Применение лексико-синтаксических шаблонов для автоматизации процесса построения онтологий

© Рабчевский Е. Булатова Г. Шарафутдинов И.

Пермский государственный университет

evgeny@ranat.ru, bulatovag@gmail.com, igor@ranat.ru

Аннотация

Рассматривается использование лексико-синтаксических шаблонов для автоматизации построения семантических моделей, соответствующих тексту. Обсуждаются шаблоны, результаты их применения, а также методы выявления новых шаблонов, опубликованные в работах других авторов. Формулируется оригинальный подход к формализации лексико-синтаксических шаблонов. Разработана XML Схема языка для формализации шаблонов. Разрабатывается программа для выявления синтаксических групп на основе синтаксического анализатора Dictum. На основе шаблонов решается задача построения таксономии понятий.

1. Введение

Онтологии активно используются научными сообществами для описания терминологии, в электронной коммерции – для описания товаров и услуг, в других приложениях в качестве баз знаний интеллектуальных систем, в Интернет – для организации данных. Поисковой системой SWOOGLE на сегодня проиндексировано свыше 10 тысяч онтологий и словарей, доступных в Веб. Все больше Интернет ресурсов внедряют семантические модели в свою архитектуру. Появляется все больше промышленных средств для работы с онтологиями.

Существует достаточное количество технических средств для создания и редактирования онтологий, однако создание онтологии требует от автора экспертных знаний в исследуемой предметной области. Поэтому задача автоматизации процесса построения онтологий является весьма актуальной.

В Интернет можно найти описательную часть большинства предметных областей человеческой деятельности. В основном этот контент представлен текстами на естественном языке. Поэтому задача формирования онтологий из текстового содержания Веб ресурсов представляется наиболее интересной.

2. Лексико-синтаксические шаблоны

Лексико-синтаксические шаблоны позволяют построить семантическую конструкцию, которая соответствует концептуальному содержанию единицы текста. Для этого используются особенности языка, на котором представлен текст.

Марти Хэрст выявила существенное количество шаблонов для идентификации отношения гипономии [1]. Её исследования показали, что, используя шаблоны на большом корпусе текстов определенной тематики, можно построить «достаточно адекватную» таксономию понятий соответствующей предметной области. В её шаблонах в качестве элементов используются, например, понятие именной группы, знаки препинания, конкретные слова и другое.

Так, например шаблон «NP {, NP}* {,} and other NP», где NP – условное обозначение именной группы, идентифицирует отношение гипонимии, которое продемонстрировано на части предложения «... temples, treasuries, and other important civic buildings …». С помощью указанного шаблона из данного фрагмента текста могут быть выявлены следующие отношения hyponym(\temple", \civic building"), hyponym(\treasury", \civic building").

Большакова и др. сформулировали язык для записи лексико-синтаксических шаблонов [2]. По их мнению, элементами шаблонов могут быть:

- литералы, т.е. конкретные лексемы;
- определенные части речи;
- определенные грамматические конструкции;
- условия, уточняющие грамматические характеристики рассмотренных элементов.

Важно отметить, что исследования Большаковой направлены на выявление определений «новых» терминов в научной литературе, то есть соответствующие шаблоны ориентированы на ограниченный тип отношений.

3. Подходы к автоматизации выявления шаблонов

В работе Хэрст [3] предложен способ выявления новых шаблонов. Метод состоит из пяти этапов:

- выбирается некое отношение R, например гипонимии;
- подбирается список терминов, для которых заранее известно, что они участвуют в выбранном отношении с другими терминами (в качестве обучающей выборки был использован тезаурус WordNet);
- в исследуемом корпусе находятся места, в которых появляются выбранные термины, участвующие в заданном отношении, «их окружение» фиксируется;
- найденные «окружения» анализируются, и формулируется новый шаблон для заданного отношения.

Недостатком данного метода является недостаточная формализация понятия окружения, а также выполнение 4-го этапа без использования средств автоматизации.

