Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска

© Рабчевский Е.А.

Пермский Государственный Университет
Кафедра компьютерных систем и телекоммуникаций
seus@rabchevsky.name

Аннотация

Обсуждается проблема автоматического построения онтологий на основе семантического анализа текстов на естественном языке. В качестве метода предлагается использование лексико-синтаксических шаблонов. Раскрывается синтаксис и семантика языка лексико-синтаксических шаблонов LSPL. Описывается программный комплекс, который позволяет:
- хранить шаблоны и корпус текстов на русском языке в базе данных
- редактировать и проводить валидацию шаблонов на корпусе русскоязычных текстов
- проводить семантический анализ текстов корпуса на основе шаблонов.
Для оценки предложенной методики семантического анализа предлагается оценивать результаты применения методики в приложении к информационному поиску. Предлагается модель информационного поиска на основе метрик TF*IDF, в которой понятие терма заменяется триплетом (атомарной единицей результатов семантического анализа). Обсуждаются результаты применения предложенной модели поиска к заданиям семинара РОМИП'2009.

1 Введение

Важнейшей проблемой в развитии Интернет является его интеллектуализация, и связанные с этим интеграция данных, качественный поиск, интеграция Веб служб и многое другое. В рамках подхода Semantic Web предлагаются эффективные средства для решения указанных задач.

Однако данные технологии предполагают наличие качественных источников семантических данных. Сегодня к таким можно отнести лишь источники, созданные при значительной поддержке корпораций или государственных структур, например база знаний Cyc или разработка языка DAML военным ведомством США. Также существуют достаточно качественные источники знаний, переведенные в форматы знаний Semantic Web из накопленных за многие годы источников. Например, онтологии генов, географических названий или RDF представление тезауруса Wordnet.

В общеупотребительных предметных областях не представлено открытых источников данных, которые бы реально повторно использовались Интернет сообществом. В этом отношении лидером можно считать RDF представление Wikipedia – DBPedia.

В целом, можно утверждать, что повторное использование и интеграция с указанными источниками данных находится на низком уровне. Это связано с тем, что источники данных не так совершенны, чтобы разработчики приложений могли с удобством их использовать или интегрировать в свои приложения. При этом потребность в приложениях подобного плана есть во всех сферах общества, отраженных в Интернет.

С русскоязычными источниками данных дело обстоит еще тяжелее.

В связи с этой проблемой проблема автоматического формирования онтологий на основе анализа текстов на естественном языке является весьма актуальной. Это подтверждается рядом современных исследований в данной области.

Методы автоматического построения используют средства компьютерной лингвистики, включающие все уровни анализа естественного языка, графематику, морфологию, синтаксис и семантику. Отличие между различными системами заключается в полноте комбинирования уровней анализа. Однако существенных результатов в данной области, особенно в применении к русскому языку, не представлено.

Распространенный критерий качества онтологии основан на оценке работы приложения, использующего онтологию. Поэтому оценка автоматически построенных онтологий является еще отдельной сложной задачей, как и их построение. Ввиду наличия отработанных методик

по оценке качества информационного поиска, последний можно рассматривать, как приложение, с помощью оценки которого, можно оценивать качество соответствующих онтологий.

Данная работа является попыткой решения проблемы автоматизации построения онтологий и оценки полученных результатов.

2 Лексико-синтаксические шаблоны как метод семантического анализа текста

Лексико-синтаксические шаблоны представляют собой характерные выражения (словосочетания и обороты), конструкции из определенных элементов языка. Такие шаблоны позволяют построить семантическую модель, соответствующую тексту, к которому они применяются.

Как метод семантического анализа, лексико-синтаксические шаблоны используются в компьютерной лингвистике более 20-ти лет. В своих исследованиях мы использовали работы таких авторов, как Марти Хэрст, Е.И. Большакова , Christopher Brewster, Fabio Ciravegna и Yorick Wilks; Ермаков А.Е., Плешко В.В., Митюнин В.А., Плешко В.В. (компания RCO).

Марти Хэрст предположила, что лексические отношения можно описать с помощью метода интерпретации образцов (шаблонов). Такой метод использует иерархию шаблонов, которые состоят главным образом из индикаторов части речи и групповых символов.

