УДК 004.825
DOI 10.25205/1818-7900-2017-15-3-79-89

Д. Е. Пальчунов [1, 2], **А. А. Финк** [1]

[1] *Новосибирский государственный университет*
ул. Пирогова, 1, Новосибирск, 630090, Россия

[2] *Институт математики им. С. Л. Соболева СО РАН*
пр. Академика Коптюга, 4, Новосибирск, 630090, Россия

palch@math.nsc.ru, a.fink@g.nsu.ru

РАЗРАБОТКА АВТОМАТИЗИРОВАННЫХ МЕТОДОВ ПОРОЖДЕНИЯ СЛУЖЕБНЫХ ДОКУМЕНТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Статья посвящена проблеме автоматизированного порождения служебных документов на естественном языке с логическим контролем их правильности. Рассмотрены существующие подходы к автоматизации порождения документов. Предложен метод автоматизированного порождения логически правильных документов на естественном языке, основанный на применении параметрических шаблонов.

Ключевые слова: извлечение знаний, представление знаний, нормативные документы, служебные документы, параметрические шаблоны.

Введение

Статья посвящена проблеме автоматизированного порождения служебных документов на естественном языке с логическим контролем их правильности.

В настоящее время нормативно-правовой блок быстро развивается. Постоянно появляются новые документы и обновляются старые. Так, например, в базу данных consultant.ru [1] новые нормативные документы поступают несколько раз в день. В таком быстрорастущем потоке информации сложно ориентироваться, а тем более отслеживать изменения в нормативных документов.

В разных ситуациях необходимо создавать (заполнять) различные служебные документы – заявления, служебные записки, отзывы, рецензии и т. д. При создании служебных документов необходимо использовать различные нормативные документы, которые во многих областях деятельности, особенно в сфере образования, постоянно меняются. В частности, издаются новые образовательные стандарты с измененными списками компетенций, меняются требования к ВКР бакалавров и магистров, вводятся новые профессиональные стандарты и т. д. Поэтому документ, который на настоящий момент является правильно составленным, через некоторое время может оказаться не соответствующим текущим правилам и нормативным документам. Для решения этой проблемы необходимо разработать методы автоматизированного порождения правильных документов на естественном языке с логическим контролем их правильности.

[1] Официальный сайт компании «КонсультантПлюс». URL: http://www.consultant.ru/law/hotdocs/

Пальчунов Д. Е., *Финк А. А.* Разработка автоматизированных методов порождения служебных документов на естественном языке // Вестн. НГУ. Серия: Информационные технологии. 2017. Т. 15, № 3. С. 79–89.

При автоматизации документооборота часто прибегают к помощи такой системы, как «1С» [2]. Она помогает создавать удобные формы автоматической генерации документов по данным, полученным от пользователя. Система является достаточно гибкой за счет того, что она может использовать неограниченное количество переменных значений, вводимых пользователем. Данная система имеет и свои недостатки. При изменении правил заполнения документов приходится вносить изменения в программу, для этого необходимо привлекать специалистов. Это не очень удобно, особенно в ситуациях, когда правила заполнения (создания) документов часто меняются. Решение задачи разработки методов автоматизированного порождения правильных документов на естественном языке с логическим контролем их правильности может решить эту проблему.

В существующих программных системах, таких, например, как «1С», документ рассматривается, как шаблон с множеством переменных, где в качестве переменных выступают значения, введенные пользователем. В разрабатываемой системе предлагается рассматривать документы как параметрические шаблоны [1]. В качестве переменных также выступают значения, введенные пользователем. Однако теперь у нас к переменным добавляются параметры. Автоматизация заполнения этих параметров помогает решить поставленную задачу.

Рассмотрим в качестве примера «Отзыв научного руководителя на выпускную квалификационную работу бакалавра». В данном документе в качестве параметров могут выступать компетенции, название итоговой работы (выпускная квалификационная работа бакалавра или магистра), название комиссии (ГЭК или ГАК) и т. д. Правильность значений этих параметров для данного создаваемого служебного документа определяется текущими нормативными документами. Таким образом, с каждым служебным документом (более точно, с каждым видом служебных документов) можно сопоставить соответствующий ему параметрический шаблон. Преимуществом использования параметрических шаблонов является то, что если такой шаблон составлен правильно, то его не нужно менять при изменениях нормативных документов и онтологии предметной области (например, предметной области высшего образования), не носящих принципиального характера.

Имея параметрический шаблон служебного документа и значения всех его параметров можно автоматически порождать данный документ. При этом в автоматизированном режиме осуществляется логический контроль правильности заполнения значений параметров и переменных.