В работе сотрудников Шеффилдского университета [4] сделана попытка автоматизации формулировки шаблонов из «окружения» терминов, состоящих в заданном отношении. В основе их работы лежит метод машинного обучения Lazy-NLP [5]. Алгоритм оперирует с морфологией текста и лексическими категориями (например, частями речи) и также использует обучение с учителем, в ходе которого пользователь указывает верные и неверные лексикализации выбранного отношения.

4. Формализация шаблонов

Предыдущие исследования [6] показали, что использование шаблонов может быть эффективным для построения разнообразных типов отношений.

Интересно использование шаблонов для выявления онтологических конструкций.

Для записи лексико-синтаксических шаблонов автор предлагает использовать унифицированный язык, который оперирует с понятиями графематики, морфологии и синтаксиса.

Для унификации записи шаблонов предлагается описывать шаблон с помощью входной и выходной схем. Входная схема позволяет иденти-

фицировать шаблон в тексте, выходная - описывает семантическую конструкцию, которая соответствует концептуальному содержанию текста.

В рамках данного исследования был произведен анализ шаблонов других авторов, а также сформулированы оригинальные лексико-синтаксические шаблоны. В результате анализа общего объема лексико-синтаксических шаблонов автор пришел к выводу, что в задаче выявления семантических моделей онтологического характера, элементами входной схемы шаблона должны быть:

- литерал;
- синтаксическая единица, такая как:
  - слово, являющееся определенной словоформой определенной леммы;
  - слово определенной части речи;
  - синтаксическая группа (грамматика непосредственно составляющих [7]);
- синтаксические элементы могут накладываться грамматические условия.

Элементы шаблона могут участвовать между собой в отношениях грамматики зависимостей [7], то есть один элемент может быть подчинен другому.

5. Синтаксическое дерево предложения и синтаксические группы

Синтаксические группы состоят из последовательности слов, входящих в часть текста. Группа характеризуется типом, который определяется типом ее главной подгруппы. Тип подгруппы определяется типом (частью речи) главного слова. Примером синтаксической группы является именная группа, группа в которой главная подгруппа – имя существительное. Таким образом, предложение можно рассматривать, как две главные группы, подлежащего и сказуемого.

Если рассматривать предложение, как дерево, в котором вершинам соответствуют слова предложения, а ребрам – подчинительные связи. И каждому ребру дерева соответствует вопрос от главного слова к зависимому.

То синтаксическую группу можно понимать как определенную сосредоточенную часть дерева или подграф. Тип группы определяется типом главного (то есть верхнего) слова в группе.

Таким образом, можно ввести понятие группы определенного слова. Это группа, главным словом в которой является заданное слово, само

слово соответственно входит в группу. А группа, подчиненная определенному слову – это группа этого слова без самого этого слова, или подграф, подчиненный узлу заданного слова. Главное слово группы само по себе может интерпретироваться, как член предложения.

Очевидно, что в графе предложения несколько узлов могут иметь одного общего родителя.

6. LSPL - язык для записи шаблонов

Предлагается XML-язык для формализации шаблонов. Входная схема шаблона записывается на XML-основанном языке LSPL (Lexical-Syntactic Pattern Language – язык разметки лексико-синтаксических шаблонов).

В элементе <inputShema> записывается входная схема шаблона, как последовательность элементов шаблона, записываемых в тегах <element>. В атрибутах данного тега записываются тип элемента (literal, wordForm, partOfSpeech и syntacticGroup), идентификатор элемента в шаблоне по порядку id, идентификатор элемента, которому подчиняется данный элемент connectedId, и флаг обязательности присутствия presence.

У элемента <element> также выделяется атрибут mainWord. Атрибут используется приложением только в случае, если тип элемента шаблона – синтаксическая группа. Наличие данного атрибута необходимо для установления того, в каком объеме синтаксическая группа участвует в шаблоне. Возможны случаи, когда элементом шаблона является только главное слово, вся группа вместе с главным словом, или группа без главного слова. Соответственно значения атрибута будут only (или без указания данного атрибута), included, not_included.