Хэрст выявила существенное количество шаблонов для идентификации отношения гипонимии [1]. Её исследования показали, что при использовании шаблонов на большом корпусе текстов одной тематики, можно построить «достаточно адекватную» таксономию понятий соответствующей предметной области. В её шаблонах в качестве элементов используются, например, понятие именной группы (NP), знаки препинания, конкретные слова.

Таким образом, шаблон «NP {, NP}* {,} and other NP», где NP – условное обозначение именной группы, определяет отношение гипонимии, которое продемонстрировано на части предложения «... temples, treasuries, and other important civic buildings …». С помощью указанного шаблона могут быть выявлены следующие отношения:

hyponym(\temple", \civic building"),

hyponym(\treasury", \civic building").

Группа разработчиков во главе с Большаковой сформулировала язык для записи лексико-синтаксических шаблонов (LSPL) [2]. По её мнению, элементами шаблонов, для наиболее точного описания, могут быть:

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

Разработанные ее коллективом шаблоны применяются для анализа научно-технических документов. Для их обработки кроме традиционных словарей (терминологического и морфологического), используется словарь общенаучных слов и выражений, лексико-синтаксические шаблоны типичных фраз научной речи.

На пример, предложение:

«По результатам генерации форм, слова были разбиты на группы, названные профилями» с помощью разработанной методики формализации авторы записывают так:

Ng «,» Pa<«названный»> T<:case=ins>

Ng.gender=Pa.gender

Ng.number=Pa.number=T.number>

Метод, разработанный Кристофером Бревстером [3] основан на разработках Марти Хэрст, и в качестве элементов шаблонов предполагается использовать словарную форму, представляющую лексему в словаре (lexical item, lemma), part of speech, syntactic role.

Для построения более четкой онтологии, с помощью указанного метода, приходится накладывать ограничения на анализируемую область, что направлено на повышение эффективности процесса обработки текста.

Командой разработчиков из компании RCO был разработан модуль, позволяющий производить сравнение цепочек лексем, заданных своими описаниями.

Описание лексемы содержит набор предопределенных атрибутов (всего более 20-ти), например:

Token.Text - строка лексемы;

Token.Type - тип лексемы (известное/неизвестное слово русского языка, латинское слово, специальная конструкция);

множество грамматических характеристик - Morph.SpeechPart (часть речи), Morph.Case (падеж), Morph.Gender (род), Morph.Number (число) и т.п.

Ориентация, описанных шаблонов, направлена на выявление специфических объектов, таких как: даты, адреса, имена юридических организаций и т.п.

В целом, можно сказать, что лексико-синтаксические шаблоны как метод семантического анализа текста являются достаточно эффективным средством при условии большого объема шаблонов (разумеется, объем зависит от специфики задачи).

Также стоит заметить, что данная методика достаточно дорогостоящая в плане процессорного времени, потому как использует все уровни анализа естественного языка.

3 Новизна и уникальность работы

Рассмотрим данную работу в сравнении с приведенными выше исследованиями.

Выделим два критерия, по которым можно проводить анализ данной работы в отношении с

другими работами. Это – цель семантического анализа и оценка качества полученных результатов.

Основным отличием является цель семантического анализа, который проводится в данной работе. В нашем случае – это построение онтологических конструкций, соответствующих тексту, на открытых языках представления знаний Semantic Web. Это предполагает использование на выходе системы форматов данных, схем и словарей, широко используемых Интернет сообществом, поддерживающим подходы Semantic Web и Linked Data. Также можно выделить отсутствие специализации методики семантического анализа на определенную задачу или предметную область, как например выделение фактографической информации в разработках RCO или анализ научно-технической прозы в работах Большаковой.

Соответственно отличается и использование результатов анализа, в нашем случае – это приложения Semantic Web, семантическая разметка ресурсов и др.

Если говорить об оценке результатов, то ввиду сложности самой задачи, в работах отечественных авторов этот вопрос не поднимается. В работах Шеффилдского университета оценка применения шаблонов и грамматик к тексту проводится. Однако они больше используются для автоматической аннотации текста, нежели построения онтологий. В данной работе оценка проводится опосредованно, что конечно вносит свое влияние, но при наличии фиксированной модели информационного поиска, на основе которого проводится оценка, можно однозначно судить о качестве построенных онтологий.

