учета пунктуации. Очевидно, что в последнем случае при выделении конструкций пунктуацию необходимо просто игнорировать. Для поддержки обоих возможностей в предлагаемом методе вводится два режима выделения, которые в дальнейшем будут обозначаться «с учетом пунктуации» и «без учета пунктуации».

Отрезок текста, представляющий вариант наложения LSPL-шаблона, включает подотрезки, соответствующие простым элементам этого шаблона, например, словам или знакам пунктуации (в случае анализа с учетом пунктуации). Кроме них, в общем случае в отрезке есть символы, незначащие с точки зрения производимого анализа, например, пробельные и управляющие символы, а также знаки пунктуации в случае выделения без учета пунктуации. В целом, рассматриваемый отрезок разбивается на непересекающиеся отрезки: **значимые** и **незначимые** для проводимого анализа, причем такое деление отрезка на непересекающиеся подотрезки зависит только от режима выделения, но не от конкретного шаблона, и его можно распространить на весь текст.

Например, при анализе без учета пунктуации текст "*Человек шел, оглядываясь назад.*"будет содержать два незначимых отрезка, соответствующих пробельным символам, один

незначимый отрезок, включающий запятую и пробел за ней, и один незначимый отрезок, соответствующий точке в конце предложения (см. Рис. 1). Если же пунктуация учитывается, то незначимыми отрезками будут только те, которые состоят из пробельных символов.

Указанное разбиение текста на непересекающиеся отрезки используется для построения внутреннего представления текста.

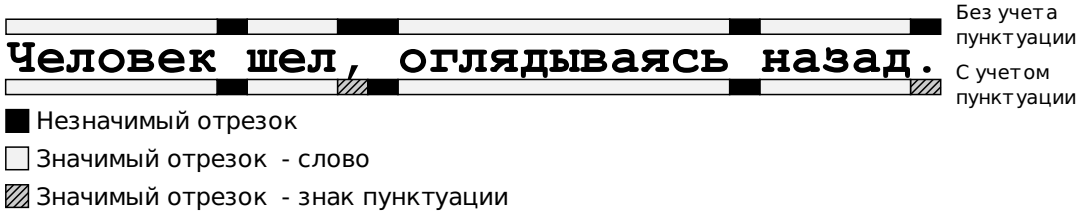

Рис. 1: Разбиение текста на отрезки

3 Внутренние структуры данных

3.1 Представление текста

Для организации эффективного выделения языковых конструкций из текста было предложено представление текста в виде графа, которым оперирует алгоритм выделения. Узлы этого графа соответствуют незначимым отрезкам текста, а ребрами являются различные синтаксические интерпретации лежащих между ними значимых отрезков текста. Под синтаксической интерпретацией отрезка текста будем понимать набор значений морфологических характеристик слов, входящих в этот отрезок. Такое представление текста в виде графа позволяет при наложении шаблонов рассматривать различные сочетания интерпретаций входящих в текст слов и словосочетаний.

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; ребра соответствуют установленным в ходе анализа синтаксическим (морфологическим) интерпретациям этих слов. Граф считается ориентированным: все ребра направлены в сторону вершин с большим номером. Кроме того, такой граф не содержит ориентированных циклов.

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

На Рис. 2 приведен пример графа текста, построенного для режима с учетом пунктуации.

В общем случае между парой вершин проходит несколько ребер — они соответствуют различным синтаксическим (морфологическим) интерпретациям слов в соответствующих отрезках текста. Как видно из рисунка, количество синтаксических интерпретаций может оказаться достаточно велико. Например, у слова «*большой*» их 6, представляющих различные сочетания морфологических характеристик: a – мужской род, именительный падеж; b – мужской род, винительный падеж; c – женский род, родительный падеж; d – женский род, дательный падеж; e – женский род, творительный падеж; f – женский род, предложный падеж.

