УДК 004.82:004.912

Лексико-семантические шаблоны как инструмент декларативного описания языковых конструкций и лингвистического анализа текста

Тимофеев П.С. (Институт систем информатики СО РАН),
Сидорова Е.А. (Институт систем информатики СО РАН)

Статья посвящена проблемам извлечения языковых конструкций, в том числе числовых и символьных данных, значимых для заданной предметной области. Предложен подход к описанию естественно-языковых конструкций с помощью лексико-семантических шаблонов и рассмотрен язык описания шаблонов на основе языка YAML. Лексико-семантический шаблон – это структурный образец целевой языковой конструкции с указанным составом и лексико-семантическими свойствами. В случае успешного сопоставления шаблона с фрагментом текста формируется лексический объект, которому приписываются формальные (позиционные) и семантические (класс и свойства) характеристики. В статье представлена архитектура веб-редактора для разработки и тестирования лексико-семантических шаблонов и описан эксперимент по созданию двух специализированных словарей: 1) словарь наименований институтов, должностей, званий и их сокращений и 2) словарь числовых/временных конструкций. Создаваемая технология поддерживает лексико-семантический анализ текста на основе шаблонов и может быть использована как независимо при решении задачи извлечения информации из небольших текстов, так и в составе других систем извлечения информации. Предлагаемый метод эффективен для распознавания параметрических конструкций, содержащих оценку значений параметров объектов (сущностей или событий) предметной области.

Ключевые слова: *лексико-семантический шаблон, извлечение языковых конструкций, язык описания шаблонов, предметно-ориентированный словарь.*

1. Введение

Необходимость извлечения информации из огромного количества существующих и постоянно появляющихся текстов на естественном языке делает востребованным развитие методов интеллектуального анализа текста. При анализе текста большой пласт информации

представляется языковыми конструкциями, не обрабатываемыми инструментами, основанными на морфологических словарях. В первую очередь это:

- числовые данные заданные в явной форме (*год, вес, расстояние*);
- символьные данные (сокращения, аббревиатуры, номера телефонов и т. д.);
- параметрические конструкции, т.е. языковые конструкции содержащие оценку значения параметров объектов выраженных не только в явной форме (лексически или числом), но и в неявной форме (например, в сравнении с другими параметрами).

Данные языковые конструкции распространены в различного рода документах - медицинских протоколах, научно-технической литературе, производственной документации и др. Они, как правило, являются значимыми с точки зрения информационного содержания документа и часто интерпретируются в зависимости от предметной области и решаемой задачи. Для извлечения такого рода конструкций используются либо регулярные выражения, либо специализированные языки шаблонов, которые ориентированы на специалистов филологов и экспертов в различных областях знаний.

В литературе, в зависимости от типа учитываемой в конструкции языковой информации, шаблоны подразделяются на грамматические [7], лексико-грамматические [5] и лексико-синтаксические [1,6]. К грамматическим относятся шаблоны, в которых указываются грамматические характеристики слов, входящих в состав языкового выражения. В лексико-грамматических шаблонах также могут указываться конкретные лексемы, а лексико-синтаксические обеспечивают проверку согласования входящих лексем по грамматическим характеристикам. Обычно, все эти виды шаблонов используются для извлечения терминов (в том числе многословных) в задачах построения словников. Следует отметить язык Jape [8], который позволяет записывать правила распознавания языковых конструкций и формировать на их основе атрибутно-объектную модель текста для последующей работы на языке Java. На основе данного языка была разработана система выделения специфических объектов из текста RCO Pattern Extractor [2]. Данная система использует собственные язык и редактор правил, но является коммерческой и закрытой, что затрудняет ее исследование и использование в научных проектах.

