УДК 81'322.2

АЛГОРИТМ СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТА, ОСНОВАННЫЙ НА БАЗОВЫХ СЕМАНТИЧЕСКИХ ШАБЛОНАХ С УДАЛЕНИЕМ

А.В. Мочалова[a]

[a] Петрозаводский государственный университет, 185910, г. Петрозаводск, Россия, stark345@gmail.com

Аннотация. В связи с ростом объема текстовой информации все более актуальными становятся системы автоматической обработки текста. Одной из основных задач таких систем является задача семантического анализа. В работе предлагается алгоритм поиска семантических зависимостей между частями предложений анализируемого текста, основанный на сопоставлении текста с базовыми семантическими шаблонами. Каждое предложение, поступающее на вход программы, постепенно сокращается: некоторые части предложения в соответствии с правилами, описанными в семантических шаблонах, добавляются в очередь с приоритетом, после чего на каждой итерации алгоритма из анализируемого предложения изымается та его часть, которая имеет в очереди наибольший приоритет. Для определения приоритета в такой очереди используются два значения: значение приоритета группы, к которой принадлежит семантическая зависимость, описанная в шаблоне, и позиция слова (или последнего слова из набора, если элемент, хранимый в очереди, состоит из нескольких слов) в анализируемом предложении. В ходе работы составлено 2160 базовых семантических шаблонов, а также на языке программирования Java реализован предлагаемый в статье алгоритм. Применение в процессе реализации алгоритма экспертной системы Drools, использующей алгоритм быстрого сопоставления с шаблонами PHREAK, обеспечило высокую скорость работы семантического анализатора. По результатам тестирования сделан вывод, что предложенный алгоритм семантического анализа без использования экспертной системы Drools работает медленнее в среднем в 6–8 раз. Программная реализация алгоритма показала, что результаты работы программы быть использованы в системах автоматической обработки текстов. Разработанный семантический анализатор используется в качестве составного модуля интеллектуальной вопросно-ответной системы.

Ключевые слова: семантические зависимости, семантический анализатор, семантические шаблоны.

ALGORITHM FOR SEMANTIC TEXT ANALYSIS BY MEANS OF BASIC SEMANTIC TEMPLATES WITH DELETION

A.V. Mochalova[a]

[a] Petrozavodsk State University, 185910, Petrozavodsk, Russia, stark345@gmail.com

Abstract. The systems of automatic text processing have become more and more important due to the constant growth of textual data. One of the main issues arising in such systems is a problem of semantic analysis. The paper deals with an algorithm for finding semantic dependencies by means of basic semantic templates with deletion. While working with the Drools expert system (and PHREAK algorithm for fast pattern matching) we have developed and implemented a semantic analyzer for construction of semantic dependencies between parts of a sentence. During the semantic analysis we add some text parts to the priority queue according to the rules described in the semantic templates, and then at each iteration of the sentence being analyzed we drop some segment of the analyzed text which has the highest priority in the queue. To determine the priority in this queue two values are used: the priority of semantic relationship group and word position. The proposed algorithm is implemented in Java. We have prepared 2160 rules using Drools expert system. The software implementation of the proposed algorithm has shown its applicability for the systems of automatic text processing. Testing results have proved that suggested algorithm of semantic analysis without Drools expert system operates 6-8 times slower, on the average. We use proposed semantic analyzer as a composite module to intellectual question-answering system.

Keywords: semantic dependencies, semantic analyzer, semantic templates.

Введение

В наши дни все больше времени люди тратят на анализ текстов, предположительно содержащих интересующие их факты. Для сокращения этого времени создаются системы автоматической обработки текста (АОТ), призванные упростить задачу нахождения нужной информации в большом объеме текста. Одной из задач автоматической обработки текстов является семантический анализ, способ реализации которого предлагается в настоящей работе.

Одним из самых распространенных способов АОТ является его сопоставление различным шаблонам. Например, в работе [1] автором описывается метод автоматического построения онтологий на основе лексико-синтаксических шаблонов. Метод синтаксических шаблонов, основанный на концепции падежной грамматики Ч. Филлмора, описанный в работах [2, 3], используется для автоматического преобразования структур знаний, хранимых в базе данных (БД), в тексты естественного языка.