Существующие решения

В настоящее время существует большое количество систем для автоматизации создания документов. Рассмотрим такие решения, как АвтоДок, Microsoft Word и «1С».

АвтоДок

Данный продукт создан компанией «Элевайз» [3]. Для автоматизации создания документов в системе используются шаблоны, которые представляют собой файлы форматов RTF, DOC (для текстовых документов) или XLS (для электронных таблиц). Внешне эти файлы очень похожи на обычные документы, однако в них присутствуют специальные наборы символов, вместо которых при генерации конечного файла будет вставляться нужная информация. Создание таких документов происходит достаточно просто. После установки программы АвтоДок в Word появляется специальная панель для добавления переменных. Далее пользователь может взять уже существующий документ и заменить в нем всю изменяемую информацию на переменные. Обработав таким образом весь файл, пользователь получает готовый шаблон.

В шаблон можно вставлять не только переменные, но и функции, которые могут включать в себя стандартные математические, строковые и временные операции. Наличие циклов и условных операторов делает систему еще более гибкой. Кроме возможности автоматиза-

[2] Официальный сайт «1С:Предприятие». URL: http://v8.1c.ru/http://compress.ru/article.aspx?id=12230
[3] Обзор «АвтоДок». URL: http://www.ixbt.com/soft/autodoc-2.shtml

ции создания документов АвтоДок также поддерживает хранение и учет этих документов. Другими словами, все сгенерированные документы хранятся в системе.

Microsoft Word

В этой системе, как и предыдущей, для автоматизации создания документов используются шаблоны [4]. Шаблоны хранятся в файлах формата DOT (DOTX) или XLT (XLTX). Редактирование шаблонов возможно с помощью пользовательского интерфейса Word и с помощью Visual Basic .NET [5]. Второй способ предпочтительнее, так как является более гибким и предоставляет больше возможностей. Система предоставляет большое количество элементов управления содержимым, такие как «форматированный текст», «раскрывающийся список» или «выбор даты». По сути, все они являются переменными шаблона. Чтобы сделать шаблон более удобным в использовании, также можно добавлять пояснительный текст.

«1С»

Данная система предназначалась не для автоматизации создания документов, но это одна из ее функций. Для начала форма создается специалистом с помощью редактора форм [6]. Форма может состоять из полей для ввода текста, выпадающих списков, радиокнопок, чекбоксов, полей для ввода даты и т. д. Из данных, введенных в эти поля конечным пользователем, формируется документ. В качестве шаблонов текста подходят обычные форматы Microsoft DOT и XLT. Наличие в системе своего языка программирования делает ее очень гибкой. По сути, данная система объединяет возможности двух предыдущих систем.

Методы извлечения и формального представления знаний: проект DBpedia

Для автоматизированного порождения правильных служебных документов необходимо использовать актуальную информацию, содержащуюся в них. Для этого необходимо реализовать автоматизированное извлечение знаний из текстов естественного языка и их формальное представление.

Одним из перспективных проектов в области извлечения и представления знаний из текстов естественного языка является проект DBpedia [7]. DBpedia содержит огромную базу знаний, извлеченных преимущественно со страниц Wikipedia. Знания представляются в формате RDF [8], что позволяет обращаться к ним с помощью языка SPARQL (SPARQL Protocol and RDF Query Language) [9]. Таким образом DBpedia, объединив знания, извлеченные с разных страниц Википедии, позволяет пользователям находить ответы на вопросы в ситуациях, когда требуемая информация разбросана по страницам Wikipedia.

Кроме знаний, представленных в формате RDF, DBpedia также имеет сравнительно небольшую онтологию, представленную на языке OWL [10]. Она создана вручную и состоит из наиболее часто используемых понятий Wikipedia. В настоящее время эта онтология состоит из 685 классов, 2 795 свойств и 4 233 000 экземпляров. Каждый класс имеет несколько имен на разных языках. Классы связаны между собой только отношениями «класс – подкласс». DBpedia использует онтологию для структуризации данных. Например, Владимир Путин считается «экземпляром» класса *President. President* – подкласс класса *Politician*.

