УДК 004.622: 004.912

ПРЕОБРАЗОВАНИЕ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ В RDF-ГРАФ

Артамонова Е.В., Лештаев С.В.

ФГБУН «Институт систем информатики им. А.П. Ершова» Сибирского отделения Российской академии наук, Новосибирск, e-mail: Artamonova.Elena.V@gmail.com

Авторы проводят исследования на стыке модели описания данных RDF и построения семантических сетей. В статье описывается авторский метод преобразования текстов на естественном языке в RDF-граф, что позволит в дальнейшем обрабатывать информацию, содержащуюся в тексте, опираясь на возможности, предоставляемые RDF. Обработка текста осуществляется в несколько этапов посредством преобразования текста в семантическую сеть при помощи специализированных приложений, разработанных авторами для лексического и синтаксического анализов. Дерево зависимостей, полученное в результате обработки анализаторами (lexer и parser соответственно), в дальнейшем преобразуется в RDF-граф. Особое внимание уделено выбору системы для хранения полученного RDF-графа и его дальнейшего использования. В результате исследований выявлено преимущество специализированной системы RDF Polar над универсальными базами данных.

Ключевые слова: Polar, RDF, Semantic Web, семантические сети, формальные грамматики, Big data

CONVERSION OF NATURAL LANGUAGE TEXTS TO RDF-GRAPH

Artamonova E.V., Leshtaev S.V.

A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, e-mail: Artamonova.Elena.V@gmail.com

Authors conduct researches on a joint of model of the description of data of RDF and Semantic Web. The article describes the author's method of converting text on a natural language into RDF-graph, that allows processing the information contained in the text, based on the opportunities provided by RDF. The conversion is carried out in several stages through the special applications implemented by authors for lexical and syntactic analysis. Dependency Tree resulting from processing analyzers (lexer and parser, respectively), later is to be transformed into RDF-graph and loaded into the database. Particular attention is paid to the selection of the optimum system to store the resulting RDF-graph for its further use, and as a result of the research revealed the advantage of a specialized system of RDF Polar in comparison with universal databases. RDF Polar provides maximum search speed for downloaded data.

Keywords: Polar, RDF, syntactic analysis, parsing, formal grammar, Big data

В последнее время исследователями уделяется много внимания проблеме обработки текстов на естественных языках, в том числе извлечению из них информации и ее структурированию. На этом фоне представляется интересной задача преобразования неформатированного текста в некоторую структуру с использованием RDF-формата, с последующим сохранением триплетов в какой-либо системе. Решение этой задачи позволило бы использовать новые возможности для работы с данными в части их поиска и структуризации [10–14]. В частности, возможность загрузки текста в RDF-формат представляется интересной как часть решения задачи генерации «информационного портрета» на основе данных из различных источников данных [1].

В данной статье приводится описание решения этой задачи: в разделах 1–4 приводится описание способа обработки текстов на естественном языке, разделы 5 и 6 содержат описание исследований, выполненных авторами и обосновывающих использование в качестве эффективной системы хранения триплетов системы RDF Polar [1–7].

Общая схема метода преобразования текста в формат RDF

Задачей проводимого исследования является создание механизма преобразования текста на естественном языке в семантическую сеть и ее представление в виде RDF-графа и загрузкой этого графа в некоторую базу данных для дальнейшего хранения и обработки.

В рамках предварительной подготовки были изучены существующие методы анализа текстов на естественном языке, и за основу был взят метод анализа русской грамматики, предложенный академиком В.В. Виноградовым [2]. Опираясь на его работы, была разработана упрощенная схема анализа текста на русском языке. В рамках этого упрощения предполагается, что наша схема анализа может быть описана некоторой формальной суффиксной грамматикой, сформулированной Н. Хомским в его работе «Синтаксические структуры» [10]. Если какие-либо фрагменты текста (например, словосочетания и сложные синтаксические обороты) не могут быть преобразованы с использованием данной схемы, то они будут пропущены. Последующие этапы ис-

следования позволят постепенно отказаться от этих упрощений.

Предлагаемый метод построен на использовании морфологической парадигмы, то есть предполагает существование для каждого слова таблицы форм, содержащей инвариантную часть слова и его специальные части (форманты). Предполагается также, что перечень грамматических значений ограничен [2].

