

На правах рукописи

**УРАЗЛИН ЮРИЙ КЛИМЕНТОВИЧ**



**АНАЛИЗ СЛАБО СТРУКТУРИРОВАННЫХ  
ТЕКСТОВЫХ ДАННЫХ**

**Специальность 05.13.18 – математическое  
моделирование, численные методы и комплексы  
программ**

**АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук**

43

**Москва – 2005**

Работа выполнена на кафедре математических основ  
управления Московского физико-технического института  
(государственного университета)

Научный руководитель:

доктор физ.-мат. наук, профессор  
Флёров Юрий Арсениевич

Официальные оппоненты:

доктор физ.-мат. наук, профессор  
Павловский Юрий Николаевич

доктор физ.-мат. наук, профессор  
Афанасьев Александр Петрович

Ведущая организация:

Московский государственный институт электроники и  
математики (технический университет)

Защита состоится «21» декабря 2005 года в 11<sup>30</sup> час. на  
заседании диссертационного совета К212.156.02 в Москов-  
ском физико-техническом институте по адресу: 141700,  
г. Долгопрудный Московской обл., Институтский пер., д. 9.

С диссертацией можно ознакомиться в библиотеке МФТИ.

Автореферат разослан «17» ноября 2005 г.

Ученый секретарь  
диссертационного совета



Федько О.С.

2006-4  
23185

2221876

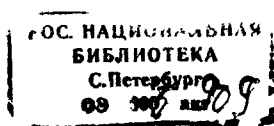
## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

### Актуальность темы

Распространение современных цифровых библиотек и популяризация сети Интернет сделали доступными широкой аудитории огромные объёмы информации в виде компьютерных текстов. В то же время эти хранилища текстовых данных зачастую не используются, а получение требуемой информации из них является достаточно сложной задачей. Проблема заключается в несоответствии модели представления текстовых данных в хранилищах и операций, которые требуется выполнить пользователям с содержащейся в них информацией. Необходимы инструменты, способные анализировать слабо структурированные текстовые данные и моделировать содержащуюся в них информацию с помощью модели, явно фиксирующей логическую структуру. Такая модель информации, представленная, например, в виде записей реляционной базы данных, делает возможной автоматическую обработку в комплексах программ и обеспечивает выполнение требуемых операций.

Традиционно решаются прикладные задачи анализа текстов фиксированного формата, при этом для каждого формата создаётся отдельное приложение. Кроме того, если формат относительно сложен, и его непросто формализовать, пользователям необходимо или часто взаимодействовать с программистами для коррекции алгоритмов работы приложения, или программировать и настраивать алгоритм анализа самостоятельно. Это либо невозможно при отсутствии программистов, либо приводит к существенным трудозатратам.

В работе описан способ создания инструментов анализа, способных решить широкий класс задач анализа текстовых данных. Инструменты автоматически настраиваются на произвольный формат и для анализа документов нового, неизвестного заранее формата не требуют ни программирования, ни специальных действий пользователей.



Таким образом, работа посвящена актуальной задаче анализа документов содержащих слабо структурированные текстовые данные, а предложенные в ней методы и технологии позволяют облегчить её решение, расширяя множество потенциально решаемых прикладных задач.

### **Цель работы**

Целью работы является разработка метода создания инструментов анализа, способных работать с произвольными форматами текстовых данных, и решение прикладной задачи анализа документов, содержащих вопросы.

В работе решены следующие задачи:

1. рассмотрена задача анализа текстов на естественном языке (NLP, natural language processing), выявлены сходства и различия решения этой задачи и задачи анализа слабо структурированных текстовых данных;
2. предложен общий способ создания инструментов анализа текстовых данных, использующий методы NLP, адаптированные для рассматриваемой задачи;
3. решена прикладная задача анализа текстов, содержащих вопросы; в частности разработан и применён метод сравнения возможных способов выполнения этапа графематического анализа; выполнение синтаксического анализа сведено к задаче классификации, реализованы и исследованы два способа эффективного её решения, формирующие правила выполнения анализа методом нисходящей индукции;
4. разработан инструмент, способный после обучения на небольшом количестве примеров выполнить анализ текстов с вопросами в произвольном формате.