[4] Сохранение документа Word в виде шаблона. URL: https://support.office.com/ru-ru/article/Сохранение-документа-Word-в-виде-шаблона-cb17846d-ecec-49d4-82ea-a6f5e3e8b9ae

[5] Автоматизация приложений Microsoft Office. URL: http://compress.ru/article.aspx?id=12230

[6] Редактор формы. URL: http://v8.1c.ru/overview/Term_000000085.htm

[7] DBpedia. 2016. URL: http://wiki.dbpedia.org/

[8] Resource Description Framework (RDF). 2014. URL: https://www.w3.org/RDF/

[9] SPARQL Query Language for RDF. 2013. URL: https://www.w3.org/TR/rdf-sparql-query/

[10] Ontology. 2017. URL: http://wiki.dbpedia.org/services-resources/ontology

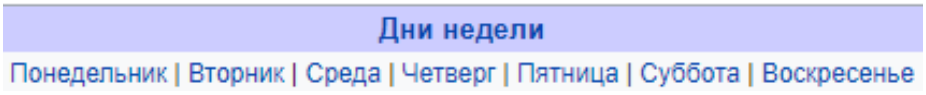

Рис. 1. Пример элемента для навигационного шаблона

Рис. 2. Пример инфобокса

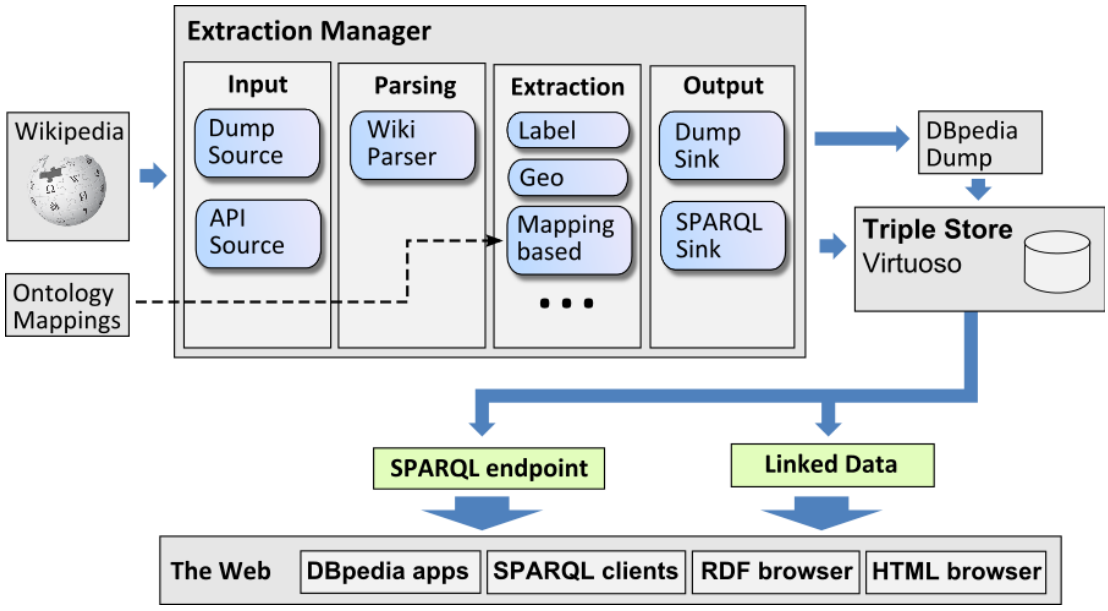

Рис. 3. Алгоритм извлечения знаний со страниц Wikipedia в рамках проекта DBpedia

Politician – подкласс класса *Person*. Такая структуризация очень удобна, потому что, если мы захотим получить список всех политиков, мы легко это сделаем, запросив все экземпляры класса *Politician* и экземпляры его подклассов.

Как было сказано, DBpedia содержит базу знаний, извлеченных со страниц Wikipedia. Извлечение знаний происходит в несколько этапов. Сначала загружаются страницы Wikipedia. Каждая страница разбирается вики-парсером, который извлекает шаблоны, содержащиеся в вики-тексте. Шаблоны – это самостоятельные элементы (тексты, рамки, изображения, таб-

лицы), встраиваемые в вики-страницу. Виды шаблонов: общие, служебные, навигационные, тематические, карточки, юзербоксы (специальный шаблон, размещающийся на личной странице участника и содержащий информацию о нем) и др. Например, навигационному шаблону *{{Неделя}}* соответствует элемент, представленный на рис. 1.

Шаблоны отправляются на обработку в экстракторы. Экстрактор – это программа по извлечению знаний из шаблона, каждый из них направлен на извлечение информации определенного вида. Например, инфобоксы [2] (рис. 2) извлекают заголовки, тезисы, географические координаты и т. п. Поскольку Wikipedia содержит большое количество шаблонов, необходимо большое количество экстрактеров, чтобы уметь обрабатывать любой вид информации. Таким образом, для каждого шаблона существует свой алгоритм отображения данных в формат RDF. Все извлеченные RDF-триплеты помещаются в хранилище. На рис. 3 представлены этапы работы алгоритма извлечения знаний со страниц Wikipedia [3].