В рамках метода предлагается сначала преобразовать текст в дерево зависимостей с помощью лексического и синтаксического анализа. Оба вида анализа выполняются посредством приложений, разработанных авторами для выполнения лексического (lexer) и синтаксического (parser) анализов, о чем рассказано в разделах 2–4.

После аналитической обработки необходимо преобразовать получившийся граф зависимостей в RDF-граф и сохранить последний в некую базу данных. При этом желательно использовать базу данных, которая будет обеспечивать эффективную скорость индексированного поиска данных и их обработки. Проведены исследования времени поиска данных, загруженных в различные базы данных – MySQL, MS SQL и RDF Polar. Сравнительные характеристики, полученные в ходе этих исследований, представлены в разделах 5 и 6 и позволяют сделать вывод о целесообразности использования для работы с текстом именно RDF Polar.

Формальная грамматика исследуемого текста на естественном языке

В качестве объекта для лексического и синтаксического анализа текста на естественном (русском) языке предполагается использовать произвольный набор предложений на этом языке.

Формализовать элементы естественного (русского) языка исследуемого текста предлагается с помощью контекстно-зависимой грамматики [9], определенной следующим образом:

● Алфавит языка X = {А, а, Б, б, ... , Я, я, 0, 1,... 9, ., ,, :, ;, -, !, ?, ", ", (,)} представлен множеством буквенных символов, цифр и знаков препинания.

● Множество терминальных символов *T* представлено грамматическим словарем А.А. Зализняка [4], содержащим около 100 тыс. слов русского языка и подходящим для проверки существования и определения начальной формы каждого слова, встречающегося в тексте.

Множество нетерминальных символов *N* = {предложение, подлежащее, сказуемое, глагол, числительное, местоимение и т.д.}, где исходный символ n_0 = «предложение» является набором частей речи и членов предложения русского языка.

В силу специфики используемого словаря А.А. Зализняка, для лексического анализа мы будем использовать суффиксную синтетическую грамматику [9]. Соответственно, одной и той же лексемой будут являться как само слово, так и его суффиксная производная, где суффикс (или несколько суффиксов) является изменяемым окончанием токена. Например, слова (токены) «причинный» и «беспричинный» являются различными лексемами, т.к. их отличает только приставочная часть, не используемая в суффиксной грамматике. При этом слова «беспричинный» и «беспричинного» соответствуют одной и той же лексеме, т.к. их различает только суффикс.

Морфемы, не включенные в используемый словарь, а также лексические омонимы и устойчивые словосочетания на текущем этапе исследования опускаются и не рассматриваются, но авторы допускают их дальнейшее включение в текущий словарь – например, посредством ручного ввода при наличии для каждого добавляемого термина экспертной оценки.

Для обработки текста используется специальное приложение, состоящее из двух блоков. В блоке обучения реализована возможность задавать продукции, вводимые экспертом вручную. Второй блок позволяет, используя словарь А.А. Зализняка, определять начальную форму слова и их грамматические свойства, используя продукции.

Лексический анализ текста

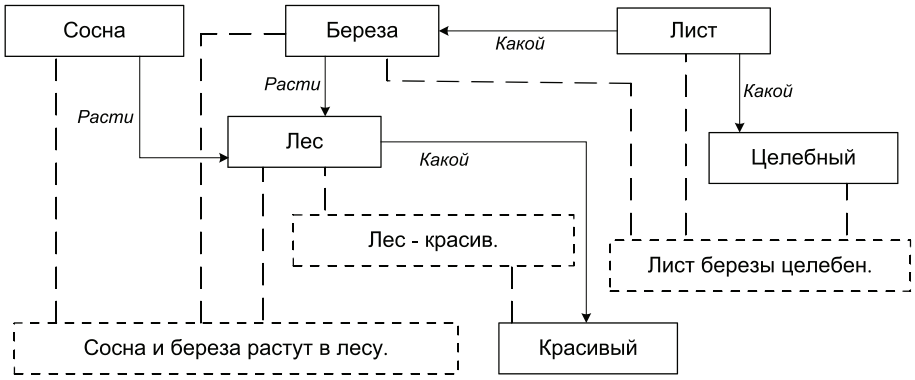
Лексический анализ осуществляется над некоторым текстом на русском языке по произвольно выбранной теме. Предполагается, что исследуемый текст можно описать с помощью формальной грамматики, предложенной авторами.

Лексический анализ текста проводился в несколько этапов:

1. «Выделение законченных предложений». Весь текст разбивается на фразы, которые ограничиваются знаками препинания (., !, ?, ...), включенными в алфавит.