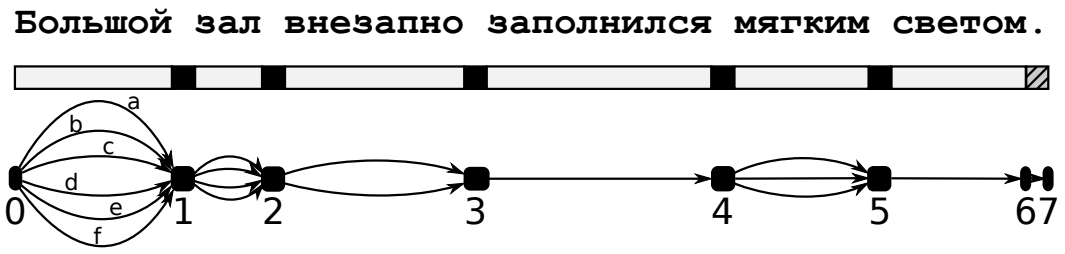

Рис. 2: Граф текста

3.2 Представление выделенных конструкций

Любая обнаруженная в тексте в результате наложения шаблона конструкция будет представляться в этом же графе текста дополнительным ребром, соединяющими вершины — конечные точки соответствующего отрезка текста. Поскольку накладываемые шаблоны конструкций могут иметь параметры, в качестве которых выступают морфологические характеристики входящих в них элементов, то и дополнительным ребрам приписываются значения этих параметров (т.е. значения морфологических характеристик элементов конструкции). В общем случае, одному и тому же отрезку может соответствовать несколько различных вариантов наложения, отличающихся только значениями морфологических характеристик. Пример представления вариантов наложения шаблона `A N <A=N>(N)` (описывающего согласованную пару из прилагательного и существительного) в графе текста приведен на Рис. 3.

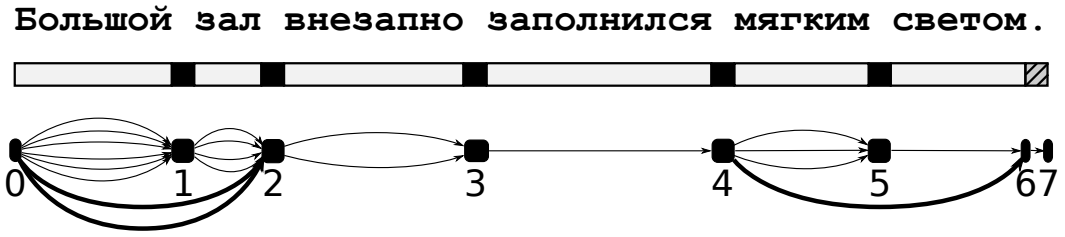

Рис. 3: Результаты наложения шаблона `A N <A=N>(N)`

На рисунке жирно выделены два ребра между вершинами 0 и 2, соответствующих двум вариантам наложения этого шаблона на отрезок текста «*большой зал*» (двум синтаксическим интерпретациям этого отрезка) и одно ребро между вершинами 4 и 6, соответствующее наложению этого шаблона на отрезок «*мягким светом*».

Рассмотренное представление в графе выделенных конструкций позволяет единообразно обрабатывать как элементы-слова, так и экземпляры шаблонов и не производить повторно выделение конструкций, соответствующих уже наложенным шаблоном.

3.3 Индексы

Для увеличения производительности наложения шаблонов одновременно с построением графа текста строится набор индексов – структур данных, позволяющих быстро определять множество ребер графа, с которых есть смысл начинать наложение шаблона. Каждый индекс

решает следующую задачу: по заданному ключу получить упорядоченный по номеру начальной вершины список ребер графа, обладающих заданными свойствами. В ходе выделения языковых конструкций используется три типа индексов:

- Индекс частей речи – ключом для этого индекса является часть речи и в качестве результата выдается множество ребер графа, представляющих все найденные в тексте словоформы заданной части речи. Индекс позволяет оптимизировать наложение шаблона в достаточно распространенной ситуации, когда шаблон начинается с элемента-слова, имеющего конкретную часть речи.

- Индекс шаблонов – ключом является некоторый уже наложенный шаблон и по нему извлекается множество всех вариантов его наложения. Таким образом оптимизируется наложение шаблона в случае, когда его первым элементом является другой, уже наложенный шаблон.

- Индекс слов – ключом является начальная форма слова, а в качестве результата выдается множество всех ребер, представляющих его словоформы в тексте. Этот индекс используется при наложении шаблонов, которые начинаются с конкретных слов (как, например, в шаблоне `A<красный>`).

На Рис. 4 изображены индексы частей речи (существительное, наречие и глагол) и индекс шаблона `A N<A=N> (N)`: индексы ссылаются на соответствующие ребра графа текста «*Большой зал внезапно заполнился мягким светом*».