Как было сказано, DBpedia использует язык SPARQL для доступа к знаниям. С помощью него можно получить только такие ответы на вопросы, которые явно прописаны в базе знаний [4]. В частности, нельзя получить информацию, которую можно было бы породить при помощи ризонеров – автоматизированных систем логического вывода. Для работы машины логического вывода необходимо перевести формализованные знания на язык логик описаний (DL) или в формат OWL DL [11]. Непосредственная реализация этого затруднена, так как DBpedia хранит все знания в формате RDF.

Методы автоматизированного порождения служебных документов на естественном языке

Для решения проблемы автоматизированного порождения правильных служебных документов на естественном языке нам необходимо решить следующие задачи:

- пополнение онтологической модели знаниями из нормативных документов;
- контроль актуальности нормативных документов;
- проверка онтологической модели на непротиворечивость;
- контроль приоритетов нормативных документов;
- порождение логически правильных документов на основе онтологической модели.

Пополнение онтологической модели

Для пополнения онтологической модели мы используем алгоритм, подобный алгоритму системы DBpedia. В большинстве случаев при обновлении документов их структура не меняется. Например, в одной версии документа утверждается, что студент защищает дипломную работу, а в следующей версии утверждается, что студент защищает выпускную квалификационную работу. В итоге изменилось наименование работы, которую защищает студент, но структура документа осталась прежней. Исходя их этого мы можем задавать шаблоны, которые позволят нам извлекать из нормативных документов наименование того, что защищает студент, наименование комиссии, которая осуществляет защиту (ГЭК или ГАК) и пр.

Для поиска шаблонов в текстах нам необходимо производить синтаксический и семантический анализ текстов естественного языка. Для этого в работе применяются синтаксический и семантический парсеры АОТ; более детально они будут описаны ниже. Эти парсеры обладают хорошей производительностью. Кроме того, выходные данные синтаксического парсера АОТ представлены в формате XML, что очень удобно для дальнейшего разбора. Синтаксический парсер АОТ хорошо совместим с семантическим парсером АОТ. В итоге информация, извлеченная из нормативных документов, добавляется к онтологической модели [5–7].

[11] OWL 2 Web Ontology Language Document Overview (Second Edition) // W3C OWL Working Group. URL: http://www.w3.org/TR/owl2-overview

Контроль актуальности нормативных документов

Обычно новые версии нормативных документов выкладывают в общий доступ на одних и тех же сайтах. Поэтому для контроля актуальности информации необходим постоянный мониторинг этих сайтов. С помощью шаблонов из сайтов извлекаются ссылки на необходимые документы. Далее документы скачиваются и сохраняются в базу данных вместе со ссылкой. Сохранение ссылки необходимо, чтобы в случае с большим количеством источников определить тот, с которого был загружен документ. Включая новый документ в базу данных, мы анализируем название документа, дату его принятия (утверждения), название организации или должностное лицо, утвердившее этот документ и т. д. Таким образом, мы имеем в своем распоряжении самые актуальные версии нормативных документов.

Проверка на непротиворечивость

Для проверки непротиворечивости запускается машина логического вывода Hermit [8]. Если она выведет, что знания находятся в несогласованном состоянии или что существует хотя бы один класс, эквивалентный owl:Nothing, то можно заключить, что найдено противоречие. Для подробного анализа и объяснения выведенного противоречия используется утилита OwlExplanation.

Контроль приоритетов нормативных документов

Различные нормативные документы имеют разные уровни приоритета. Например, постановление правительства или министерства имеет бóльшую степень приоритета, чем приказ ректора вуза. Это важно учитывать при обнаружении и разрешении противоречий в нормативных документах. Изданное постановление министерства автоматически отменяет все противоречащие ему положения, содержащиеся в документах, утвержденных нижестоящими организациями.

Для контроля приоритетов нормативных документов используется представленная на языке OWL DL онтология организаций, относящихся к рассматриваемой предметной области. В этой онтологии организации представлены в виде классов языка OWL, связанных между собой транзитивным свойством «подчиняться». Например, класс «Новосибирский государственный университет» обладает свойством «подчиняться» классу «Министерство образования». При обнаружении противоречия между двумя и более документами мы можем определить, какой из документов имеет более низкий приоритет и, таким образом, разрешить противоречие.