2. «Выделение слов». Анализируем каждую фразу в отдельности. На текущем этапе при помощи символов алфавита (,, ;, -;) разбиваем фразу на набор токенов.

3. При помощи специализированного приложения «Лексический анализатор» и заранее введенного экспертом набора продукций преобразуем произвольный набор слов во множество агрегированных объектов, содержащих входное слово, его начальную форму, морфологические и грамматические свойства, в качестве вывода этих продукций.



Пример семантической сети

Таблица 1

Пример таблицы данных RDF-графа, созданного на основе семантической сети

Объект (ID)	Предикат	Данные
1	Имя	Лес
2	Имя	Сосна
3	Имя	Береза
4	Имя	Лист
5	Имя	Красивый
6	Имя	Целебный

Таблица 2

Пример таблицы триплетов RDF-графа, созданного на основе семантической сети

Объект	Предикат	Субъект
1	Какой	5
2	Расти	1
3	Расти	1
4	Какой	6
4	Какой	3

Таким образом, существует набор правил преобразования, позволяющий перевести слово из его текущей формы в словарную форму, то есть определить для каждого токена лексему. В результате на выходе формируется набор лексем и данных морфологического и грамматического разбора для каждой лексемы, подающихся далее на вход синтаксического анализатора.

Данная реализация лексического анализатора будет работать корректно только для языков, для которых составлен грамматический словарь, аналогичный использованному авторами словарю А.А. Зализняка, а также имеющих свой набор правил для перевода произвольно выбранного токена в его словарную форму.

Синтаксический анализ текста

Одной из важных задач, решаемых в рамках настоящего исследования, является синтаксический анализ текста на естественном языке. Анализ осуществляется посредством специально разработанного приложения (parser).

Как сказано в предыдущем разделе, на вход синтаксического анализатора подается набор лексем и данных морфологического и грамматического разбора, полученных с помощью лексического анализатора.

Далее, основываясь на упрощенной схеме анализа русского языка, синтаксический анализатор выполняет синтаксический разбор фразы. Например, правило, гласящее, что подлежащее в предложении – это существительное либо местоимение в именительном падеже, позволяет сформулировать продукцию, использующую на входе вышеописанные агрегированные объекты и выдающую на выходе подлежащее для каждой фразы.

Тогда схему, предложенную авторами, можно записать следующим набором продукций формальной грамматики:

P = {p1: предложение → (группа подлежащего), (группа сказуемого),

p2: группа подлежащего → (определение) (подлежащее) (дополнение),

p3: группа сказуемого → (сказуемое) (обстоятельство)}.

Используя эту схему, синтаксическую структуру предложения можно представить в виде дерева зависимостей (дерева разбора или синтаксического дерева). Такое дерево представляет собой линейно-упорядоченное множество токенов (узлов), где каждая пара узлов соединена подчинительной связью, причем направление этой связи однозначно определено.

Далее, при построении дерева зависимостей каждое формально выделенное

вхождение токена рассматривается, как отдельный элемент, и все связи являются подчинительными. Традиционно эти особенности трактуются как недостатки представления синтаксических структур в виде деревьев зависимостей, но в нашем случае они являются скорее достоинствами, поскольку демонстрируют, что структура таких деревьев родственна структуре RDF-графа, что позволяет на следующем этапе однозначно преобразовать одно в другое.

Выбранный подход существенно упрощает трансформацию дерева зависимостей в предложенную авторами модель семантической сети.

Таким образом, после трансформации текст на естественном языке представляется в виде семантической сети, которая, в свою очередь, может быть представлена в виде RDF-графа. Соответственно, дерево зависимостей можно представить в терминах RDF, где в узлах находятся субъекты или данные, а ребра будут представлять собой предикатные отношения между соответствующими субъектом и объектом или субъектом данных. Например, значительная часть объектов и субъектов, включенных в граф, может быть представлена именами существительными и прилагательными, а в качестве предикатов могут выступить стандартный набор вопросов, задаваемых от главного слова к зависимому (напр. «Какой?», «Как?» и т.д.), а также все слова, содержащиеся в словаре и определенные parser как глаголы (напр., «Делать», «Расти» и т.д.).

Здесь следует сразу отметить, что имена существительные, которые в нашем графе будут представлять собой субъекты, важно проверять на совпадение, чтобы избежать дублирования субъектов.