Шаблоны также используются для формализации текстовой информации, что описано автором работы [4]. В этой же работе предлагается метод автоматического формирования шаблонов для идентификации сущностей и событий, а также алгоритмы формирования графа синтаксико-семантических отношений с помощью синтаксико-семантических шаблонов, создание которых предлагается автоматизировать. Синтаксические шаблоны сборки именных групп применяются для извлечения терминов-словосочетаний [5, 6].

Одним из эффективных методов извлечения семантических отношений является метод лексических шаблонов [7, 8]. Marti Hearst [8] показал, что данный метод показывает «достаточно адекватный» результат для идентификации родо-видовых отношений.

Шаблоны являются неотъемлемой частью машинного поиска в коллекции документов, а также во многих других областях автоматического анализа текста.

Все вышеупомянутые шаблоны, хотя и отличаются друг от друга, как отличаются и цели их составления, однако имеют одну общую характеристику: они сопоставляются с естественно-языковым текстом, который на протяжении всей операции сопоставления остается неизменным. В данной работе предлагается метод сопоставления текста с базовыми семантическими шаблонами, в результате чего формируются семантические зависимости, связывающие части анализируемого предложения. Особенность этого алгоритма и отличие от вышеупомянутых способов сопоставления текста с шаблонами заключается в том, что анализируемое предложение, поступающее на вход семантического анализатора, в ходе анализа постепенно сокращается: некоторые части предложения удаляются из последующего анализа при выполнении определенных условий, описанных в базовых семантических шаблонах.

Предлагаемый способ семантического анализа естественно-языкового текста предполагает формирование базовых семантических шаблонов вручную, однако количество подобных шаблонов значительно меньше количества шаблонов, сопоставление с которыми происходит по классическим алгоритмам, не подразумевающим последовательное сокращение анализируемого текста. Вследствие небольшого количества базовых семантических шаблонов работа семантического анализатора значительно ускоряется.

Базовые семантические шаблоны

Семантическое отношение. Семантическое отношение – это некая универсальная связь, усматриваемая носителем языка в тексте. Эта связь бинарна, т.е. она идет от одного семантического узла к другому узлу [9]. В качестве семантических узлов удобно рассматривать неделимые смысловые единицы языка, которые могут быть представлены, например, именованными сущностями. Будем говорить, что два различных семантических узла α и β из одного предложения связывает семантическая зависимость с именем R (обозначим $R(\alpha,\beta)$), если между α и β существует некая универсальная бинарная связь. Для конкретных семантических узлов α, β и зависимости R направление выбирается таким образом, чтобы формула была эквивалентна утверждению, что «β является R для α». В идеале множество семантических зависимостей, используемое при машинном анализе текста, должно покрывать все возможные связи между частями текста [1].

Главным аргументом в семантической связи назовем тот узел, от которого можно задать вопрос ко второму семантическому узлу из рассматриваемой семантической связи. Например, для связи ПРИЗНАК (хижина, ветхая) главным аргументом является слово «хижина», так как можно задать вопрос: «хижина какая?» – «ветхая». Для определенности главный атрибут в семантической связи всегда будем располагать первым.

Базовые семантические шаблоны. Базовым семантическим шаблоном назовем правило, по которому в анализируемом тексте находится семантическая зависимость. Базовый семантический шаблон состоит из 4 основных частей:

1. последовательность слов или неделимых смысловых единиц, для которых указаны их морфологические признаки, а в некоторых случаях, когда это особенно важно для семантического анализа, приведены названия этих слов и смысловых единиц;
2. название семантического отношения, которое должно быть сформировано в случае обнаружения в тексте последовательности, описанной в предыдущем пункте;
3. последовательность чисел, определяющая позиции в последовательности из п. 1, элементы которой должны быть добавлены в очередь с приоритетом, в соответствии с которой впоследствии будут удаляться слова из анализируемого предложения, подаваемого на вход семантическому анализатору;
4. число, обозначающее значение приоритета, группы семантических зависимостей, к которой относится данное семантическое отношение.

Очередь с приоритетом. В классическом определении очередь с приоритетом определяется как абстрактный тип данных, позволяющий хранить пары (ключ, значение) и поддерживающий следующие операции [10]:

- init – инициализация новой пустой очереди;
- insert – добавление нового элемента в очередь;
- remove – удаление и возвращение элемента с наивысшим приоритетом из очереди;
- isEmpty – проверка очереди на наличие в ней элементов.