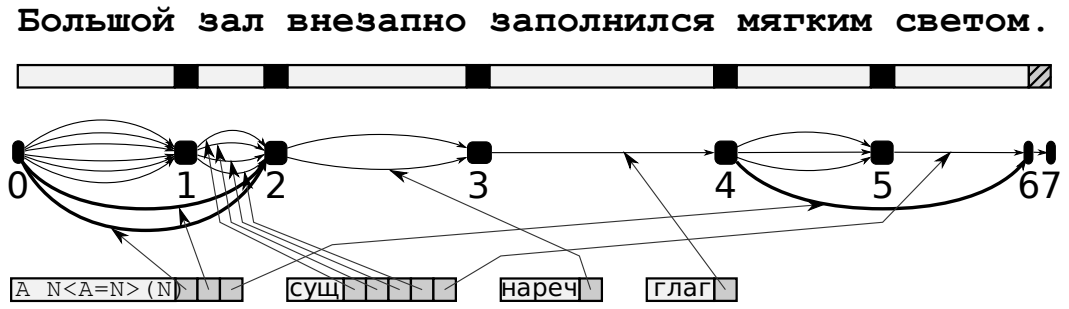

Рис. 4: Индексы в графе текста

3.4 Представление шаблонов

LSPL-шаблоны, описывающие выделяемую в тексте конструкцию, также представляются в специальной форме. В общем случае, каждый шаблон состоит из набора альтернатив, каждая из которых представляет из себя последовательность элементов шаблона.

Во внутреннем представлении каждый элемент шаблона имеет набор контекстных условий, описывающих ограничения на его наложение. Этими условиями являются различные ограничения на морфологические характеристики элементов (в частности, конкретизация падежа, рода, числа и т.п.), а также условия грамматического согласования. Однако, если в самом шаблоне они обычно расположены в конце, то при переводе шаблона во внутреннее представление условия согласования сдвигаются максимально влево, до позиции последнего элемента, участвующего в согласовании. При наложении шаблона это позволяет более эффективно отбрасывать несогласованные части конструкции.

4 Алгоритм наложения шаблона

Наложение шаблона на текст рассматривается как поиск в графе текста и включает 3 основных этапа:

I Определение начальных ребер – множества ребер, с которых может быть начат поиск в графе;

II Поиск путей в графе, соответствующих шаблону, начиная с полученных на предыдущем этапе ребер;

III Группировка вариантов наложения – найденные пути группируются и образуют варианты наложения, добавляемые в соответствующий индекс и граф в виде новых ребер.

Рассмотрим каждый этап детальнее.

4.1 Определение начальных ребер

Цель этого этапа – максимально сузить множество ребер, с которых могут начинаться языковые конструкции, выделенные по шаблону. Для этого предлагается использовать следующую схему работы этапа:

1. Для каждой альтернативы шаблона определить множество начальных элементов-слов или экземпляров других шаблонов;

2. Для каждого элемента этого множества по соответствующим индексам извлекаются начальные ребра графа, которые могут соответствовать этому элементу. В частности, для элемента-слова извлекается множество всех ребер, представляющих слова той же части речи, для экземпляра шаблона – множество всех ребер, представляющих варианты наложения этого шаблона. Если для какого-то элемента шаблона необходимых данных в индексе не находится (например, используемый шаблон еще не был наложен), то выполнение этапа заканчивается и следующий этап алгоритма (поиск путей в графе) осуществляется в предположении, что шаблон может начинаться с любого ребра;

Например, при наложении шаблона `A N<A=N>(N)` работа по указанной схеме происходит следующим образом:

1. Шаблон содержит одну альтернативу, множество начальных элементов состоит из одного элемента-слова – `A` (прилагательное, без каких-либо конкретизированных характеристик);

2. Начальные ребра определяются на основе индекса частей речи (индекс прилагательных) и формируется множество всех ребер, представляющих в тексте прилагательные в их конкретных синтаксических интепретациях; на Рис. 5 для рассматриваемого примера эти ребра выделены жирно.

Дальнейший поиск вариантов наложения в графе начинается последовательно с каждого выделенного ребра, представляющего прилагательное в тексте: шести ребер, начинающихся с узла 0 (синтаксические интепретации прилагательного "*большой*"), одного ребра, начинающегося с узла 2 (прилагательное "*внезапный*"в краткой форме) и трех ребер, начинающихся с узла 4 (прилагательное "*мягкий*").