Интересной, по мнению авторов, проблемой, заслуживающей подробного рассмотрения в ходе дальнейших исследований, является идентификация персоналий. Она включает в себя правила распознавания имен собственных, не включенных в общий словарь, и правила определения дубликатов (для слов вида «Ваня» / «Ванюша»). Для этого предполагается использовать словарь Про-Линг, содержащий, в отличие от словаря Зализняка, помимо имен нарицательных еще и имена собственные [8]. В дальнейшем предполагается разработать дополнительный самообучающийся блок, который будет автоматически отыскивать имена собственные и добавлять их в специализированный словарь имен.

Специфика системы Polar RDF

Лексический и синтаксический анализ формирует дерево зависимостей, пригодное для преобразования в RDF-граф.

Использование RDF-модели представляется авторам перспективным, поскольку опирается на широкий спектр современных исследований и открывает новые перспективы использования данных. Для хранения RDF-триплетов, после ряда экспериментов, описанных в разделе 6, было принято решение использовать RDF Polar.

RDF Polar является оригинальной разработкой, представленной сотрудниками ИСИ СО РАН и описанной циклом статей [1–7]. Система RDF Polar предоставляет широкие возможности для работы с информацией большого объема, включая большие данные (big data). Она обеспечивает быстрое сохранение большого объема данных, а также контекстный поиск триплетов по условию. Это дало авторам возможность представить любой неформатированный текст в виде структурированных данных посредством создания RDF-триплетов.

Изначально БД может быть размещена на локальном сервере, а по мере разрастания можно переместить ее в облачные вычисления [8].

RDF-хранилище на основе Polar состоит из нескольких структур: множество триплетов, в которых каждый IRI заменён натуральным числом (кодом), множество пар (IRI, код) для кодирования/декодирования, индексы для поиска по условию, когда заданы не все составляющие триплетов. Каждый индекс может находиться в одном из двух режимов:

1. Полностью загруженные в оперативную память для быстрого поиска.

2. Размещенные на жёстком диске в виде файла для больших объёмов данных.

Таблица 3

Сравнительные характеристики реляционных и Polar хранилищ триплетов

Число триплетов в данных	MS SQL	MySQL	Polar
100 000	2,0 мс	0,3 мс	0,2 мс
1 000 000	2,6 мс	0,5 мс	0,3 мс
10 000 000	2,8 мс	2,0 мс	0,3 мс
100 000 000	Не удалось загрузить	Время загрузки более 10 часов	2,96 мс

RDF-хранилище является .NET библиотекой в решении, не использует сторонних индексных решений, не требует установки и настройки.

Polar является открытой системой, что дает возможность использовать триплеты данных в дальнейших разработках.

Сравнительный анализ времени поиска данных в системе Polar RDF и универсальных реляционных СУБД

В качестве оценки эффективности работы Polar, как RDF-хранилища, проведен следующий эксперимент. Была реализована RDF-модель описания данных для Polar и для популярных реляционных баз данных MS SQL и MySQL. Модель представляла собой коллекцию триплетов, хранящихся в одной таблице (Id, *s*, *p*, *o*), где Id – целочисленный уникальный идентификатор триплета, *s* – субъект триплета, заданный строкой, *p* – строка предиката, *o* – строка объекта. Для таблицы создан индекс по столбцу *s*. Для эксперимента были сгенерированы RDF данные: *k* субъектов: $s_1, s_2, \ldots s_k$ – строки. Для каждого s_i создано по 10 триплетов:

$$(s_i, p_1, o_1), (s_i, p_2, o_2), \ldots, (s_i, p_{10}, o_{10}),$$

где p_j, o_j – тоже строки, *i* от 1 до *k*, а *J* от 1 до 10. В качестве параметра оценки эффективности работы системы авторы выбрали время поиска всех 10 триплетов для заранее случайно выбранного s_i объекта, а для оценки зависимости производительности системы от количества записей в исследуемой таблице авторы производили измерения для разного числа триплетов.

Важно заметить, что, в отличие от сравниваемых реляционных баз данных, Polar не использует так называемую «ленивую инициализацию» (lazy initialization), и для получения повторяемых результатов оказалось необходимо проводить две дополнительные процедуры: перед поиском сервер необходимо перезапустить, что обеспечивает полную очистку системного кеша, организовать «разогрев» системы, при котором части файла данных Polar будут помещены в системный кэш. Результаты эксперимента приведены в табл. 3. В ней указано время в миллисекундах поиска триплетов, содержащих случайно выбранный субъект.