В рамках настоящей работы «очередь с приоритетом» используется для хранения слов или набора слов, являющихся правым аргументом некоторой семантической связи, найденной в анализируемом предложении. Для определения приоритета элемента в такой очереди используются два значения:

- значение приоритета группы, к которой принадлежит данная семантическая связь;
- позиция слова (или последнего слова из набора, если элемент, хранимый в очереди, состоит из нескольких слов) в анализируемом предложении.

Будем говорить, что элемент из описываемой очереди обладает наивысшим приоритетом, если значение приоритета семантической группы минимально, а значение позиции последнего слова из набора, образующего элемент, максимально. Таким образом, элементы очереди с приоритетом сортируются по возрастанию приоритетов групп семантических зависимостей. Если в очереди нашлось несколько элементов с одинаковыми значениями приоритетов семантических групп, то такие элементы сортируются по убыванию позиции последнего слова, относящегося к рассматриваемому элементу, в анализируемом предложении. Приведем код метода compareTo, реализованный на языке программирования Java и осуществляющий сравнение двух объектов класса *Unit*, каждый из которых имеет свой приоритет (*prioritet*) и позицию в анализируемом предложении (*positionInSentence*):

```
public int compareTo(Unit o)
  {
        int ret = Integer.compare(prioritet, o.prioritet);

        if (ret = = 0)
              ret = Integer.compare(o. positionInSentence, positionInSentence);

        return ret;
  }
```

Ниже описаны правила добавления в очередь Q с приоритетом элемента (α', sp', pos'), где α' – значение элемента; sp' – приоритет семантической группы; pos' – позиция элемента в анализируемом предложении для случая, когда Q содержит элемент (α, sp, pos) такой, что $(\alpha' = \alpha)$ и $(pos' = pos)$:

1. $(sp' > sp)$, следовательно, $Q = Q \setminus (\alpha, sp, pos) \cup (\alpha', sp', pos')$;
2. $(sp' \le sp)$, следовательно, Q не изменяется.

На рис. 1 представлен с пояснениями пример базового семантического шаблона. Здесь последовательность слов представлена тремя составляющими: глаголом (Г) мужского рода (мр), в единственном числе (ед), инфинитивом (ИНФ) и существительным (С) в именительном падеже (им), единственном числе (ед), мужского рода (мр). В случае обнаружения в тексте указанной последовательности, не разделенной другими словами и знаками препинания, формируется семантическая связь ДЕЙСТВИЕ, аргументы в которой располагаются в той последовательности, в которой они указаны в круглых скобках, учитывая, что нумеруются элементы последовательности с нуля (нумерация указана в квадратных скобках над описанием элементов последовательности). Левый и правый аргументы семантической связи в шаблоне разделены знаком «|». После того, как семантическая связь ДЕЙСТВИЕ сформирована, слово из последовательности с номерами 2 ставится в очередь на удаление, организованную в виде очереди с приоритетом, как это было описано выше.

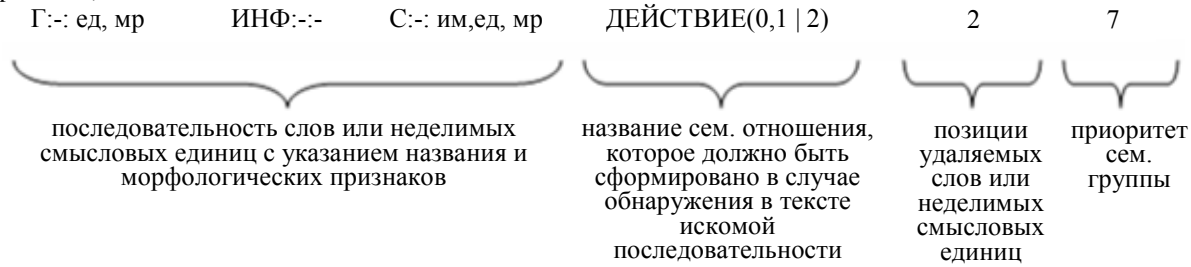

Рис. 1. Пример базового семантического шаблона

Очередь с приоритетом, хранящая части предложения, которые будут удаляться из анализируемого текста T, организована таким образом, что каждый раз после того, как все базовые семантические шаблоны в T найдены, из T удаляются по одному слову или набору слов, образующему правый аргумент некоторой семантической связи. При этом удаляемый из T элемент принадлежит к паре из Q с наивысшим приоритетом.