Разрабатываемое в нашем коллективе решение позволяет анализировать произвольные символьные конструкции и формировать на их основе множество объектов предметной области. В рамках данного решения формируются предметно-ориентированные словари лексико-семантических шаблонов и разрабатывается инструментарий, обеспечивающий разработку, тестирование и применение словарей в лингвистическом анализе текста. В предыдущих работах [3, 4] для разработки шаблонов использовались специализированные

редакторы, которые обеспечивали только графическое представление шаблонов. Данные средства, несмотря на удобство для конечного пользователя, не поддерживают совместную разработку ресурсов, сложны для использование сторонними пограммными продуктами, а также трудоемки в поддержке и масштабировании решений. Таким образом, назрела потребность в разработке языка для записи шаблонов на основе стандартных форматов представления данных и создание веб-ресурса, обеспечивающего пользователя универсальной средой для разработки специализированных словарей .

2. Язык описания лексико-семантических шаблонов

Лексико-семантический шаблон – это структурный образец целевой языковой конструкции с указанным составом и лексико-семантическими свойствами. В случае успешного сопоставления шаблона с фрагментом текста формируется лексический объект, которому приписываются формальные (позиционные) и семантические (класс и свойства) характеристики. Разрабатываемый язык описания лексико-семантических шаблонов Diglex поддерживает набор логических конструкций: альтернатива, ссылка на шаблон, повторитель, опциональность, условие на контекст, дистантный контекст и др. В рамках данной работы предложен новый синтаксис языка на основе языка YAML (http://yaml.org/spec/1.2/spec.html), который обладает следующими преимуществами.

- Широкое распространение и известность.
- Поддержка всеми основными редакторами исходного кода (продуктами JetBrains, Sublime, Atom, Eclipse, VS Code и другими).
- Богатый синтаксис, состоящий из известных типов данных, таких как: ассоциативный массив, список, строка, число и т. д.
- Компактность записи.

Рассмотрим примеры простых шаблонов Diglex, обеспечивающих извлечение из текста объектов класса “должность”.

```
---
# символом “#” обозначаются комментарии
class:
  # наименование класса
  name: должность
  # наименование базового класса (не обязательно)
  parent: базовый-класс
  # произвольный набор имен и значений свойств класса
```

```
  properties:
    Одушевленность: true
    Тематика: работа и профессия
  # для каждого класса описывается список шаблонов
  templates:
    - name: младший научный сотрудник
      properties:
        Квалификационный-уровень: 1
      # Список возможных образцов для сопоставления шаблона
      cases:
        - младш{...} научн{...} сотрудник{...}
        - мнс
    - name: ведущий научный сотрудник
      properties:
        Квалификационный-уровень: 3
      cases:
        - ведущ{...} научн{...} сотрудник{...}
        - внс
```

Класс описывается именем (поле “name”), указанием родительского класса (поле “parent”) ассоциативным массивом произвольным набором свойств (поле “properties”), а также набором шаблонов (поле “templates”). В данном примере шаблоны записываются вместе с определением класса. Также допустим синтаксис, в котором шаблон описывается отдельно от класса с указанием его имени.

```
---
template:
  - name: старший научный сотрудник
    class: должность
    properties:
      Квалификационный-уровень: 2
    cases:
      - старш{...} научн{...} сотрудник{...}
      - снс
```

Имя класса служит уникальным идентификатором для организации связи с шаблонами и другими классами при наследовании. Набор свойств класса (поле “properties”) пользователь задает в зависимости от решаемой задачи и предметной области. Данные свойства – это

семантические атрибуты (приписываемые найденным по шаблонам объектам), которые можно описать на уровне класса с возможностью переопределить значения на уровне шаблона. В Diglex каждое свойство имеет свой тип определяемый в зависимости от типа YAML-значения по умолчанию. В данном примере поле “Одушевленность” имеет логический тип данных, “Тематика” – строковый, а “Квалификационный-уровень” – числовой. Отсутствие требования всегда явно указывать тип данных приводит к более компактной записи.

Каждый шаблон содержит список образцов (поле “cases”), состоящий хотя бы из одного образца для сопоставления с текстом на естественном языке. Сопоставление с шаблоном считается успешным при успешном сопоставлении с любым элементом из списка “cases”. Для записи таких образцов предложен формальный язык подобный регулярным выражениям. Например, сопоставление с образцом ‘снс’ будет успешным только при точном вхождении в текст (без учета регистра). Язык поддерживает использование следующих возможностей.