Автор считает, что использование лексико-синтаксических шаблонов как средства для автоматического построения онтологий является обоснованной методикой. Это обусловлено тем, что например, в работах Марти Хэрст получены качественные результаты выделения отношения гипономии из текста с помощью данной методики. Существуют и другие примеры, демонстрирующие состоятельность данной методики, реализованные также в работах Шеффилдского университета [21], однако в применении к автоматической аннотации.

4 Язык лексико-синтаксических шаблонов LSPL (ПГУ)

Автор занялся разработкой собственной системы формализации лексико-синтаксических шаблонов в 2006 г. [5], что по времени совпадает или на год позже разработок коллектива Большаковой. В тот момент также было выбрано название LSPL, однако о разработке Большаковой нам тогда известно не было.

Во избежание путаницы между данными системами формализации следует заметить, что далее речь пойдет только о данной разработке.

Формализация шаблонов производится на XML-подобном языке LSPL (Lexical-Syntactic Pattern Language). Тело шаблона состоит из входной и выходной схем. Входная схема – характерное описание части предложения, по которому в сочетании с входным текстом, можно однозначно построить выходную семантическую модель, соответствующую анализируемому тексту. Выходная семантическая модель представляется набором RDF триплетов, состоящих из субъекта, объекта и предиката.

Входная схема шаблона записывается в элементе <inputSchema>, который находится в теле основного тега <pattern>. Описание соответствующих элементов шаблона производится в теге <element>, дочернем относительно <inputSchema>. Атрибутами данного тега являются:

- type – тип элемента (literal, wordForm, partOfSpeech и syntacticGroup);
- id – идентификатор элемента в шаблоне по порядку;

В дочернем, относительно <element>, элементе <content> указывается содержание элемента шаблона. На данный момент поддерживаются следующие элементы:

1) Literal –слово, указанное внутри <content>: и, или, это

Пример:

```
<element type=”literal” id=”1”>
    <content>это</content>
</element>
```

2) wordForm – форма указанного слова

Пример:

```
<element type=” wordForm” id=”1”>
<content [грамматические значения] > место</content>
</element>
```

Рассматриваются любые или конкретно указанные формы слова внутри <content>

3) partOfSpeech – слово указанной части речи. На данный момент поддерживаются глаголы, существительные, прилагательные, числительные, предлоги, местоимения и наречия.

.Пример:

```
<element type=”partOfSpeech” id=”1”>
<content [грамматические значения] > noun</content>
</element>
```

4) syntacticGroup – синтаксическая группа, состоящая из нескольких слов, идущих подряд. В <content> содержится часть речи главного слова

Пример:

```
<element type=” syntacticGroup” id=”1”>
<content> noun</content>
</element>
```

5) punctualMark – под него подходит знак препинания. Если <content> пуст – то подойдёт любой знак препинания.

```
<element type=” punctualMark” id=”1”>
<content>,</content>
</element>
```

Вид выходной схемы описывается в теге <outputSchema>, дочернем для <pattern>. Элементы

каждого триплета образца выходной модели записываются в теге <statement>, в котором указывается URI-ссылки субъекта, объекта и предиката.

Применение лексико-синтаксических шаблонов заключается в следующем:

1. на вход анализатора или интерпретатора языка LSPL подается входная схема шаблона и анализируемый текст;
2. анализатор, используя отмеченные в шаблоне свойства термов (часть речи, число, род, падеж, наклонение), последовательность слов, строит семантическую карту анализируемого текста в соответствии с выходной схемой шаблона

Пример:

Анализируемое предложение	Студент - это человек, который учится в университете
Входная схема	<pattern> <inputSchema> <element type="partOfSpeech" id="1"> <content>noun</content> </element> <element type="punctualMark" id="2"> <content>-</content> </element> <element type="literal" id="3"> <content>это</content> </element> <element type="partOfSpeech" id="4"> <content>noun</content> </element> <element type="punctualMark" id="5"> <content>,</content> </element> <element type="wordForm" id="6"> <content>который</content> </element> <element type="partOfSpeech" id="7"> <content>verb</content> </element> </inputSchema>
Выходная схема	<outputSchema> <statement> <subject>http://result/subject/##1##</subject> <object>http://result/object/###4###</object> <property>http://result/property/#subClassOf</property> </statement></outputSchema></pattern>
Полученные триплеты (за исключением вспомогательных триплетов языка RDFS)	http://result/subject/Студент http://result/property/#subClassOf http://result/bject/человек

5 Интерпретатор языка лексико-синтаксических шаблонов LSPL

Для обеспечения семантики языка LSPL использовался синтаксический анализатор DictaScope [6]. Анализатор предоставлен компанией "Диктум" для некоммерческого использования и с согласия разработчика морфологического словаря А.Коваленко.