В данной работе предлагается способ поиска семантических отношений с помощью сопоставления текста анализируемого предложения с набором базовых семантических шаблонов.

Алгоритм нахождения семантических зависимостей с помощью базовых семантических шаблонов

Обозначения, используемые при описании алгоритма:

- T – анализируемое предложение;
- t_i – i -ая неделимая смысловая единица анализируемого предложения T;
- S – множество всех базовых семантических шаблонов;

- s_i – i-й шаблон множества S;
- sp_i – приоритет шаблона s_i;
- $R_i(t_{i_1}, t_{i_2})$ – семантическая зависимость R_i, определяемая шаблоном s_i и связывающая две неделимые смысловые единицы t_{i_1} и t_{i_2};
- $pos(t_i)$ – позиция в анализируемом предложении последнего слова из t_i;
- Q – очередь с приоритетами;
- (t_{i_2}, sp_i, pos_i) – элемент очереди Q, образованный посредством шаблона s_i.

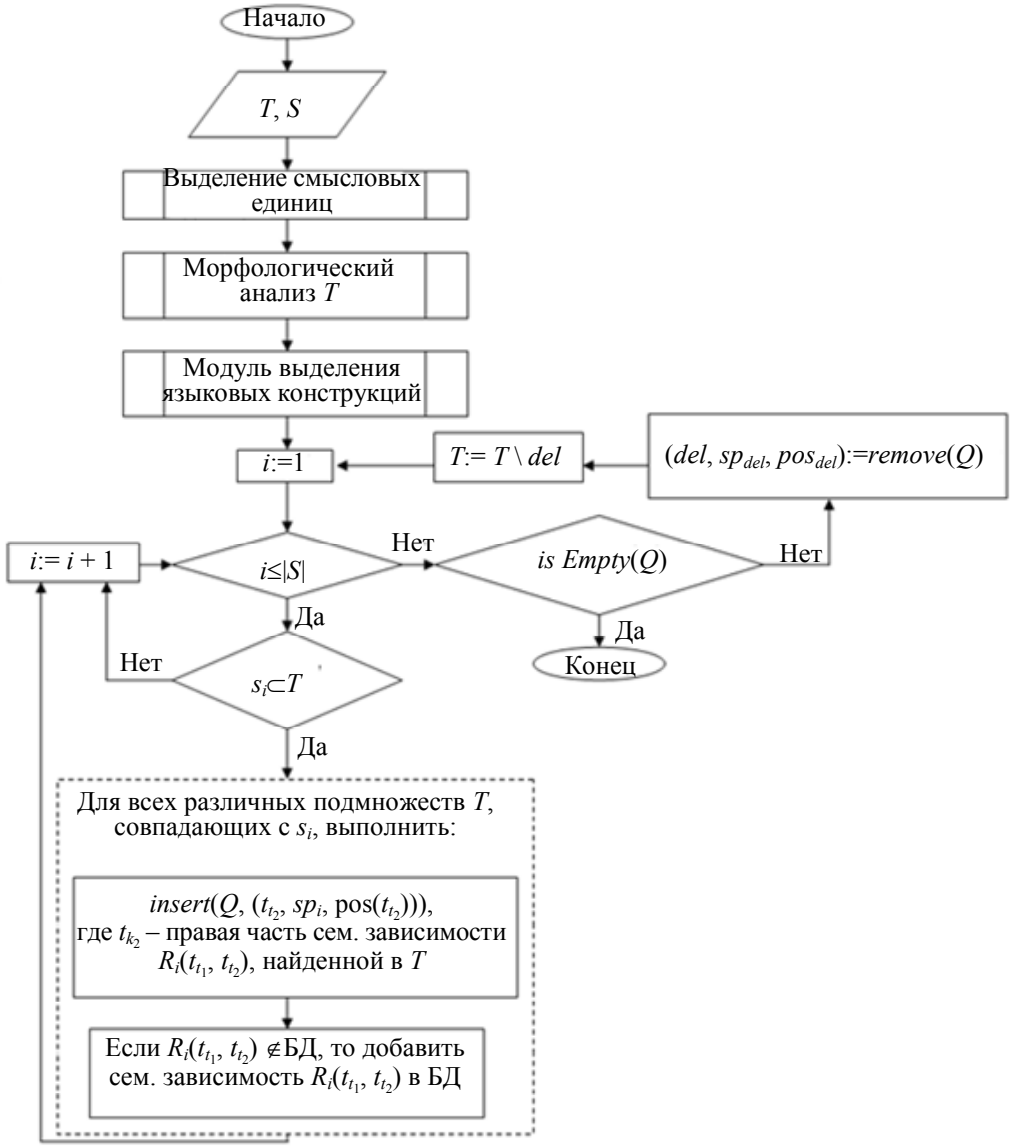

Рис. 2. Блок-схема алгоритма поиска семантических зависимостей

Алгоритм нахождения семантических зависимостей. На вход семантическому анализатору подается предложение T на естественном языке и множество $S = \{s_1, s_2, s_3, ...\}$ базовых семантических шаблонов, где s_i – отдельный семантический шаблон. Предложение T разделяется на неделимые смысловые единицы, обозначаемые $t_1, t_2, t_3, \ ... \ , t_n$, состоящие либо из одного слова, либо из набора слов, который может являться именованной сущностью (например, название государства, название мероприятия, титул человека и т. п.). Таким образом, получается представление T в виде упорядоченного набора t_i, где $i = 1..n$. Набор $t_1, t_2, t_3, \ ... \ , t_n$ упорядочен в том смысле, что при последовательном написание всех t_i, получится анализируемое предложение T, т.е. T можно представить так: $T = t_1 \cup t_2 \cup t_3 \cup \ ... \ \cup t_n$, сохраняя при этом порядок следования t_i.

После формирования набора неделимых смысловых единиц производится морфологический анализ каждой такой единицы. Затем в модуле выделения языковых конструкций происходит поиск таких сложных языковых конструкций, как вводные, причастные и деепричастные обороты, придаточные предложения и т. д.