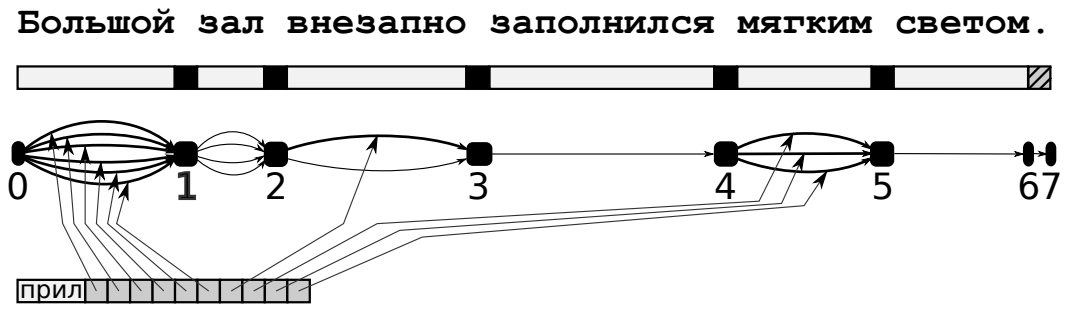

Рис. 5: Начальные ребра

4.2 Поиск путей в графе

На этом 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 проверяются все допустимые продолжения пути, которые соответствуют текущему элементу шаблона (например, слову определенной части речи или варианту наложения заданного шаблона) и его контекстным условиям (например, конкретизированные падеж, род и другие морфологические характеристики).

Множество рассматриваемых при поиске путей образует так называемое дерево поиска, которое получается склеиванием путей графа с одинаковым префиксом. Дерево поиска отражает процесс перебора ребер графа при поиске пути. Например, для шаблона `A N<A=N>(N)` обход графа осуществляется по путям, показанным на Рис. 6.

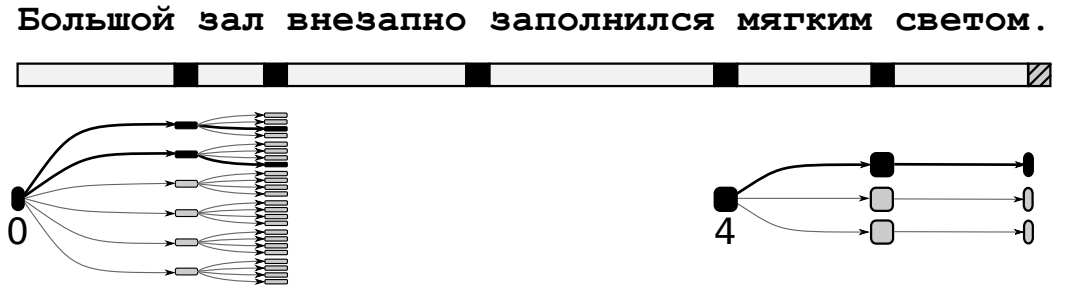

Рис. 6: Деревья поиска шаблона `A N<A=N>(N)`

На рисунке жирным выделены пути в деревьях, приводящие к успешному наложению шаблона: это 3 пути, удовлетворяющих условию грамматического согласования из 27 возможных в графе путей, соответствующих последовательности элементов накладываемого шаблона. Таким образом, условия согласования позволяют отбросить значительное количество результатов.

Перед описанием третьего этапа рассмотрим подробнее два важных момента на этапе поиска путей в графе – обработку условий согласования и сложных конструкций.

4.3 Обработка условий согласования и сложных конструкций

Для обработки условий согласования в процессе наложении шаблона необходимо сравнивать характеристики различных элементов. Один из способов обработки состоит в том, чтобы

проверять условия согласования после того, как вся последовательность ребер графа, образующих вариант наложения, построена. Однако значительно более эффективной является проверка условий согласования сразу, как только становятся конкретизированы все входящие в условие морфологические характеристики.

Для этого 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 проверяются на предмет выполнения условия. Поскольку во внутреннем представлении шаблонов условия согласования максимально сдвинуты влево, такой способ обеспечивает более раннее отсечение несогласованных вариантов наложения.

На Рис. 7 приведено дерево поиска вариантов наложения шаблона `A N Av V <A=N,N=V,Av=V>` (прилагательное `A` согласовано с существительным `N`, существительное – с глаголом `V`, а глагол – с наречием `Av`) на текст «*Большой зал внезапно заполнился мягким светом*»

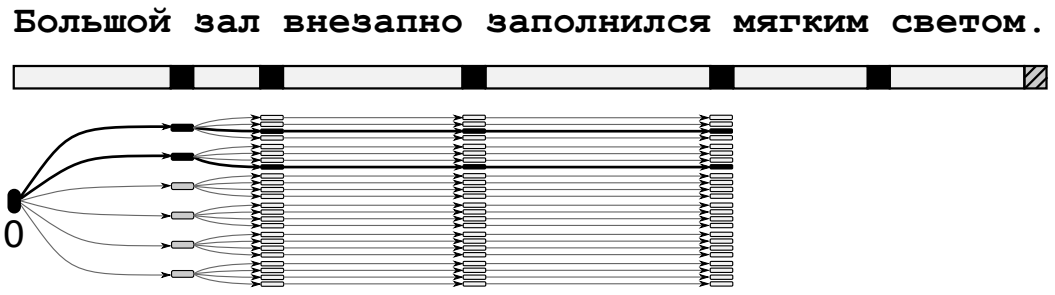

Рис. 7: Дерево поиска шаблона `A N Av V <A=N,N=V,Av=V>`

Жирными линиями на рисунке отмечены два пути, приводящие к успешному наложению. Если осуществлять проверку согласования после рассмотрения всех элементов шаблона, то это привело бы к рассмотрению всех путей, изображенных на рисунке. Однако то, что согласование `A=N` может быть проверено сразу после обработки отрезка текста «*зал*» позволяет обрывать пути в дереве поиска, рассматривая только те, которые выделены жирным на рисунке. Как видно, это приводит к значительному сокращению множества рассматриваемых путей.