Порождение логически правильных документов

Будем представлять создаваемые нормативные документы как параметрические шаблоны. Зададим сигнатуру, состоящую из множества констант, множества параметров и множества переменных. Текст, который мы считаем неизменным, мы относим к множеству констант C. Части текста, которые зависят от существующих правил и нормативных документов, мы относим к множеству параметров Q. Части текста, которые должен ввести (заполнить) пользователь, мы относим к множеству переменных X. Структура шаблона, так же как и множество констант, задается вручную пользователем. Следовательно, при возникновении необходимости описать новый документ, это можно сделать, не прибегая к помощи программиста.

В качестве примера рассмотрим четыре документа: «Отзыв научного руководителя на выпускную квалификационную работу бакалавра», «Отзыв научного руководителя на выпускную квалификационную работу магистранта», «Рецензия на выпускную квалификационную работу бакалавра», «Рецензия на выпускную квалификационную работу магистранта». Для каждого из них специалист в предметной области высшего образования определяет параметрический шаблон. Исходя из имеющейся онтологической модели определяются значения параметров, входящих в данный шаблон. Например, значение параметра, определяющего название комиссии, которая принимает защиту студента: «ГАК» или «ГЭК». В качестве пе-

ременных будут выступать: ФИО студента, ФИО рецензента (научного руководителя), тема работы и т. д. Таким образом мы получим четыре документа, в которых пользователю надо заполнить минимум полей, не беспокоясь о корректности остального текста.

Синтаксические парсеры

В настоящее время имеется большое количество синтаксических парсеров [12], которые различаются по методам работы, скорости работы, доступности и по используемым платформам. Примеры некоторых популярных синтаксических парсеров приведены в табл. 1.

Таблица 1

Синтаксические парсеры

Название	Метод	Лицензия	Платформа
AOT	Грамматика HPSG	LGPL	Linux, Windows
MaltParser	Машинное обучение	Собственная	Java
Link Grammar Parser	Грамматика связей	LGPL	Linux, Windows

Синтаксический парсер AOT

На вход синтаксическому парсеру AOT [13] подается текст на естественном языке. Парсер для каждого предложения строит множество синтаксических групп (табл. 2). Каждая группа определяется следующими характеристиками: номер первого и последнего слова, тип группы (строковая константа), главная подгруппа (например, для ПРИЛ-СУЩ главная группа – существительное), граммемы группы (морфологические характеристики). Синтаксические группы разделяют на атомарные (состоящие из одного слова) и составные.

Таблица 2

Синтаксические группы [*]

Тип	Сокращенное название	Пример
Глагольная форма + контактный инфинитив	ГЛАГ_ИНФ	пойти выпить
Подлежащее	ПОДЛ	я пошел
Количественная группа	КОЛИЧ	двадцать три
Генитивная пара	ГЕНИТ_ИГ	рука Москвы
Одно или несколько прилагательных, согласованных по роду, числу и падежу со стоящим сразу после них существительным.	ПРИЛ-СУЩ	длинная унылая дорога
Предложная группа	ПГ	на холме
Однородные ИГ	ОДНОР_ИГ	Сын и дочь
Отрицание + глагольная форма	ОТР_ФОРМА	не знать
Глагольная форма + контактное прямое дополнение	ПРЯМ_ДОП	резать хлеб

[*] См.: Перечень названий групп и клауз русского синтаксиса. 2015. URL: http://aot.ru/demo/rus_syn_consts.html

[12] Синтаксический анализ. 2015. URL: https://nlpub.ru/Синтаксический_анализ
[13] Синтаксический анализ AOT. URL: http://aot.ru/docs/synan.html

Следует заметить, что главное слово атомарной группы – это единственное слово данной группы, а главное слово составной группы – это главное слово главной подгруппы. Изначально все группы рассматриваются как атомарные. Далее программа с помощью синтаксических правил пытается построить новую синтаксическую группу, проверяя одно правило за другим в определенном порядке [9]. После того как группа построена, у нее определяются главная подгруппа, список граммем и тип. Процесс повторяется циклически, пока все группы не будут построены. На выходе парсер выдает синтаксическую структуру всех предложений в формате XML [14] (рис. 4). Так как основные этапы работы программы выполняются в несколько потоков, парсер АОТ обладает очень хорошей производительностью.

```
<chunk>
<input>test.txt</input>
<sent>
  <rel name="ПРИЛ_СУЩ" gramrel="пр,мн," lemmprnt="ОБЩЕЖИТИЕ" grmprnt="но,ср,пр,мн,"
       lemmchld="СТУДЕНЧЕСКИЙ" grmchld="но,од,пр,мн," > общежитиях -> студенческих </rel>
  <rel name="ГЕНИТ_ИГ" gramrel="но,ср,им,ед," lemmprnt="ПРЕБЫВАНИЕ" grmprnt="но,ср,им,ед,"
       lemmchld="ПОСЕТИТЕЛЬ" grmchld="од,мр,рд,мн," > Пребывание -> посетителей </rel>
  <rel name="ПГ" gramrel="пр," lemmprnt="В" grmprnt=""
       lemmchld="ОБЩЕЖИТИЕ" grmchld="но,ср,пр,мн," > в -> общежитиях </rel>
  <rel name="ПГ" gramrel="тв,вн,рд," lemmprnt="С" grmprnt=""
       lemmchld="8" grmchld="тв,вн,рд," > с -> 8 </rel>
  <rel name="ПГ" gramrel="рд," lemmprnt="ДО" grmprnt=""
       lemmchld="01" grmchld="ср,жр,мр,рд," > до -> 01 </rel>
  <rel name="ПОДЛ" gramrel="" lemmprnt="ДОПУСКАТЬСЯ" grmprnt="дст,нп,нс,3л,нст,ед,"
       lemmchld="ПРЕБЫВАНИЕ" grmchld="но,ср,им,ед," > допускается -> Пребывание </rel>
</sent>
</chunk>
```