Далее последовательно происходит поиск совпадений каждого базового семантического шаблона из множества S в множестве T, при этом как для базовых семантических шаблонов, так и для всех t_i учитываются морфологические характеристики.

В случае обнаружения совпадения семантического шаблона s_i, с некоторыми подмножествами множества $T = t_1 \cup t_2 \cup t_3 \cup \ ... \ \cup t_n$ (на блок-схеме алгоритма обозначено как $s_i \subset T$), для всех различных подмножеств T, совпадающих с s_i, формируются семантические зависимости $R_i(t_{i_1}, t_{i_2})$, которые записываются в БД, если они обнаружены в анализируемом тексте впервые. При этом в очередь Q с приоритетом добавляется новый элемент, представленный тройкой $(t_{i_2}, sp_i, pos(t_{t_2}))$, где sp_i – приоритет группы, к которой относится семантическая зависимость R_i. Поиск базовых семантических шаблонов в T происходит до тех пор, пока не будут проверены на совпадение все шаблоны. После окончания поиска в T шаблонов происходит проверка очереди Q с приоритетом на пустоту с помощью функции isEmpty: если она пуста, это означает, что на очередном этапе сопоставления шаблонов с T новых семантических зависимостей не найдено, и программа завершает свою работу. В противном случае посредством функции remove, описанной выше, получаем элемент $(del, sp_{del}, pos_{del})$ из очереди Q с наивысшим приоритетом, после чего значение del удаляется из текущего множества T, представляющего оставшиеся для дальнейшего анализа слова из анализируемого предложения. По определению функции remove после ее вызова происходит удаление элемента $(del, sp_{del}, pos_{del})$ из Q.

Далее повторяется поиск базовых семантических шаблонов $S = \{s_1, s_2, s_3, ...\}$ среди оставшихся неделимых семантических единиц множества T. Так продолжается до тех пор, пока множество Q не станет пустым (это означает, что в анализируемом предложении найдены все возможные семантические зависимости, описанные базовыми семантическими шаблонами S). Блок-схема описанного алгоритма представлена на рис. 2.