Для практического использования лексико-синтаксических шаблонов нами была разработана на языке Java библиотека PatternLib. Библиотека разделена на пакеты со следующей функциональностью

- взаимодействие с синтаксическим анализатором DictaScope
- обработка шаблонов и их применение к тексту
- визуализация полученных при анализе RDF графов с помощью библиотеки GraphViz [7]
- хранения шаблонов и анализируемых документов в базе данных.
- парсинг текстовых коллекций семинара РОМИП’2009 [8]

О последних двух пакетов подробнее будет сказано главе, посвященной участию в семинаре РОМИП.

Для использования возможностей библиотек DictaScope и GraphViz на платформе Java были разработаны две вспомогательные DLL-библиотеки, использующие для связи механизм JNI: LoaderLib.dll и GraphLib.dll соответственно.

Библиотека PatternLib сама не имеет пользовательского интерфейса и используется посредством следующих программам:

- программа редактирования и валидации шаблонов
- классы поисковой системы SEUS, отвечающие за индексацию документов, подробнее об этом будет сказано в главе, посвященной поиску на основе семантики
- online версия анализатора, которая доступна на сайте проекта SEUS [9].

Алгоритм применения шаблонов к тексту выглядит следующим образом.

Берётся 1-ый элемент шаблона, после чего последовательно перебираются элементы предложения и уровни вложенности для

синтаксических групп, до тех пор, пока не найдётся соответствующий элемент предложения.

После нахождения соответствия для 1-ого элемента аналогичная операция проводится для 2-ого элемента шаблона, но поиск ведётся с позиции, следующей за позицией первого элемента. Аналогично для следующих элементов шаблона.

Если на шаге N для N-ого элемента не нашлось лексикализации, то алгоритм поиска возвращается на уровень N-1.

Если найдено соответствие для последнего элемента шаблона (т.е. найдено соответствие для всего шаблона) – то формируются результирующий набор триплетов на основе выходной схемы шаблона..

После обработки выходной схемы алгоритм возвращается на уровень N-1 для поиска остальных возможных вариантов вхождения.

В дальнейшем планируется использовать поиск лексикализаций на основе би-деревьев.

6 Online демонстрация семантического анализатора на основе языка LSPL

Для демонстрации работы библиотеки PatternLib нами был разработан веб сервис анализатора. При разработке online сервиса использовались фреймворк Struts [10], и некоторые функции JavaScript из библиотеки компонентов YUI (Yahoo User Interface) [11].

Работа online версии анализатора проходит следующим образом:

1) На главной странице вводится текст или загружается файл с указанием кодировки.

2) Выбираются шаблоны, которые будут использоваться при анализе текста (по умолчанию используются все) – при выборе названия шаблона из списка, справа отображается его содержимое

3) При нажатии кнопки «Анализировать» производится анализ заданного текста и выводится набор полученных триплетов и уникальных RDF ресурсов, являющихся субъектами или объектами.

4) Результат анализа можно просмотреть в виде RDF-графа. Для этого нужно нажать кнопку «Граф».

Используемые анализатором шаблоны хранятся в базе данных. На сегодняшний день не предусмотрен Веб интерфейс для добавления, удаления или редактирования шаблонов в базе данных, эта операция выполняется администратором системы в ручную (через средства администрирования СУБД). В дальнейшем планируется реализовать веб интерфейс для управления шаблонами анализатора.

7 Разработка, хранение и валидация шаблонов на корпусе текстов

Для поиска новых шаблонов от разработчика требуется вручную анализировать тексты. Это занимает очень много времени, поэтому было разработано Web-приложение для редактирования и проверки шаблонов. Предполагается, что данный программный продукт (валидатор шаблонов – рабочее название Vallyweb) [12] сможет упростить создание, анализ и проверку шаблонов, описанных на языке LSPL.