### **Научная новизна**

В работе разработана технология создания обучаемых инструментов анализа слабо структурированных текстовых данных. В отличие от систем, которые также используют

примеры для формирования алгоритма анализа, предложенный в работе способ имеет следующие особенности:

1. этапы указания примеров и применения полученного алгоритма анализа объединены, что облегчает обучение программы новому формату, практически устраняя дополнительные действия пользователей;
2. решается задача полного безошибочного анализа текста всего документа, в то время как традиционно анализ выполняется лишь частично, с некоторой точностью;
3. в связи с этим выполнение анализа документов интегрировано с определением ошибок текущего способа анализа, что позволяет автоматически обнаружить ошибки и привлечь к ним внимание пользователей;
4. предложен метод автоматического сравнения адекватности возможных способов реализации графематического анализа решаемой прикладной задаче;
5. на этапе выполнения синтаксического анализа возможно использование произвольной информации о выделяемом логическом элементе информации, а не только текста, который находится в непосредственной близости.

При решении задачи анализа текстов с вопросами:

1. для выполнения синтаксического анализа разработаны два способа решения задачи классификации, формирующие правила анализа методом нисходящей индукции; исследована возможность использования статистических методов;
2. с помощью предложенного в работе метода выбран способ выполнения графематического анализа;
3. разработан метод автоматического определения ошибок анализа, основанный на регулярных грамматиках;
4. экспериментально исследована эффективность предложенного в работе способа выполнения анализа.

### **Практическая ценность**

Разработанные технологии применимы для широкого круга прикладных задач анализа слабо структурированных текстовых данных. Анализируемые документы могут содержать информацию о вопросах, оглавлениях книг, требованиях к программному обеспечению, продаваемых товарах, и т.п. В работе приведено подробное описание нескольких прикладных задач, указаны примеры документов.

Предложенный способ анализа слабо структурированных текстов существенно сокращает сроки анализа авторских документов, содержащих неструктурированное представление информации, которая используется системами дистанционного обучения. Разработанный с его помощью инструмент анализа текстов с вопросами способен в кратчайшее время построить структурированную модель содержащейся в авторских документах информации, и проанализировать тексты с тысячами вопросов. Он использовался при создании программных комплексов «Физика 7-11 классы» и «Биология, химия, экология». В настоящее время системы дистанционного обучения, разработанные компанией ФИЗИКОН, широко используются в процессе обучения, тестирования и самопроверки.

### **Апробация работы**

Основные результаты работы докладывались и обсуждались на научных конференциях МФТИ (Долгопрудный, 2003, 2004, 2005), научных семинарах кафедры математических основ управления МФТИ и Центра сетевых образовательных технологий и систем МФТИ, на международной конференции «Компьютерное моделирование 2005» (Санкт-Петербург, 2005).

### **Публикации**

По теме диссертации опубликовано 8 печатных работ.

### **Структура и объем работы**

Диссертация состоит из введения, пяти глав, заключения, списка использованных источников, содержит 88 иллюстраций. Общий объем работы составляет 144 страницы.

## **СОДЕРЖАНИЕ РАБОТЫ**

Во введении обоснована актуальность темы исследования, описана рассматриваемая проблема, приведён пример прикладной задачи, которая может быть решена с помощью предложенной в работе технологии, сформулировано предложенное решение. Также проведён обзор анализаторов слабо структурированных текстов, выполнено сравнение современных систем анализа с разработанным в работе подходом, указаны отличительные особенности предложенного решения, кратко описано содержание работы.

В главе 1 формализована решаемая задача. Определено, что входом анализа являются наполненные данными и близкие в онтологическом смысле документы, содержащие слабо структурированные текстовые данные, при этом содержащаяся в документах информация может иметь сложную иерархическую структуру. Приведены четыре примера входных данных, а вместе с тем, и прикладных задач, которые могут быть решены с помощью предложенного подхода. Выбран и обоснован формат выхода анализа, им является структурированная модель информации, представленная в виде XML документов специального вида. Определено, что преобразование может быть выполнено полуавтоматически, указаны этапы, которые могут быть автоматизированы, и этапы, которые принципиально невозможно выполнить без участия пользователя.

В главе 2 описан предлагаемый общий способ решения задачи анализа слабо структурированных текстовых данных, то

есть, описаны принципы построения универсального программного модуля, и набор методик, которые должны быть использованы при решении любого класса задач анализа слабо структурированных текстовых данных.