Сложные элементы шаблона, т.е. повторения (например, `{A}`) и экземпляры шаблонов (например, использование шаблона `NG` в шаблоне `NG V`) 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), процесс наложения сложных элементов включает добавление специального группирующего ребра над всей последовательностью, соответствующей сложному элементу.

При обработке экземпляра шаблона возможна ситуация, когда соответствующий шаблон еще не был наложен и, следовательно, его варианты наложения отсутствуют в графе текста – в этом случае с вершины графа, на которой остановился процесс поиска, начинается процесс наложения этого шаблона.

4.4 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, возможны различные варианты наложения шаблона на один и тот же отрезок текста.

Количество вариантов наложения очень быстро растет с увеличением длины и сложности шаблона, что приводит к увеличению расхода памяти на поддержание графа и значительному снижению эффективности поиска. Кроме того, для человека описанная ситуация выглядит как множество практически неотличимых друг от друга вариантов наложений шаблона на один и тот же отрезок текста.

Для того, чтобы избежать подобных проблем в ходе поиска, в графе используется информация о параметрах шаблона: предполагается, что те морфологические характеристики, которые не вошли в параметры шаблона, не являются важными (в том смысле, что могут быть опущены при дальнейшем анализе результатов человеком и наложении других шаблонов). Например, для шаблона `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 значениями характеристик, не являющихся важными, что приводит к значительному уменьшению количества вариантов наложения.

Например, шаблон `A N<A=N>(N)` имеет 2 варианта наложения на отрезке «*Большой зал*» (см. Рис. 6), соответствующие именительному и винительному падежу слова «зал» (результаты сгруппированы только по синтаксическим интепретациям прилагательного), а шаблон `A N <A=N>`, отличающийся только тем, что не имеет параметров после группировки имеет уже только один вариант наложения на тот же отрезок (результат группировки по синтаксическим интепретациям как прилагательного, так и существительного).

5 Заключение

Описанный метод был реализован в программном комплексе, предназначенном для анализа русскоязычных текстов с использованием лексико-синтаксических шаблонов. Реализованный программный комплекс состоит из четырех компонентов:

- Ядро, написанное на языке C++, и реализующее предложенный метод;
- Набор консольных утилит, реализованных на языке C++, предназначенных для автоматического анализа текста и интеграции ядра с различными скриптами;
- Прикладной интерфейс для языка Java, позволяющий интегрировать ядро в приложения обработки естественного языка, разработанные на языке Java;
- Графический пользовательский интерфейс для анализа текста лингвистом.

Комплекс был успешно протестирован в задаче распознавания регулярных конструкций русского языка [2] и в настоящее время используется для решения задачи терминологического анализа русскоязычного текста.

Представленный метод позволяет осуществлять выделение в тексте на русском языке различных языковых конструкций, описанных в виде LSPL-шаблонов и применим в различных областях обработки текста на естественном языке.

Список литературы

[1] 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.

[2] Большакова Е.И., Васильева Н.Э. *Формализация лексико-синтаксической информации для распознавания регулярных конструкций естественного языка.* Программные продукты и системы, 2008, № 4, с. 103 - 106.

[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] Резникова Т.И., Копотев М.В. *Лингвистически аннотированные корпуса русского языка (обзор общедоступных ресурсов)* Национальный корпус русского языка: 2003—2005. М.: Индрик, 2005.

[5] Bontcheva K., Cunningham H., Maynard, D., Tablan, V., and Saggion, H. *Developing Reusable and Robust Language Processing Components for Information Systems using GATE.* In: Proceedings of the 13th international Workshop on Database and Expert Systems Applications, DEXA. IEEE Computer Society, Washington, DC. 2002, pp. 223-227.

[6] Petasis G., Karkaletsis V., Paliouras G., Androutsopoulos I. and Spyropoulos C. D. *Ellogon: A New Text Engineering Platform.* In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002). Las Palmas, Canary Islands, 2002, pp. 72-78.