1. Хвост. Позволяет указать неизменяемую часть слова.
   Пример: *старш{...}*
   Такой образец будет соответствовать словам старший, старшая и т. д. Кроме того, имеется возможность указать длину хвоста интервалом значений: *старш{...<2,3>}*
2. Регистр. По умолчанию регистр слов игнорируется, т. е. образец “внс” будет соответствовать как слову “ВНС” так и “внс”, “Внс” и т. д. Для указания того, что регистр учитывается, необходимо писать: *внс<cs>*.
3. Ссылка на другой шаблон. Текст образца может иметь ссылку на другой шаблон, для этого необходимо указать имя шаблона в квадратных скобках. Пример: *”институт{...} математики [СО РАН]”*.
4. Повторитель. Используется для описания последовательности повторяющихся элементов. Имеется возможность указать точную длину последовательности либо интервал с количеством возможных повторений. Пример образца для извлечения номера телефона: *[цифра]<2,3>-[цифра]<2>-[цифра]<2>*.
5. Опция. Используется для обозначения необязательных элементов. Сопоставление с образцом произойдет в случае наличия или отсутствия элемента.
   Пример: *н{.}<?>{ }<?>с{.}<?>*
   Данный шаблон соответствует таким строкам: “н.с.”, “н. с. ”, “нс”, и т. д.

6. Дистантный контекст. Позволяет выделить фрагмент текста неизвестной заранее длины по заданным начальным и конечным элементам.
   Пример: *Уважаемый<-.->!*
   Элемент <-.-> будет соответствовать любому фрагменту текста, начинающемуся со слова “Уважаемый” и заканчивающемуся на восклицательный знак.
7. Исключающее условие. Предназначено для накладывания ограничений на контекст образца.
   Пример: *научный<not> сотрудник{...}*
   Такой шаблон будет соответствовать строке “Наш сотрудник”, но не “научный сотрудник”.
8. Группировка. Для применения модификатора к группе элементов применяется следующий синтаксис:
   {научный сотрудник}<not> опубликовал статью
   В данном примере модификатор <not> применяется к строке “научный сотрудник”.

3. Веб-редактор шаблонов

Благодаря синтаксису YAML для создания шаблонов Diglex пользователь может воспользоваться практически любым современным редактором без каких-либо специальных плагинов. Это удобно для быстрого старта и для пользователей, которые уже имеют привычку работы в определенных редакторах. Однако, в таком случае возникают следующие проблемы.

- Так как имена классов и шаблонов это просто строки в YAML - отсутствует их автоматическое дополнение и возможность “перейти к определению”.
- Инструменты поиска, сортировки и быстрого доступа к шаблонам по имени и другим характеристикам.
- Образцы для сопоставления в рамках YAML представлены обычными строками. Соответственно для них отсутствует не только автоматическое дополнение, но и элементарная подсветка синтаксиса.
- Отсутствие встроенной возможности проверять работу шаблонов во время их разработки, например путем применения к небольшому тексту.
- Отсутствие проверки корректности системы шаблонов, например, отсутствие циклов в ссылках, отсутствие использование неопределенных шаблонов и т.п.
- Необходимость использовать дополнительные инструменты для коллективной работы над шаблонами. При этом, обычной практикой является использование систем

контроля версий таких как Git, что является достаточно сложной задачей для неподготовленного пользователя.

Для решения этих проблем есть два пути: разработать плагин для одного или нескольких редакторов либо разработать специализированный редактор базирующийся на существующих решениях с открытым/свободным исходным кодом. Подход с созданием плагина требует выбора определенного редактора, что сильно ограничивает дальнейшее развитие его возможностями. Более перспективным представляется вариант разработки специализированного редактора, причем работающего в режиме онлайн, что может позволить предоставить простые возможности коллективной разработки. Также немаловажным аргументом в пользу веб-редактора является отсутствие требования установки дополнительного ПО на компьютер пользователя. Это удобно с точки зрения пользователя, а также снимает с разработчика необходимость поддерживать сборку редактора и ядра Diglex для различных операционных систем.