Рассмотрена задача анализа текстов на естественном языке, для демонстрации того, как методы её решения были адаптированы для решения рассматриваемой в работе задачи. Подробно рассмотрены реализации графематического и синтаксического этапов анализа. Указаны сходства и различия задач анализа текстов на естественном языке и рассматриваемой задачи. Так, в NLP этап графематического анализа, имеет относительно простые способы решения. Реализация синтаксического анализа текстов на естественном языке является намного более сложной. Её сводят к задаче классификации объектов на основе набора признаков, что позволяет успешно применить для её решения богатый арсенал областей Искусственного Интеллекта и Обучения Машин. Описаны различные подходы к его решению

1. основанные на индукции правил методы
2. методы, основанные на прецедентах
3. статистические методы
4. комбинированные методы

Указаны преимущества и недостатки перечисленных подходов к выполнению синтаксического анализа.

Далее описан предлагаемый способ анализа. Предложенную процедуру анализа слабо структурированных текстовых данных можно представить следующим образом.

1. Сделать некоторое предположение о том, как внутреннее представление текста соотносится с логической структурой информации, которую содержит текст. На основе этого предположения выбрать известный алгоритм анализа текста
2. Пользуясь алгоритмом, выполнить анализ текста.
3. Выявить ошибки анализа
  - а. Если есть ошибки, то перейти к шагу 4.



- в. Если нет ошибок, то перейти к шагу 6.
4. Определить, как надо правильно анализировать большую часть текста, содержащую ошибки
  5. Скорректировать алгоритм анализа и перейти к шагу 2.
  6. Анализ текста завершен.

Схема процедуры выполнения анализа представлена на следующем рисунке.

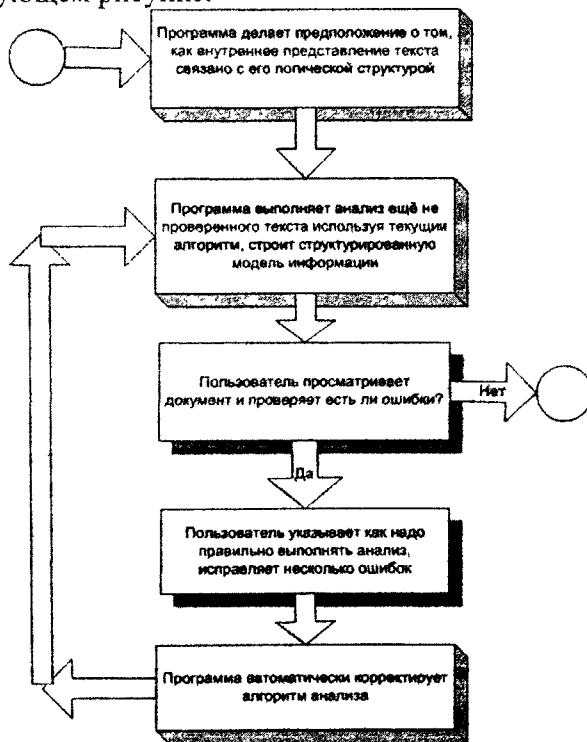


Рисунок 1. Схема процедуры выполнения анализа

С помощью указанной процедуры выполняется полный безошибочный анализ. При этом текст, находящийся в начале документа, и уже проверенный пользователем, используется модулем обучения, как пример правильного выполнения анализа, а сгенерированный алгоритм применяется для ещё не проверенного текста. Наибольшей сложностью обладает шаг

корректировки алгоритма анализа. На этом шаге используются методы и технологии из области Обучения Машин.

Выполнение анализа документа делится на два относительно простых этапа:

1. Графематический
2. Синтаксический

Для реализации этапа графематического анализа используются относительно несложные и эффективные алгоритмы. Однако этот этап является более сложным в сравнении с аналогичным этапом анализа текстов на естественном языке. Сложность заключается в произвольности текстовых данных и, соответственно, произвольности используемого набора графематических типов. Дело в том, что в отличие от задач анализа текстов на естественном языке, рассматриваемая задача имеет дело с различными структурами информации, которые могут задаваться различными способами форматирования, в то время как в задачах анализа текстов на естественном языке структура информации всегда одинакова. Для каждого выделенного класса прикладных задач необходимо использовать специальный набор графематических типов лексем, эффективный при решении рассматриваемого класса задач. Поэтому для решения произвольного класса задач анализа текстовых данных предложен статистический способ сравнения возможных способов выполнения этапа графематического анализа.