Рис. 4. Пример результата работы синтаксического парсера АОТ

Для работы парсер использует компьютерный, финансовый и локативный тезаурусы. Программа доступна для Windows и Linux под лицензией LGPL. Кроме русского также поддерживается немецкий язык.

Синтаксический парсер MaltParser

MaltParser – это синтаксический парсер, который основан на машинном обучении [15]. Программа способна работать в двух режимах: *learn* и *parse*. При запуске в режиме *learn* программе на вход подается размеченный корпус. На нем программа обучается и возвращает обученный модуль. В официальной версии доступны готовые модули для английского, французского, шведского и испанского языков. Поэтому для работы с русским языком сначала придется обучить модуль на корпусе русского языка. В качестве такого можно использовать СинТагРус. Для того чтобы запустить парсинг текста, необходимо запустить MaltParser в режиме *parse*, а также подать ему на вход обученный модуль и разбираемый текст. Начиная с версии 1.7 программа доступна для использования на языке Java. Парсер имеет одну важную особенность, отличающую его от других синтаксических парсеров. Для получения наилучшей производительности парсер необходимо оптимизировать, подбирая и изменяя большое количество параметров. Простое использование системы «из коробки» с настройками по умолчанию, скорее всего, приведет к неоптимальной производительности.

Синтаксический парсер Link Grammar Parser

Link Grammar Parser – это синтаксический парсер, основанный на грамматике связей [16]. Для работы системы создан словарь, в котором для каждого слова указаны разъемы. Разъем –

[14] URL: http://aot.ru/docs/synan.html
[15] MaltParser. URL: http://www.maltparser.org/
[16] Link Grammar Parser. 2017. URL: https://www.abisource.com/projects/link-grammar/#download

это тип связи, который может быть создан с помощью данного слова. Каждый разъем помечается знаками «+» или «–». Два разъема с противоположными знаками вместе формируют связь. Например, запись в словаре «документ: A+;» означает, что слово *документ* может быть соединено связью *A* с некоторым словом, имеющим разъем *A–*. Также существуют глобальные правила, которые контролируют, как слова могут быть соединены. Во-первых, ссылки не могут пересекаться, т. е. слова не могут соединятся через другие слова. Во-вторых, все слова в предложении должны быть косвенно связаны с каждым другим словом.

Для придания гибкости системе разъемы могут объединяться с помощью логических операций. Например, «документ: A– & B+ & C+;». На официальном сайте [17] можно найти подробное описание всех типов связей с примерами. В последних версиях программы увеличено количество поддерживаемых языков, а именно: английский, русский, немецкий и др. Также начиная с версии 5.0 была изменена лицензия с BSD на LGPL.

Семантический парсер АОТ

Достаточно известными являются семантические парсеры АОТ [18] и «Синтактико-семантический анализ русского языка» [19]. Рассмотрим более подробно семантический парсер АОТ.

Семантический парсер АОТ преобразует предложение русского языка в набор семантических узлов и семантических отношений. Семантическими узлами могут быть: слова, знаки препинания, устойчивые обороты (например, *по правде говоря*), абстрактные узлы (например, *этот человек*), устойчивые словосочетания (например, *не хватило духа*), жесткие синтаксические группы (например, *двадцать два*). Семантическое отношение – это некоторое двуместное отношение (связь), усматриваемое в тексте носителем языка. Семантические отношения записываются как *R*(*A*, *B*), где *R* – вид семантического отношения, *A* – зависимый член, а *B* – главный член. При этом считается, что отношение *R*(*A*, *B*) эквивалентно утверждению, что «*A* является *R* для *B*». Например, для фразы *ножка стула* будет построена формула *ЧАСТЬ*(*ножка*, *стул*), которой эквивалентно утверждение «Ножка является *ЧАСТЬ*ю стула». В табл. 3 приведены примеры основных семантических отношений [20].