Рассмотрим требования к веб-редактору Diglex.

- Подсветка синтаксиса.
- Возможность навигации по иерархии классов и шаблонов.
- Возможность перейти к определению класса или именованного шаблона.
- Автоматическое дополнение. Если пользователь начинает вводить имя класса (шаблона) в месте предусмотренном для этого синтаксисом языка, то он должен увидеть выпадающий список со всеми возможными именами классов (шаблонов).
- Возможность проверить корректность работы шаблона в режиме онлайн путем его сопоставления с текстом, который также доступен для редактирования.
- Экспорт и импорт словаря шаблонов, текстов и результатов их обработки как на локальную машину пользователя, так и на серверную часть.
- Возможность управления правами доступа пользователей для редактирования словарей Diglex.

На данный момент реализован прототип редактора (diglex.forkode.ru), удовлетворяющий минимальным требованиям и позволяющий опробовать предложенный язык лексико-семантических шаблонов на практических примерах. Архитектура редактора (см. Рис.1) заложена такой, чтобы делать редактор простым в реализации и одновременно не препятствовать его дальнейшему развитию.

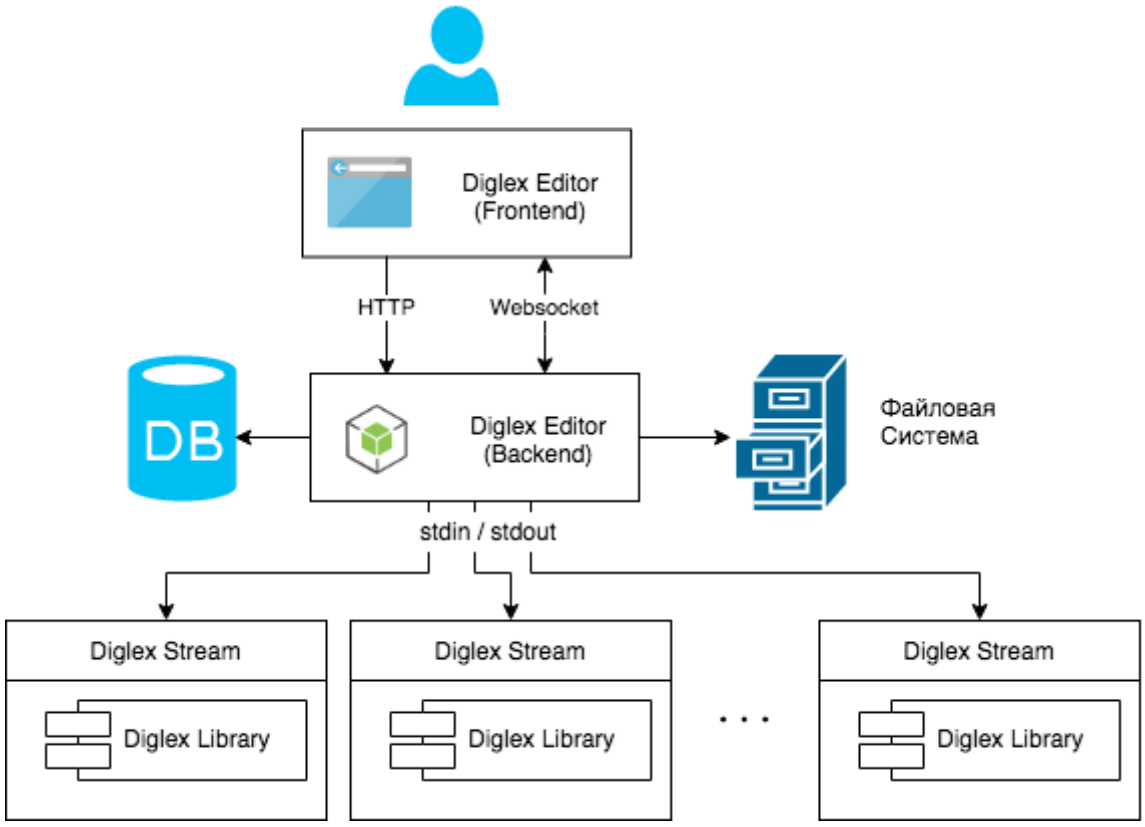

Рис.1. Архитектура веб-редактора Diglex.