Валидатор предназначен для работы с текстами, которые хранятся в виде HTML документов в файловой системе сервера. На сегодняшний день для работы используются коллекции документов, используемые в дорожках семинара РОМИП’2009.

Ниже представлен алгоритм работы валидатора.

Работа с валидатором строится следующим образом:

На вход программы подается шаблон (выбирается из списка). Текст шаблона записывается в соответствующее окно и становится доступным для редактирования.

Так же на вход подается анализируемый корпус текстов (указывается адрес до папки с файлами). Для получения содержимого указанных выше документов используется парсер PatternX3M, реализованный на компонентах библиотеки Lucene [13].

Для анализа выбирается выбранное случайным образом некоторое число документов;

Во время парсинга текст разделяется на отдельные предложения, каждое из которых анализируется с помощью библиотеки PatternLib.

Результат выводиться в виде таблицы из двух колонок. В строчках таблицы отображаются все лексикализации шаблона и соответствующие наборы триплетов. Например:

Предложение	Соответствующая семантическая модель
Студент - это человек, который учится в университете	http://result/subject/Студент http://result/property/#subClassOf http://result/bject/человек

В результате, пользователь может оценить полученные лексикализации шаблона и откорректировать шаблон соответствующим образом. Текст шаблона можно исправить прямо в программе.

При создании и валидации шаблона исследователь может ввести коэффициент доверия шаблона, который отражает адекватность работы шаблона. Данный коэффициент можно рассматривать, как вероятность успешной работы шаблона, то есть отношение количества выходных семантических моделей, полученных при применении шаблона, реально соответствующих семантике предложения, к общему количеству лексикализаций шаблона.

На сегодняшний день интерфейс для вычисления коэффициента доверия шаблонов не реализован, его реализация планируется в дальнейшем.

Шаблоны, с которыми работает валидатор хранятся в виде XML файлов на сервере. В дальнейшем для хранения шаблонов планируется использовать ту же базу данных, в которой хранятся

шаблоны, используемые online анализатором. Это позволит организовать централизованное хранилище шаблонов и избавиться от дублирования данных.

Так же в будущем планируется предоставить доступ к валидатору всем желающим, для того чтобы с помощью сообщества исследователей создавать и проводить валидацию новых шаблонов более быстро и эффективно. Для этого потребуется реализовать механизм разграничения прав доступа и контроля версий.

8 Информационный поиск на основе семантики

Наиболее распространенными моделями информационного поиска по текстовым коллекциям документов являются:

1. Статистические методы
2. Методы поиска по семантическим сетям
3. Комбинированные методы

Кратко рассмотрим метод поиска на основе метрик TF-IDF.

Для коллекции документов строится свой, особый «алфавит», в который входят все (за исключением стоп-слов и словоформ, отличающихся от нормальных) встречающиеся в данных документах слова (термы).

Затем для каждого терма определяется частота встречаемости его в каждом документе. Таким образом, для каждого документа можно построить вектор частот Dm(t1,t2,…,tn), где t1 – частота встречаемости терма 1 в документе m, t2 – частота встречаемости терма 2 в документе m, и т.д. m – уникальный номер документа в коллекции, n – количество известных термов.

В итоге, в индексе (матрице из векторов частот отдельных документов) поисковой машины хранятся частотные вектора всех документов. При обработке запроса, сначала выбираются все термы, которые присутствуют в тексте запроса и строится соответствующий вектор Q(tx,ty,…tz), где tx, ty, tz – частоты входящих в запрос термов.

После построения вектора частот запроса, вектора частот документов дополняются нулями для тех термов, которые входят в запрос, но не входят в алфавит, а вектор частот запроса дополняется нулями для всех термов из алфавита, которые не входят в текст запроса.

Таким образом, все вектора приводятся к одной размерности. В конечном итоге, вычисляется условный косинус угла между векторами запроса и документов. Чем меньше данная величина, тем более релевантным считается документ.