Таблица 3

Семантические отношения

Название	Примеры	Структура
ЧАСТЬ	ножка стула	**ЧАСТЬ**(НОЖКА, СТУЛ)
ЗНАЧ	Высота дома – 20 метров	**ЗНАЧ**(20 МЕТРОВ, ВЫСОТА)
ИДЕНТ	Квартира N 5	**ИДЕНТ** (N 5, КВАРТИРА)
ПРИЗН	Большая ваза	**ПРИЗН**(БОЛЬШОЙ, ВАЗА)
ПРИНАДЛ	дом Пушкина	**ПРИНАДЛ**(ДОМ, ПУШКИН)
ПРИЧ	деревья повалены ураганом	**ПРИЧ**(УРАГАН,ПОВАЛИТЬ)
РЕЗЛТ	испечь пирог	**РЕЗЛТ**(ПИРОГ,ИСПЕЧЬ)
АВТОР	Роман Пушкина	**АВТОР**(ПУШКИН, РОМАН)
ВРЕМЯ	Это будет завтра	**ВРЕМЯ**(ЗАВТРА, БУДЕТ)
ИМЯ	Президент Иванов	**ИМЯ**(ИВАНОВ, ПРЕЗИДЕНТ)
ИНСТР	Пилить пилой	**ИНСТР**(ПИЛА, ПИЛИТЬ)
КОЛИЧ	три банана	**КОЛИЧ**(ТРИ, БАНАН)

[17] Ibid.

[18] Первичный семантический анализ АОТ. URL: http://aot.ru/docs/seman.html

[19] Технологии автоматического анализа текстов. URL: http://nlp.isa.ru/index.php/component/portal/?view=projsintsemanalysis

[20] Семантические отношения, используемые в модуле поверхностно семантического анализа «Диалинг». 2015. URL: http://aot.ru/docs/SemRels.htm

Следует отметить, что кроме указанных отношений также используются «технические» связи, которые никак не характеризуют текстовую зависимость по смыслу. Они могут возникать в ситуации, если в семантических словарях не нашлось необходимой информации, а связь необходимо создать.

В основе системы АОТ лежит несколько весьма важных словарей, в том числе русский общесемантический словарь (РОСС) [10], словарь оборотов и словарь групп времени (Time Ross).

Кроме словарей система АОТ также использует тезаурусы. Они состоят из синонимических множеств, называемых концептами. Это аналог синсетов в WordNet. Понятия («концепты») связаны друг с другом бинарными отношениями. В итоге в системе АОТ присутствуют три тезауруса: локативный, финансовый, компьютерный. Локативный тезаурус, отвечает за слова-категории и слова, связанные с конкретными географическими объектами. Финансовый тезаурус содержит около 2 500 концептов и отвечает за экономические термины, такие как организация, должность, профессия и т. д. Компьютерный тезаурус состоит из 1 500 концептов и отвечает за компьютерную терминологию.

Следует заметить, что работа системы обеспечивается не только словарями и тезаурусами, но также и синтаксическим парсером АОТ, который предоставляет информацию, необходимую для начала семантического анализа.

Программа доступна для Windows и Linux под лицензией LGPL.

Заключение

В настоящей работе представлен краткий обзор систем, предназначенных для автоматизации порождения документов. Разработаны автоматизированные методы извлечения знаний из нормативных документов, основанные на семантическом и синтаксическом анализе текстов естественного языка. Предложен подход к автоматизированному порождению корректных документов на естественном языке с логическим контролем их правильности, основанный на использовании параметрических шаблонов.

Список литературы

1. *Деревянко Д. В.*, *Пальчунов Д. Е.* Формальные методы разработки вопросно-ответной системы на естественном языке // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2014. Т. 12, вып. 3. С. 34 –47.

2. *Wu F.*, *Weld D. S.* Automatically refining the wikipedia infobox ontology // Proc. of the 17th InternationalConference on World Wide Web, WWW. ACM, 2008. P. 635–644.

3. *Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., Mendes P. N., Hellmann S., Morsey M., Kleef P. van, Auer S., Bizer C.* DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia // Semantic Web Journal. 2015. Vol. 6, iss. 2. P. 167–195.