В результате были сформулированы следующие архитектурные принципы на основе клиент-серверного подхода:

- Клиентская часть (Frontend) отвечает за взаимодействие с пользователем. Построена по принципу “Single Page Application” (SPA), т. е. используется единственный HTML-документ который подгружает все необходимые для работы скрипты и стили динамически. Это позволяет сделать богатый и отзывчивый интерфейс по качеству сравнимый с нативными приложениями для операционных систем. Для Frontend части был выбран JavaScript и React фреймворк.
- Серверная часть (Backend) отвечает за логику работы редактора такую как: работа с файловой системой, управление пользователями, постановка задач на вычисление с помощью ядра Diglex и предоставляет полученные результаты для Frontend. Backend контролирует работу ядра Diglex на предмет наличия ошибок и обеспечивает его перезапуск в случае необходимости. Для реализации был выбрана платформа Node.js.
- Серверная часть не линкуется напрямую с ядром Diglex, вместо этого создано отдельное приложение Diglex Stream (язык реализации C++), которое взаимодействует с Backend посредством стандартных потоков ввода-вывода (stdin, stdout).

Предложенная архитектура делает модули хорошо изолированными и простыми в программировании, сохраняя возможность легкого масштабирования. Масштабирование может производиться по принципу один процесс Diglex Stream - на некоторое количество пользователей, либо на каждый словарь - некоторое количество процессов Diglex Stream.

4. Экспериментальное исследование

С использованием разработанного инструментария были созданы два словаря: 1) словарь наименований институтов, должностей, званий и их сокращений и 2) словарь числовых/временных конструкций. Первый словарь был разработан на основе номенклатурных списков РАН. Всего в словарь входит 646 шаблонов и 36 классов. Словарь позволяет находить в тексте:

- Наименования организаций РАН. В частности: наименования академий, институтов, университетов, колледжей, заводов, библиотек, училищ, музеев и научных центров.
- Наименование должностей, степеней и званий РАН. Например: *ведущий специалист, младший научный сотрудник, доктор биологических наук, доцент, академик-секретарь* и т. д.

Второй словарь разработан с целью представить шаблоны с семантикой “время”. В него входят шаблоны позволяющие искать в текстах следующие временные конструкции.

- Часовое время. Отвечают на вопросы: “сколько времени?”, “который час?”, “когда?”, “в котором часу” и т.д. Например: *в час дня, три часа дня, пятнадцать минут первого*.
- Обозначение даты, времени действия. Такие конструкции отвечают на вопросы: “какое сегодня число?”, “Когда?”, “Какого числа?” и т. д., “В какой день?”. Например: *сегодня пятое декабря, этой зимой, на той неделе*.
- Обозначение продолжительности действия. Такие конструкции отвечают на вопросы: “сколько времени?” -Например: *одну секунду, в течение часа, на протяжении* и т. д.
- Обозначение момента начала и конца действия. Такие конструкции отвечают на вопросы: “с какого времени”, “до какого времени?”, “с которого часа и до которого часа?” и т. д. Например: *через два часа, с утра до вечера, с апреля по ноябрь включительно*.
- Обозначение времени, по прошествии которого произойдет, закончится или начнется действие. Такие конструкции отвечают на вопросы: “когда?”, “через сколько времени?”. Например: через пятнадцать минут, спустя годы.