Статистические методы, в настоящий момент, являются наиболее распространенными методами информационного поиска. Основной их особенностью является качественная математическая модель, позволяющая получать хорошие оценки релевантности для документов коллекции. Поисковые машины, основанные на данных методах, отличаются простотой интерфейса. Основным минусом данного метода является то факт, что не учитывается смысловая нагрузка текста документов коллекции и текста запроса.

Отсутствие учета смысловой нагрузки текстов (документов и запросов), зачастую приводит к нерелевантным результатам. Примерами поисковых машин такого типа являются популярные поисковые машины Google, Yandex, Rambler, Yahoo и т.д.

Основная идея второй группы методов информационного поиска заключается в том, что все исходные данные представлены в виде объектов семантических моделей, а поиск представляет собой навигацию по графу онтологии. Данные методы, в отличие от статистических, учитывают смысловую нагрузку информации, поскольку информация изначально представлена в виде онтологии или ассоциирована с ней посредством семантической разметки документов. Однако данные методы имеют ряд недостатков:

- Сложность пользовательского интерфейса, требующая от пользователя дополнительных затрат на конкретизацию объектов и свойств.
- Большинство информации в Интернет представлено в виде HTML-страниц и не содержат семантического описания контента. А ручная семантическая разметка документов представляет собой огромный объем работы.

В качестве примера подобной системы можно рассматривать систему АСНИ (Автоматизированная система научных исследований) [14] или проект SHOE [15].

К третьей группе методов информационного поиска относятся методы, которые помимо статистических методов поиска используют методы семантического анализа текстов. Данная группа методов развивается в настоящее время наиболее интенсивно. Основным плюсом систем комбинированного типа является комбинация качественной статистической модели поиска и учета семантических конструкций.

Основные минусы подобных систем, существующих в настоящее время:

- Большое время отклика
- Мало где используются механизмы логического вывода
- Ограничения на структуру запроса (при использовании простого пользовательского интерфейса)
- Необходимость установки дополнительных параметров поиска (при использовании сложных пользовательских интерфейсов)
- Большинство систем подобного типа используют в качестве исходной информации стандартные тексты, проводя семантический анализ на конечном этапе задачи поиска, что приводит к медлительности данных систем.

В качестве примера такой системы можно рассматривать поисковую машину AskNet [16].

Несмотря на то, что третья группа методов наиболее полно отвечает требованиям. предъявляемым к системам информационного поиска на основе семантики, все системы данного типа имеют недостатки.

Была поставлена задача разработать метод поиска, который бы был основан на статистическом методе поиска, учитывал семантическую структуру текстов, а так же был лишен таких недостатков третьей группы методов, как большое время отклика, ограничения на структуру запроса, и сложный пользовательский интерфейс. Разработка получила название SEUS (search engine using semantics) [17]

Решение заключается в том, чтобы исходную информацию представить в виде семантической сети, и работать уже не с отдельными словами, а с элементами данной сети (RDF-триплетами). Для приведения исходных данных из текстов на естественном языке в семантическую сеть, предлагается использовать модель представления информации, основанную на автоматическом построении онтологии с использованием лексико-синтаксических шаблонов.

Суть данной модели заключается в следующем:

1. Из предложений исходного документа извлекаются триплеты, которые в совокупности составляют полную онтологию данного документа. Для этого используются 4 механизма получения триплетов:

- используя триплеты, заранее встроенные в HTML документы с помощью микроформатов, например RDF/A [18]
- используя лексико-синтаксические шаблоны
- используя логический вывод, реализованный на базе библиотеки для работы с онтологиями Jena [19]
- а также логический вывод, специально разработанный для информационного поиска
- предполагается, что также будут использоваться и RDF ресурсы, описанные в популярных онтологиях и словарях, таких VCard FOAF и т.д.

2. Полученные таким образом триплеты сохраняются в БД, в качестве Jena-моделей, имеющих уникальные идентификаторы и ссылки на документы, к которым они относятся.

3. После получения всевозможных триплетов, которые формируют онтологию документа, считаем, что содержимое данного документа – есть набор идентификаторов триплетов, каждый из которых будет считаться отдельным термом.

4. Таким образом, после проведенных преобразований, можно воспользоваться существующей моделью TF/IDF, для которой алфавит составят идентификаторы триплетов, входящих в документ.