Предположим, заданы два набора графематических типов:  $\{T1_i\}$  и  $\{T2_i\}$  и необходимо определить, какой набор эффективнее для решения некоторого класса задач. Сравнение эффективности использования различных наборов графематических типов выполняется с помощью понятия энтропии, и основывается на следующем предположении. Чем «больше порядка», при прочих равных условиях, в полученных в результате графематического анализа последовательностях лексем, тем проще выявить закономерности в текстовых данных на последующем этапе синтаксического анализа. Для сравне-

ния используется образцовый набор текстов, который содержит особенности текстов выбранной прикладной задачи. Для сравнения наборов необходимо выполнить графематический анализ образцовых текстов с помощью каждого сравниваемого набора графематических типов. В результате анализа получаются две последовательности лексем  $\{L1_i\}$  и  $\{L2_i\}$ . Пусть этим последовательностям лексем соответствуют последовательности типов лексем:  $\{LT1_i\}$  и  $\{LT2_i\}$ . С помощью следующей формулы можно определить условную энтропию в последовательностях типов лексем.

$$\begin{aligned}
 H_n &= - \sum_{i,j} p(b_i, j) * \log_2 p_{b_i}(j) \\
 &= - \sum_{i,j} p(b_i, j) * \log_2 p(b_i, j) + \sum_{i,j} p(b_i, j) * \log_2 p(b_i) \\
 &= - \sum_{i,j} p(b_i, j) * \log_2 p(b_i, j) + \sum_i p(b_i) * \log_2 p(b_i)
 \end{aligned}$$

Рисунок 2. Формула расчёта энтропии последовательности типов лексем

В этой формуле

- $b_i$  – это блок из  $n-1$  типа,  $j$  – произвольный тип, который следует за  $b_i$
- $p(b_{i,j})$  – вероятность последовательности  $b_i, j$
- $p_{b_i}(j)$  – условная вероятность типа  $j$  при условии  $b_i$ , то есть  $p(b_{i,j})/p(b_i)$

Значения энтропии необходимо сравнивать для значений  $n$  порядка размера используемого при синтаксическом анализе контекста лексемы. Далее в работе продемонстрировано применение предложенного способа сравнения, и приведены эксперименты, подтверждающие его адекватность.

Этап синтаксического анализа является самым сложным, поскольку правила его выполнения трудно формализуемы из-за того, что в каждом выделенном классе прикладных задач существует много различных форматов текстовых до-

кументов. Алгоритм анализа документов выделенного класса задач можно описать лишь частично, и даже для этого необходимы сотни правил, и исключений из этих правил. В силу произвольности текстовых данных правила выполнения анализа не могут быть известны заранее, их можно определить лишь непосредственно во время выполнения анализа. Поэтому этап синтаксического анализа содержит модуль обучения, который изучает примеры правильного выделения информации, и автоматически формирует алгоритм выполнения синтаксического анализа текста. Выполнение синтаксического анализа сведено к задачам классификации, как это делается в анализе текстов на естественном языке. Так объединение лексем в непересекающиеся группы можно задать, разбив лексемы на классы, то есть, решив задачу классификации. Например, с помощью классов следующим образом можно задать группы лексем.

(NP You) (VP will start to see) (NP shows) (ADVP where)  
 (NP viewers) (VP program) (NP the program).

|         |        |
|---------|--------|
| You     | B-NP   |
| will    | B-VP   |
| start   | I-VP   |
| to      | I-VP   |
| see     | I-VP   |
| shows   | B-NP   |
| where   | B-ADVP |
| viewers | B-NP   |
| program | B-VP   |
| the     | B-NP   |
| program | I-NP   |
|         | O      |

Рисунок 3. Выделение групп лексем с помощью разбиения на классы

То есть объединение лексем в группы эквивалентно отнесению лексем к элементу заранее заданного множества классов.

Выделение групп лексем, в виде задачи классификации представлено на следующем рисунке.