Программная реализация семантического анализатора

В ходе работы в соответствии с описанным алгоритмом поиска семантических зависимостей с помощью базовых семантических шаблонов с удалением на языке программирования Java был реализован семантический анализатор русского языка. Для ускорения работы программы была использована экспертная система Drools 6.0 [11], использующая алгоритм быстрого сопоставления с шаблоном PHREAK. На вход программе поступают предложения на русском языке, а на выходе программа предоставляет набор семантических отношений, сформированных по анализируемому тексту.

Для реализации семантического анализатора было построено 2160 базовых семантических шаблонов, определяющих на данный момент 539 семантических зависимостей. Все семантические зависимости разбиты на 17 групп – по значениям приоритетов удаления, значения которых используются для формирования очереди с приоритетом на удаление. Морфологический анализ предложения, предшествующий семантическому, реализован на базе грамматического словаря А.А. Зализняка [12], а морфология неизвестных слов, не найденных в словаре, определяется по алгоритму, предложенному авторами [13] и основанному на статистическом анализе последовательностей последних букв предложения. Морфологическая омонимия в текущей программной реализации частично снимается посредством использования программы *mystem* [14] и специальных правил, учитывающих возможность морфологического согласования различных частей речи, обладающими определенными морфологическими признаками. В таблице приведено несколько примеров таких правил.

В будущем для снятия морфологической омонимии предполагается использовать вероятностную модель, предложенную авторами работы [15], которая основывается на нормализующих подстановках.

Снятие омонимии со слов, имеющих одинаковые морфологические характеристики, но различающиеся по смыслу (например, «детский лагерь» и «лагерь демократов») в текущей программной реализации не реализовано. В будущем данную проблему планируется разрешать посредством интеграции семантического анализатора с онтологией, а также путем анализа контекста слова, которому могут соответствовать несколько смыслов.

Описанный в этой работе алгоритм позволяет устанавливать связь между частями предложения, разделенными такими сложными языковыми конструкциями, как вводные, причастные и деепричастные обороты, придаточные предложения и т. д. Достигается это путем использования соответствующих семантических шаблонов и синтаксического анализатора в модуле выявления языковых конструкций. При

этом аргументами семантических отношений могут быть как сами обороты или придаточные предложения целиком, так слова или словосочетания внутри них.

Правило	Пример морфологической омонимии	Снятие морфологической омонимии
Ближайшее справа от предлога существительное не может находиться в именительном падеже	На (ПРЕДЛ) стол (С:им,ед,мр || С:вн,мн,мр)	На (ПРЕДЛ) стол (С:вн,мн,мр)
Ближайшее справа от предлога «для» существительное может находиться только в родительном падеже	Для (ПРЕДЛ) школы (С:им, мн, жр || С:рд, ед, жр || С:вн,мн,жр)	Для (ПРЕДЛ) школы (С:рд,ед,жр)
Если перед словом X, определенным морфологическим анализатором как существительное или глагол стоит качественное наречие, то слово X является глаголом	Мыла (С:ср,рд,ед || С:ср,им,мн || С:ср,вн,мн || Г:прш,жр,ед) быстро (Н:кач)	Мыла (Г:прш,жр,ед) быстро (Н:кач)

Таблица. «Правила снятия морфологической омонимии»

Ниже приведен пример работы семантического анализатора, на вход которому подается предложение «Отличники школы яхтенного спорта, завоевав переходящий кубок, выехали в лагерь.»:

- ДЕЕПРИЧ_ОБОРОТ (выехали | завоевав переходящий кубок)
- ПРИЗНАК (кубок | переходящий)
- ЧТО (завоевав | кубок)
- ЧЕГО (отличники | школы)
- ЧЕГО (школы | спорта)
- ПРИЗНАК (спорта | яхтенного)
- МЕСТО (выехали | в лагерь)
- ДЕЙСТВИЕ (выехали | отличники).

Заключение

В ходе работы был разработан и программно реализован алгоритм работы семантического анализатора русскоязычного текста, основанный на базовых семантических шаблонах с удалением.