- Обозначение срока действия и времени необходимого для достижения результата действия. Такие конструкции отвечают на вопросы: “за сколько времени?”, “за какое время?”. Например: *в одно мгновение, за одну минуту*.
- Обозначение срока, в течение которого сохраняется результат действия. Такие конструкции отвечают на вопросы: “на сколько времени?”, “на какой срок?”, “на какое время?”. Например: *на пять дней, на две недели*.
- Обозначение времени повторяющегося действия. Такие конструкции отвечают на вопросы: “когда?”, “как часто?”. Примеры: *ежедневно, время от времени, раз в два дня*.
- Обозначение одновременности действий. Такие конструкции отвечают на вопросы: “когда?”, “в какое время?”. Примеры: *в ходе, в то же самое время*.
- Выражение последовательности действий. Такие конструкции отвечают на вопросы: “когда?”, “в какое время?”. Например: *до того как, прежде чем*.

Созданные словари были опробованы на двух коллекциях текстов: “Хроники СО РАН” (http://www.nsc.ru/HBC/events/chronicle.html) и “Постановления Президиума СО РАН”. коллекция “Хроники СО РАН” содержит краткие описания 1218 событий, связанных с СО РАН, происходивших с 1957 г. по 1992 г. Каждый текст – это отрывок из документов различных жанров: официальных постановлений, деловых писем, газетных статей и т.п. В коллекцию “Постановления Президиума СО РАН” входит 159 официальных постановлений, положений и приложений к ним. Статистика результатов применения словаря на данных коллекциях текстов представлена в Таблице 1.

Отсутствие размеченного корпуса текста не позволяет применить классические способы оценки полноты и точности анализа. Общим свойством разрабатываемых вручную шаблонов является обеспечение высокой точности. Мы предложили эвристическую методику оценки точности и полноты на основе косвенных признаков, поддающихся автоматическому вычислению. В качестве критерия точности мы рассматриваем наличие вариативности извлеченных объектов в одной позиции. Например, из текста “*если во время приема граждан решение поставленных вопросов невозможно*” извлекается конструкция “*во время*” как объект обозначающий продолжительность действия и, одновременно, как объект обозначающий одновременные действия. Верный вариант может быть только один, но в данном случае система не в состоянии его определить и формирует оба. В качестве критерия полноты рассматриваем наличие частично собранных шаблонов, таких как инициалы с невыделенной фамилией (как показатель ненайденной фамилии персоны), аббревиатуры, используемые в составе названий институтов (СО, АН, СССР) и др. Наличие таких частей

означает, как правило, что объект не смог собраться полностью из-за отсутствия необходимых описания, что ухудшает показатели полноты.

Таблица 1. Результаты применения лексико-семантических шаблонов на текстовых коллекциях.

	Коллекция “Хроники СО РАН”		**Коллекция “Постановления Президиума СО РАН”**	
	Словарь организаций	Словарь временных конструкций	Словарь организаций	Словарь временных конструкций
Размер коллекции (число слов)	57 521		153 331	
Применено шаблонов	120	18	134	18
Найдено классов	19	12	19	14
Извлечено объектов	7 400	2 642	27 296	5 248
Точность	90.64 %	99.47 %	84.81 %	99.40 %
Полнота	94.03 %	90.82 %	90.04 %	74,14%

Результаты показали, что полученные словари извлекают достаточно большое количество объектов и достаточно хорошо покрывают требуемые конструкции. Низкие показатели точности относительно полноты для словаря наименований связаны со сменой аббревиатуры в названиях институтах в 90-х годах, что дало вариативность при определенных сокращениях. Во втором словаре показатель точности ожидаемо высок. Также получены хорошие показатели полноты для словарей наименований, что связано в первую очередь с жанром источников, в которых принято использовать номенклатурные наименования. Для словаря временных конструкций получены низкие показатели полноты, что связано с необходимостью совершенствовать систему временных шаблонов, в частности для извлечения дат.