В дочернем элементе <content> указывается содержание элемента, которое для литерала означает слово, которое должно быть элементом шаблона, для словоформы – лемму, от которой берется словоформа. Либо часть речи или синтаксическая группа, которыми можно охарактеризовать элемент шаблона.

Для литерала значение тега <content> «plainText» зарезервировано под обозначение части текста, которая также участвует в шаблоне, но на нее не накладывается каких-либо лингвистических условий. Например, это может быть использовано для обозначения элемента в шаблоне, к которому должны быть применены другие шаблоны.

В элементе <grammaticValue>, дочернем для <element>, в виде атрибутов указываются дополнительные грамматические условия, которые накладываются на элемент, такие как род, число и падеж. Как одно из возможных значений указанных атрибутов может быть указано значение переменной, которое может быть использовано прикладной программой.

Тело тега может использоваться, для обозначения всех грамматических характеристик с помощью одного идентификатора.

У одного элемента <element> может быть несколько элементов <grammaticValue>. Это позволяет формулировать шаблоны в более общей форме, например, шаблон может позволять элементу находиться в нескольких падежах.

Рассмотрим пример записи шаблона следующего вида:

[NP1] – это [NP2], [который (грамматически согласованный с NP2)] [V] [NP3 (подчиненная V)].

Где в квадратных скобках обозначены элементы шаблона, NP – именная группа, V – глагол.

Представим входную схему данного шаблона

```
<pattern>
  <inputShema>
    <element type=”syntaxGroup” id=”1” >
      <content>nounPhrase</content>
    </element>
    <element type=”literal” id=”2” >
      <content>-</content>
    </element>
    <element type=”literal” id=”3” >
      <content>это</content>
    </element>
    <element type=”syntaxGroup” id=”4” >
      <content>nounPhrase</content>
      <grammaticValue>var1<grammaticValue>
    </element>
    <element type=”literal” id=”5” >
      <content>,</content>
    </element>
    <element type=”wordForm” id=”6” >
      <content>который</content>
      <grammaticValue>var1<grammaticValue>
    </element>
    <element type=”partOfSpeech” id=”7” >
      <content>verb</content>
    </element>
    <element type=”syntaxGroup” id=”8” >
      <content>nounPhrase</content>
    </element>
  </inputShema>
```

```
</pattern>
```

В атрибуте path элемента <outputShema> указывается ссылка на OWL файл выходной схемы шаблона, в котором указывается соответствующая семантическая модель. Либо сам OWL-код записывается в теле тега <outputShema>. Приведем выходную схему данного шаблона.

```
…
<owl:Ontology rdf:about=""/>
<owl:Class rdf:ID="element4"/>
<owl:Class rdf:ID="element8"/>
<owl:Class rdf:ID="element1"/>
  <rdfs:subClassOf rdf:resource="#element4"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#element7" />
      <owl:hasValue rdf:resource="#element8" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:about="#element7">
  <rdfs:domain rdf:resource="#element4"/>
</owl:ObjectProperty>
…
```

7. Сложные шаблоны и группы шаблонов

Рассмотрим несколько шаблонов:

[NP1] – это [NP2], [который (грамматически согласованный с NP2)] [V] [Vinf] [NP3 (подчиненная Vinf)],

[NP1] – это [NP2], [который (грамматически согласованный с NP2)] [V] [P],

где Vinf – глагол в инфинитиве, P – предикатив.

Эти шаблоны и шаблон из примера шестой главы частично совпадают по входной и выходной схемам. Так в выходной схеме всех упомянутых шаблонов будет присутствовать конструкция, приведенная в примере главы 6. Различия будут заключаться в том, каким образом будут называться свойство, на которое накладывается ограничение owl:hasValue и соответствующие классы.