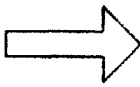
| Вход: объекты и свойства |     | Выход: синтаксические классы  |        |
|--------------------------|-----|---|--------|
| You                      | PRP |   | B-NP   |
| will                     | MD  |   | B-VP   |
| start                    | VB  |   | I-VP   |
| to                       | TO  |   | I-VP   |
| see                      | VB  |   | I-VP   |
| shows                    | NNS |  | B-NP   |
| where                    | WRB |   | B-ADVP |
| viewers                  | NNS |   | B-NP   |
| program                  | VBP |   | B-VP   |
| the                      | DT  |   | B-NP   |
| program                  | NN  |   | I-NP   |
|                          |     |   | O      |

Рисунок 4. Выделение групп лексем как классификация объектов

Для выполнения синтаксического анализа необходимо решить одну, или несколько таких задач классификации. Более одной задачи классификации необходимо решать в случае, когда используется сложная структура информации. Количество решаемых задач равно максимальной глубине вложенности логической структуры выделяемой информации.

Классификация объектов реализована с помощью универсального программного модуля, который решает абстрактную задачу классификации и не специализируется на анализе текстов. При этом задача классификации может быть сформулирована следующим образом.

- задан объект, обладающий набором свойств;
- заданы примеры правильного выполнения классификации, то есть объекты, обладающие аналогичными свойствами, и правильно определённые их классы;

- необходимо проанализировать примеры, на их основе составить алгоритм классификации, и с его помощью определить класс рассматриваемого объекта.

В программе способ решения задачи классификации представлен в виде абстрактного типа, показанного на следующем рисунке.

```
public interface Classifier
{
    //1. this is used to sort properly classified samples
    //this method is invoked for each sample
    void learn(Case c, Object clazz) throws Exception;

    //2 after all samples are learned this method is invoked
    //to deduct classification algorithm
    void train() throws Exception;

    //3 then this method is invoked to provide cases to be classified
    void setCaseSource(CaseSource caseSource);
    //4 and finally this one to set predicted class of every case
    //in CaseSource
    void predict() throws Exception;
}
```

Рисунок 5. Представление произвольного способа классификации в программе

Описаны исследованные при решении задачи анализа текстов с вопросами способы классификации:

- Модель максимальной энтропии
- Способы, основанные на индукции правил

Проведён обзор принципов работы этих методов, главные преимущества и недостатки.

Кроме непосредственного выполнения анализа текста, возможно также автоматическое выполнение проверки результатов анализа. То есть, программа может автоматически определять, правильно ли выполнен анализ текста, и указывать ошибки анализа, способ исправления которых всё же

должен задать пользователь. Проверка правильности выделения информации заключается в определении того, удовлетворяет ли некоторым ограничениям построенная структурированная модель информации, содержащейся в тексте. Ограничения, которым должна удовлетворять структурированная модель, можно разделить на две группы:

- определяемые прикладной задачей;
- специфические для конкретного текста.

В предложенном решении реализован способ обнаружения ограничений второго рода, заданных в виде регулярных грамматик. Этим ограничениям должны удовлетворять последовательности типов логических единиц информации, построенной структурированной модели.

В главе 3 показано, как предложенный общий способ может быть применён для решения прикладной задачи анализа текстов, содержащих вопросы.

Для выполнения графематического этапа анализа, с помощью предложенного в работе метода сравнения выбран следующий набор графематических типов.

| Название                              | Регулярное выражение               |
|---------------------------------------|------------------------------------|
| Перенос строки                        | <code>\r \n \r\n</code>            |
| Открывающий тег                       | <code>&lt;\\w+[^&gt;]*&gt;</code>  |
| Закрывающий тег                       | <code>&lt;\\/w+[^&gt;]*&gt;</code> |
| Пробел                                | <code>\\s+</code>                  |
| Число                                 | <code>\\d+</code>                  |
| Слово                                 | <code>\\w+</code>                  |
| Пунктуационный знак конца предложений | <code>[.,!;\\?]</code>             |
| Пунктуационный знак                   | <code>[.,-;:]</code>               |
| Другое                                | <code>[^\\s]</code>                |

Рисунок 6. Набор графематических типов для анализа документов с вопросами

Кроме этих типов в результате анализа многих документов с вопросами были выделены ещё 5 дополнительных типов, которые часто участвуют в выделении логических элементов информации этой прикладной задачи и, безусловно, облегчают выполнение последующих этапов анализа.