Помимо преобразования документов к набору идентификаторов триплетов, такому же преобразованию должен быть подвергнут и запрос. То есть, используя механизм лексико-синтаксических шаблонов, можно получить набор триплетов запроса, для тех которые уже имеются в алфавите, указать соответствующие идентификаторы, а для тех, которых в алфавите нет – ввести отрицательную нумерацию идентификаторов, что позволит учитывать их при использовании методики TF/IDF.

Кроме того, поскольку результаты применения лексико-синтаксических шаблонов к тексту могут не всегда отражать реальную семантику текста, они обладают некоторыми экспериментально определенными коэффициентами доверия. Поэтому данный коэффициент доверия нужно учитывать при определении частот термов. Учтен он будет следующим образом: частота терма в документе и запросе будет умножена на соответствующий коэффициент доверия.

Таким образом, предложенный метод использует статистический метод поиска, учитывает смысловую нагрузку исходного текста, за счет того, что он представляется в виде набора RDF-триплетов.

Данный метод лишен недостатков систем, использующих комбинированные методы. Поскольку семантический анализ исходных текстов перенесен из последнего этапа задачи поиска в задачу представления исходной информации и проводится еще до этапа индексирования исходной коллекции документов, на конечном этапе задачи поиска существенно сокращается время отклика поисковой системы.

Если не удалось получить триплеты из текста запроса, система автоматически переключается на работу со стандартным статистическим методом.

Для того, чтобы использовался комбинированный метод поиска, необходимо из запроса получить хотя бы один триплет. Поскольку триплеты из запроса получаются с помощью механизма лексико-синтаксических шаблонов, то ограничения на запрос определяются лишь их распространенностью.

В качестве интерфейса поисковой машины, использующей данный метод поиска можно использовать обычную строку поиска.

9 SEUS на РОМИП 2009

На базе открытой библиотеки для полнотекстового поиска Lucene в рамках проекта SEUS нами был реализован программный комплекс, который применяет к тексту лексико-синтаксические шаблоны, то есть получает соответствующие триплеты, а также строит семантический индекс, соответствующий модели, описанной в предыдущей главе. На данный момент реализована стандартная модель поиска Lucene и интерфейс в виде Веб приложения со строкой запроса [20].

На данный момент, описанный ранее механизм поиска не дает желаемых результатов. Это обусловлено тем, что набор лексико-

синтаксических шаблонов сейчас достаточно мал (12 штук). Кроме того, в механизме предварительного анализа не реализовано использование триплетов, заложенных в исходные документы с помощью микроформатов. Механизм логического вывода, специфический для информационного поиска пока так же не реализован.

Участники проекта SEUS подали заявку на участие системы в семинаре РОМИП'2009. Однако приведенная модель поиска не использовалась при решении заданий семинара. Это связано с низким качеством полученных практических результатов. Вместо этого на семинар были представлены результаты, полученные на основе стандартной модели поиска библиотеки Lucene.

Имея таблицы релевантности для заданий семинара, полученные от экспертов, в дальнейшем планируется доработать все элементы системы SEUS согласно модели поиска с учетом семантики.

Заключение

В рамках проекта SEUS были разработаны интерпретатор языка LSPL, а также online анализатор и валидатор шаблонов на его базе. Предполагается, что это станет значительным шагом к получению объема шаблонов, необходимого для качественного семантического анализа текстов на русском языке.

Предложенная модель поиска с учетом семантики требует более качественного семантического анализа, который ожидается, будет в связи с появлением анализатора и валидатора шаблонов будет получен в ближайшем будущем.

Также во время подготовки к участию в семинаре РОМИП авторы осознали, что поиск с учетом семантики сам по себе является обширной задачей и не может рассматриваться, лишь как метод оценки работы лексико-синтаксических шаблонов.

Дальнейшая работа над проектом SEUS будет двигаться в следующих направлениях:

- доработка и открытие в общий доступ валидатора шаблонов с целью привлечения заинтересованных специалистов
- получение количества шаблонов достаточного для качественного семантического анализа
- разработка собственного механизма логического вывода, специально предназначенного для информационного поиска
- получение результатов поиска с учетом семантики приближенных к стандартным моделям TF/IDF.