Программная реализация данного алгоритма показала, что при достаточном количестве базовых семантических шаблонов, используемых анализатором, работа программы может быть сопоставима с результатами работы такого известного семантического анализатора русских текстов, как продукт, разработанный группой Aot.ru [16].

Использование в предлагаемом алгоритме базовых семантических шаблонов, подразумевающее постепенное сокращение анализируемого текста, а также применение быстрого алгоритма сопоставления с шаблонами PHREAK обеспечивает высокую скорость работы семантического анализатора. Так, например, описанный в работе алгоритм семантического анализа текста, использующий экспертную систему Drools и 2160 базовых семантических шаблонов, в тексте сказки Э.Т.А. Гофмана «Золотой горшок» определил 8213 семантических отношений за 6930 мс. Предложенный алгоритм семантического анализа без использования экспертной системы Drools работает медленнее в среднем в 6–8 раз. Тестирование проводилось на процессоре Intel Core i3 M 350 CPU 2.27 ГГц в операционной системе Ubuntu 12.04.

Разработанный семантический анализатор используется в качестве составного модуля интеллектуальной вопросно-ответной системы, описанной в работе [17]. В дальнейшем планируется интегрировать данный семантический анализатор с онтологией.

Литература

1. Рабчевский Е.А. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска // Труды XI Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». Петрозаводск, 2009. С. 69–77.
2. Филлмор Ч. Дело о падеже // Новое в зарубежной лингвистике. Вып. X. М.: Прогресс, 1981. С. 369–495.
3. Филлмор Ч. Дело о падеже открывается вновь // Новое в зарубежной лингвистике. Вып. X. М.: Прогресс, 1981. С. 496–530.
4. Чубинидзе К.А. Метод синтактико-семантических шаблонов и его применение в информационной технологии интерпретации текстов: дис… . канд. техн. наук. М., 2006. 156 с.
5. Большаков И.А. Какие словосочетания следует хранить в словарях? // Труды международного семинара Диалог'2002 по компьютерной лингвистике и ее приложениям. Протвино: 2002. Т. 2. С. 61–69.

6. Загорулько Ю.А., Сидорова. Е.А. Система извлечения предметной терминологии из текста на основе лексико-синтаксических шаблонов // Труды XIII Международной конференции «Проблемы управления и моделирования в сложных системах». Самара, 2011. С. 506–511.
7. Hearst M.A. Automatic acquisition of hyponyms from large text corpora // Proc. 14th International Conference on Computational Linguistics, 1992. P. 539–545.
8. Лайонз Дж. Введение в теоретическую лингвистику. М.: Прогресс, 1978. 544 с.
9. Сокирко А.В. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ): дис. … канд. техн. наук. М., 2001. 120 с.
10. Downey A.B. Think Python. O'Reilly Media, 2012. 300 p.
11. Drools Documentation [Электронный ресурс]. Режим доступа: http://docs.jboss.org/drools/release/6.0.1.Final/drools-docs/html_single, свободный. Яз. англ. (дата обращения 25.05.2014).
12. Зализняк А.А. Грамматический словарь русского языка. Словоизменение. М.: Русский язык, 1980. 880 с.
13. Белоногов Г.Г., Зеленков Ю.Г. Алгоритм морфологического анализа русских слов // Вопросы информационной теории и практики. 1985. № 53. С. 62–93.
14. О программе mystem [Электронный ресурс]. Режим доступа: http://api.yandex.ru/mystem, свободный. Яз. рус. (дата обращения 17.03.2014)
15. Зеленков Ю.Г., Сегалович И.В., Титов В.А. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов // Компьютерная лингвистика и интеллектуальные технологии. 2005. С. 188–197.
16. Автоматическая обработка текста [Электронный ресурс]. Режим доступа: http://www.aot.ru, свободный. Яз. рус. (дата обращения 12.05.2014).
17. Мочалова А.В., Мочалов В.А. Интеллектуальная вопросно-ответная система // Информационные технологии. 2011. № 5. С. 6–12.

Мочалова Анастасия Викторовна – соискатель, Петрозаводский государственный университет, 185910, г. Петрозаводск, Россия, stark345@gmail.com

Anastasia V. Mochalova – applicant, Petrozavodsk State University, 185910, Petrozavodsk, Russia, stark345@gmail.com

Принято к печати 03.06.14
Accepted 03.06.14