В ходе разработки шаблонов при необходимости записать в шаблоне множество словоформ одной лексемы наблюдалась нехватка возможности указания их грамматических

признаков. Частично эту проблему решает возможность указания “хвоста”, но для ряда случаев он охватывает словоформы других лексем. Характерный пример: все словоформы лексемы “май” записываются как “ма{...}”, что соответствует как слову “май”, так и слову “март”. Несмотря на обнаруженные недостатки, система в целом продемонстрировала свою пригодность для выполнения практических задач.

5. Заключение

В работе предложен подход к описанию естественно-языковых конструкций с помощью лексико-семантических шаблонов. Представлен язык описания шаблонов Diglex на основе синтаксиса языка YAML, рассмотрены его основные конструкции и возможности. Для удобства разработки шаблонов Diglex спроектирован веб-редактор и реализован его прототип. Полученный инструментарий опробован при создании двух словарей.

Язык шаблонов Diglex отличается простотой и декларативностью. С его помощью удобно создавать шаблоны для извлечения параметрических конструкций, извлечения числовых значений и нестандартных символьных обозначений. Разрабатываемый веб-редактор снижает порог вхождения для новых пользователей Diglex и предоставляет им возможности для коллективной работы. Система в целом расширяет возможности традиционных лексикографических систем и упрощает их использование.

Одним из недостатков существующего на данный момент языка лексико-семантических шаблонов является отсутствие средств для указания грамматических признаков распознаваемых лексических единиц, что является направлением дальнейших исследований. Также требует развития веб-редактор. На данный момент разработан только его прототип и планируется разработка полноценного редактора в соответствии с требованиями, обозначенными в настоящей статье.

Работа выполнена при поддержке РФФИ (грант № 17-07-01600).

Список литературы

1. Большакова Е.И., Баева Н.В., Бордаченкова Е.А., Васильева Н.Э., Морозов С.С. Лексико-синтаксические шаблоны в задачах автоматической обработки текстов // Компьютерная лингвистика и интеллектуальные технологии: тр. межд. конференции Диалог‘2007. М.: Изд-во РГГУ, 2007. С. 70-75.
2. Ермаков А.Е., Плешко В.В., Митюнин В.А. RCO Pattern Extractor: компонент выделения особых объектов в тексте. //X II Международная научная конференция. Сборник трудов Москва, 2003. С. 312-317.

3. Жигалов В. А. Жигалов Д. В., Жуков А. А., Кононенко И.С., Соколова Е.Г., Толдова С.Ю Система Alex, как средство для многоцелевой автоматизированной обработки текстов // Труды международного семинара Диалог'2002 "Компьютерная лингвистика и интеллектуальные технологии". — Москва : Наука, 2002. — Т. 2. – С.192-208.
4. Ковалев А.И., Сидорова Е.А. Инструмент разработки предметных словарей на основе лексических шаблонов DigLex // Материалы Всероссийской конференции с международным участием «Знания – Онтологии – Теории» (ЗОНТ–2015), 6 - 8 октября 2015 г., Новосибирск. – Новосибирск: Институт математики им. С.Л. Соболева СО РАН, 2015. –Т1. – С. 123-130.
5. Митрофанова О.А., Захаров В.П. Автоматизированный анализ терминологии в русскоязычном корпусе текстов // Компьютерная лингвистика и интеллектуальные технологии: тр. межд. конференции «Диалог–2009». М.: 2009. С. 321-328.
6. Рабчевский Е., Булатова Г., Шарафутдинов И. Формализм записи лексико-синтаксических шаблонов в задаче автоматизации процесса построения онтологий // Труды десятой всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» –RCDL'2008. Дубна: ОИЯИ, 2008. С. 103-106.
7. Сидорова Е. А. Многоцелевая словарная под- система извлечения предметной лексики // Компьютерная лингвистика и интеллектуальные технологии: тр. межд. конф. «Диалог–2008». М.: 2008. Вып. 7 (14). М.: Изд–во РГГУ, 2008. С. 475-481.
8. General Architecture for Text Engineering. http://www.gate.ac.uk/