Автор благодарит компанию «Диктум», ее руководителя В.В. Окатьева и разработчика морфологического словаря А.Коваленко за предоставление синтаксического анализатора DictaScope. А также д. ф.-м. н., профессора заведующего кафедры компьютерных систем и телекоммуникаций Пермского Государственного Университета М.А. Марценюка за обсуждение работы и предоставление материально технической базы для проведения исследований. А также студентов кафедры за предоставление практических наработок, на базе которых написана статья.

Литература

[1] Marti A. Hearst, Automatic Acquisition of Hyponyms from Large Text Corpora // Proceedings of the 14th conference on Computational linguistics - Volume 2, Pages: 539 - 545 , Nantes, France, Association for Computational Linguistics, Morristown, NJ, USA, 1992.

[2] Большакова Е.И., Васильева Н.Э., Морозов С.С. Лексико-синтаксические шаблоны для автоматического анализа научно-технических текстов // Десятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2006. Труды конференции в 3-х томах. М.: Физматлит, 2006. Т. 2. С.506-524

[3] Christopher Brewster, Fabio Ciravegna и Yorick Wilks, User Centred Ontology Learning for Knowledge Management // Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers, Pages: 203 - 207, Springer-Verlag London, UK, 2002.

[4] Ермаков А.Е., Плешко В.В., Митюнин В.А. RCO Pattern Extractor: компонент выделения особых объектов в тексте. // Информатизация и информационная безопасность правоохранительных органов: XII Международная научная конференция. Сборник трудов - Москва, 2003. - С. 312-317. (http://www.rco.ru/article.asp?ob_no=237)

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

[6] Домашняя страничка синтаксического анализатора DictaScope http://www.dictum.ru/?main=products&sub=dictascope

[7] Домашняя страничка проекта GraphViz http://www.graphviz.org/

[8] Домашняя страничка семинара РОМИП http://romip.ru/

[9] Online анализатор на базе лексико-синтаксических шаблонов http://seus.rabchevsky.name:8080/DemoServlet/

[10] Домашняя страница проекта Struts http://struts.apache.org/

[11] Домашняя страница библиотеки YUI (Yahoo User Interface) http://developer.yahoo.com/yui/

[12] Onlne валидатор шаблонов Vallyweb http://seus.rabchevsky.name:8080/VallyWeb/

[13] Домашняя страница библиотеки для полнотекстового поиска Lucene http://lucene.apache.org/
[14] ПОДСИСТЕМА УТОЧНЯЕМОГО ПОИСКА СЕМАНТИЧЕСКОЙ ИНФОРМАЦИИ В ФОРМЕ ГРАФОВЫХ МОДЕЛЕЙ АСНИ http://network-journal.mpei.ac.ru/cgi-bin/main.pl?l=ru&n=9&pa=12&ar=8
[15] Домашняя страница системы поиска по семантически размеченным документам http://www.cs.umd.edu/projects/plus/SHOE/index.html
[16] Система полнотекстового поиска AskNet http://info.asknet.ru/technology.htm
[17] Домашняя страница проекта SEUS http://seus.rabchevsky.name/
[18] Микроформат для внедрения RDF графов в HTML документы RDF/A http://www.w3.org/TR/xhtml-rdfa-primer/
[19] Домашняя страница библиотеки для работы с онтологиями Jena http://jena.sourceforge.net/
[20] Online демонстрация поисковой системы SEUS http://seus.rabchevsky.name:8080/SEUS/
[21] Обзор методов аннотирования в Semantic Web в работах Шеффилдского университета http://rabchevsky.name/sheffield

Automatic ontology construction based on lexical-sintactic patterns for information retrieval

Evgeny Rabchevsky
Perm State University
seus@rabchevsky.name

The problem of automatic construction of ontology on a basis of semantic analysis of natural language is under discussion. The use of lexical-syntactic patterns is suggested. Syntax and semantics of language of lexical-syntactic patterns is considered. Developed software is able to

- store the patterns and text's corpora in Russian language database;
- edit and validate patterns on the Russian language on the corpora of Russian language texts;
- make semantic analysis of texts' corpora on a basis of patterns.

The results of the method is suggested to evaluate as a quality of information retrieval. The definition of term in the information retrieval model based on TF/IDF is been changed to RDF triple. The results of the given model application to ROMIP seminar tasks are discussed.