В этом смысле приведенные шаблоны целесообразно записывать в виде одного (сложного шаблона), как конструкцию из элементов, указанных в главе 4, и литерала со значением «plainText». А при применении данного шаблона к тексту – помимо самого шаблона, к литералу со зна-

чением «plainText» и части сложного шаблона применять другой или другие шаблоны.

Шаблоны, которые могут применяться к части сложного шаблона и к литералу со значением «plainText» в рамках одного шаблона могут быть объединены в группы, принадлежность к которым указывается в атрибуте group тега <pattern>.

Понятие группы шаблонов должно упростить запись шаблонов.

Необходим механизм для работы со сложными шаблонами и группами шаблонов, однако на данный момент, данный формализм не реализован.

8. Применение LSPL

В рамках данного исследования была проведена попытка использования лингвистических процессоров [8] и синтаксического анализатора Dictum [9] для идентификации шаблонов в тексте. Однако оба средства не обеспечили полной функциональности, заложенной в LSPL.

На данный момент разрабатывается программа, которая позволила бы выявлять синтаксические группы на основе синтаксического анализатора Dictum.

Последний предоставляет на выходе текст входного предложения в виде дерева, в узлах которого находятся слова с обозначенными частями речи и грамматическими условиями.

В дальнейшем планируется использовать полученное средство и несколько шаблонов, подобных указанному в примере, для построения таксономии понятий для ограниченной предметной области.

9. Заключение

Создан формализм LSPL. Использование созданного языка LSPL позволит формировать систематизированную, открытую для развития среду лексико-синтаксических шаблонов.

Предполагается, что данный формализм послужит базой для последующего исследования методов автоматического формирования шаблонов.

Авторы считают, что создание данного формализма является ключевой задачей в контексте применения лексико-синтаксических шаблонов для автоматизации процесса построения семантических моделей.

Авторы благодарят профессора, заведующего кафедрой КСиТ Пермского государственного университета, М.А. Марценюка за обсуждение работы.

Литература

[1] Automatic Acquisition of Hyponyms from Large Text Corpora. Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes France, July 1992

[2] Большакова Е.И., Баева Н.В., Васильева Н.Э. Структурирование и извлечение знаний, представленных в научных текстах // Девятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2004. Труды конференции в 3-х томах. Т. 2. М.: Физматлит, 2004.

[3] Hearst, M.A. Automated Discovery of WordNet Relations. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database. MIT Press, Cambridge, 1998.

[4] Christopher Brewster, Fabio Ciravegna and Yorick Wilks. User Centred Ontology Learning for Knowledge Management. Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems, Stockholm, June 27-28, 2002, Lecture Notes in Computer Sciences, Springer Verlag.

[5] Fabio Ciravegna: Adaptive Information Extraction from Text by Rule Induction and Generalisation. in Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001) , Seattle, August 2001.

[6] Рабчевский Е.А., Булатова Г.И., Автоматическое построение онтологий // Научно-технические ведомости СПбГПУ № 4 2007. – Санкт-Петербург: Издательство Политехнического Университета 2007.

[7] Марчук Ю.Н. Компьютерная лингвистика: учебное пособие – М.: АСТ: Восток – Запад, 2007.

[8] Автоматическая обработка текста http://www.aot.ru

[9] Синтаксический анализатор Dictum http://www.dictum.ru

Application of lixical-synatactic patterns to the automation of ontology building process

Rabchevsky E., Bulatova G., Sharafutdinov I.

The paper is devoted to the usage of lexical-syntactic patterns for the automation of semantic model extraction from text process. Some patterns, results of their usage and extraction of new patterns that were published in paper of other authors are discussed. We offer the original way for lexical-syntax pattern formalization. We name it

LSPL - Lexical-Syntactic Pattern Language and we create proper XML Schema. The program for extraction of syntactic groups based on syntactic analyzer Dictum is developed. We plan to use some patterns and program for extraction of syntactic groups to build taxonomy some domain.