4. *Sirin E., Parsia B.* SPARQL-DL: SPARQL query for OWL-DL // Third OWL Experiences and Directions Workshop (OWLED). 2007.

5. *Palchunov D., Yakhyaeva G., Yasinskaya O.* Software system for the diagnosis of thae spine diseases using case-based reasoning // Proc. of the International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON / SibMedInfo – 2015). Novosibirsk, 2015. P. 150–155.

6. *Naydanov Ch., Palchunov D., Sazonova P.* Development of automated methods for the prevention of risks of critical conditions, based on the analysis of the knowledge extracted from the medical histories // Сиб. науч. мед. журн. 2016. Т. 36, вып. 1. С. 105–113.

7. *Palchunov D., Yakhyaeva G., Dolgusheva E.* Conceptual Methods for Identifying Needs of Mobile Network Subscribers // Proc. of the 13th International Conference on Concept Lattices and Their Applications. Moscow, 2016. P. 147–160.

8. *Motik B., Shearer R., Horrocks I.* Hypertableau reasoning for description logics // Journal of Artificial Intelligence Research. 2009. No. 36. P. 165–228.

9. *Сокирко А. В.* Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ): Дис. … д-ра техн. наук. М., 2001. 120 с.

10. *Леонтьева Н. Н.* Русский общесемантический словарь (РОСС): структура, наполнение // НТИ. Сер. 2. 1997. № 12. С. 5–20.

Материал поступил в редколлегию 31.07.2017

D. E. Palchunov, **A. A. Fink**

[1] *Novosibirsk State University*
1 Pirogov St., *Novosibirsk*, *630090*, *Russian Federation*

[2] *Institute of Mathematics SB RAS*
4 Academician Koptyug Ave., *Novosibirsk*, *630090*, *Russian Federation*

palch@math.nsc.ru, *a.fink@g.nsu.ru*

THE DEVELOPMENT OF AUTOMATED METHODS OF GENERATION OF OFFICIAL DOCUMENTS IN NATURAL LANGUAGE

The paper is devoted to the problem of automated generation of official documents in natural language with logical control of their correctness. Existing approaches to automation of document generation are considered. We present a method of automated generation of logically correct documents in natural language based on the use of parametric templates.

Keywords: knowledge extraction, knowledge representation, regulatory documents, official documents, parametric templates.

References

1. Derevyanko D. V., Palchunov D. E. Formal methods of development of the question-answering system on natural language. *Vestnik NSU. Series: Information Technologies*, 2014, vol. 12, no. 3, p. 34–47. (In Russ.)

2. Wu F., Weld D. S. Automatically refining the wikipedia infobox ontology. *Proc. of the 17th InternationalConference on World Wide Web, WWW*. ACM, 2008, p. 635–644.

3. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., Mendes P. N., Hellmann S., Morsey M., Kleef P. van, Auer S., Bizer C. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2015, vol. 6, iss. 2, p. 167–195.

4. Sirin E., Parsia B. SPARQL-DL: SPARQL query for OWL-DL. *Third OWL Experiences and Directions Workshop* (*OWLED*), 2007.

5. Palchunov D., Yakhyaeva G., Yasinskaya O. Software system for the diagnosis of thae spine diseases using case-based reasoning. *Proc. of the International Conference on Biomedical Engineering and Computational Technologies* (*SIBIRCON* / *SibMedInfo – 2015*). Novosibirsk, 2015, p. 150–155.

6. Naydanov Ch., Palchunov D., Sazonova P. Development of automated methods for the prevention of risks of critical conditions, based on the analysis of the knowledge extracted from the medical histories. *Siberian Scientific Medical Journal*, 2016, vol. 36, iss. 1, p. 105–113.

7. Palchunov D., Yakhyaeva G., Dolgusheva E. Conceptual Methods for Identifying Needs of Mobile Network Subscribers. *Proc. of the 13th International Conference on Concept Lattices and Their Applications*. Moscow, 2016, p. 147–160.

8. Motik B., Shearer R., Horrocks I. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 2009, no. 36, p. 165–228.

9. Sokirko A. V. Semantic dictionaries in automatic text parsing (On materials of the system DIALING): Dissertation of Doctor of Technical Sciences. Moscow, 2001, 120 p. (In Russ.)

10. Leonteva N. N. Russian all-semantic dictionary (ROSS): structure, filling. *NTI. Series 2*, 1997, no. 12, p. 5–20. (In Russ.)

For citation:

Palchunov D. E., Fink A. A. The Development of Automated Methods of Generation of Official Documents in Natural Language. *Vestnik NSU. Series*: *Information Technologies*, 2017, vol. 15, no. 3, p. 79–89. (In Russ.)