Столбцы MS SQL и MySQL соответствуют реляционным решениям с Microsoft SQL Express 12 (2016) x64 и MySQL 5.7.10 Community Server x64 соответственно.

Табл. 3 демонстрирует явное преимущество RDF Polar над реляционными хранилищами, что позволило авторам отдать ему предпочтение при разработке синтаксического анализатора.

Заключение

В статье приведено описание метода преобразования фрагментов неформатированного текста на естественном языке в семантическую сеть с дальнейшей ее трансформацией в RDF-графы и последующим их сохранением в базу данных. В дальнейшем метод предполагается использовать для упрощения поиска информации и ее структуризации.

Построение RDF-графа становится возможным лишь после обработки входящего текста лексическим и синтаксическим анализатором.

Учитывая аспекты, приведенные в разделе 6, авторы сочли целесообразным использовать Polar RDF для загрузки и обработки RDF-графов, полученных из неформатированных фрагментов текста.

Список литературы

1. Артамонова Е.В. Современные проблемы персонификации и экстракции данных // Проблемы информатики. – 2015. – № 2. – С. 44–58; URL: http://problem-info.sscc.ru/?q=node/125 (дата обращения: 09.07.2015).

2. Виноградов В.В. Русский язык. Грамматическое учение о слове. – М., Высшая школа, 1972. – 616 с.

3. Зализняк А.А. Из заметок о любительской лингвистике. – М.: Русскій Миръ, 2010. – 240 с. (Серия: Литературная премия Александра Солженицына).

4. Зализняк А.А. Грамматический словарь русского языка. Словоизменение. – М., 1977. Изд. 2-е, испр. и доп. – М.: Русский язык, 1980. Изд. 3-е. – М.: Русский язык, 1987. Изд. 4-е, испр. и доп. – М.: Русские словари, 2003. Изд. 5-е, испр. – М.: Аст-пресс, 2008.

5. Марчук А.Г., Лештаев С.В. Экспериментальная реализация Sparql-1.1 и RDF Triple Store // Аналитика и управление данными в областях с интенсивным использованием данных, XVII Международная конференция DAMDID/RCDL'2015, Обнинск, 13–16 октября 2015 года, Труды конференции, С. 83–87.

6. Марчук А.Г. PolarDB – система создания специализированных NoSQL баз данных и СУБД // Моделирование и анализ информационных систем. – 2014. – Т. 21, № 6. – С. 169–175.

7. Марчук А.Г., Марчук П.А. Платформа реализации электронных архивов данных и документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XIV Всероссийской научной конференции RCDL'2012. Переславль-Залесский, Россия, 15–18 октября 2012 г. – г. Переславль-Залесский: изд-во «Университет города Переславля», 2012. – С. 332–338.

8. Словарь Про-Линг [Электронный ресурс]. – Режим доступа: http://mytts.forum2x2.ru/t142-topic (дата обращения: 25.04.2016).

9. Платонов Ю.Г., Артамонова Е.В. Метод Business Community и «облачные» вычисления (Cloud computing) // Фундаментальные исследования. – 2013. – № 4–5. – С. 1089–1093; URL: http://www.rae.ru/fs/?section=content&op=show_article&article_id=10000577 (дата обращения: 02.04.2015).

10. Хомский Н. Синтаксические структуры// Новое в лингвистике. – М., 1962. – Вып. II. – С. 412–527.

11. Bizer Chris, Cyganiak Richard. D2R Server – Publishing Relational Databases on the Semantic Web. Poster at the 5th International Semantic Web Conference (ISWC2006), 2006.

12. Sanders Alton. Grammar, Theories // Encyclopaedia of Linguistics / Philipp Strazny, editor. – New York, Oxon: Fitzroy Dearborn, 2005. – P. 397–401.

13. The Linking Open Data cloud diagram Официальный сайт [Электронный ресурс]. – Режим доступа: http:// http://lod-cloud.net// (дата обращения: 12.04.2015).

14. Volz Julius, Bizer Christian, Gaedke Martin, Kobilarov Georgi. Silk – A Link Discovery Framework for the Web of Data [Электронный ресурс]. – Режим доступа: http://events.linkeddata.org/ldow2009/papers/ldow2009 _paper13.pdf (дата обращения: 02.04.2015).