| Название         | Регулярное выражение             |
|------------------|----------------------------------|
| Число, точка     | $\backslash d + \backslash .$    |
| Буква в скобках  | $\backslash ( \backslash w )$    |
| Буква, точка     | $[ a - h , A - H ] \backslash .$ |
| Заглавная буква  | $[ A - H ] \backslash s +$       |
| Символ звёздочки | $[ \backslash * ] +$             |

Рисунок 7. Дополнительные графематические типы

На этапе синтаксического анализа решаются две задачи классификации:

- Строк при определении границ вопросов
- Лексем при выделении атрибутов вопросов

Для этих задач построены два способа решения, основанные на индукции правил. Оба способа построения алгоритма работают по тому же принципу, что и описанная в работе система ALLiS: сгенерировать – проверить – уточнить – проверить – уточнить, и т. д. Также исследована возможность применения статистических методов классификации, экспериментально определено, что Модель Максимальной Энтропии, неприменима для этой прикладной задачи.

Найденный способ построения алгоритма классификации при определении границ вопросов работает следующим образом. Пусть задан набор классифицированных объектов образцов, при этом

- список свойств объектов -  $\{P_j\}$ ,  $j=1 \dots j_0$
- множество наборов контекстов объектов -  $\{b_i\}$ ,  $i=1 \dots i_0$
- каждый контекст  $b_i$  - это набор значений свойств -  $\{V_{ij}\}$
- присвоенные объектам классы -  $\{C_i\}$

Способ построения алгоритма:

1. Выбирается первый образец - первая строка.  $i=1$
2. Выбирается первое по важности свойство из списка  $P_j$  ( $j=1$ ), фиксируется его значение -  $V_{ij}$ .
3. Пусть класс, присвоенный строке -  $C_i$ .
4. Строится правило:

$$b \in (B_{ij} = \{b: P_m = V_{im}, m=1 \dots j\}) \Rightarrow C_i$$

5. Правило проверяется на наборе образцов.



- а. Если найдётся такой объект  $\{b_k, C_k\}$ , что  $V_{km}=V_{im}$ ,  $m=1...j$ , но  $C_k \neq C_i$ , то правило объявляется неправильным. Правило уточняется:  $j=j+1$ . Алгоритм переходит к пункту 3.
- б. Если такого объекта нет, то правило считается верным, и добавляется в список сгенерированных правил. Все объекты, которые ему удовлетворяют, исключаются из рассмотрения. Алгоритм переходит к следующей строке ( $i=i+1$ ), к пункту 2.

Пространство и способ поиска, используемые в предложенном способе построения алгоритма, можно представить с помощью следующего рисунка. Поиск начинается с первого свойства и последовательно проходит свойства, упорядоченные по важности, то есть уровни не могут быть пропущены.

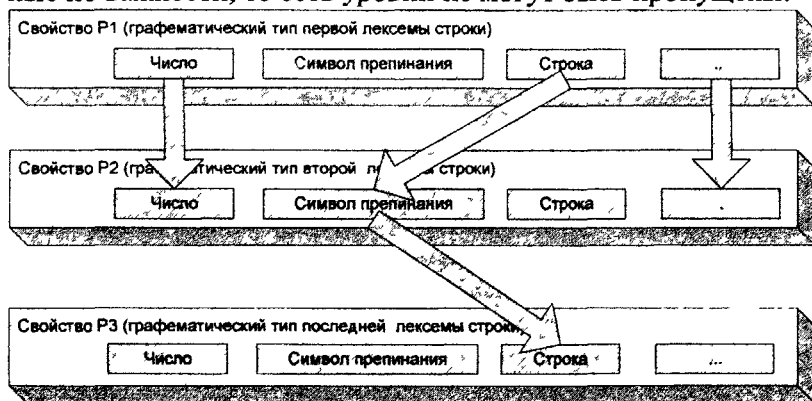


Рисунок 8. Пространство и способ поиска правил при выделении вопросов

Для классификации необходимо использовать набор свойств, составленный из графематических типов первых и последних лексем рассматриваемой, и соседних с ней строк. Определено, что сложность этого алгоритма -  $O(j_0 * i_0^2)$ . Оценено время, необходимое для построения алгоритма, из чего следует, что для устойчивости алгоритма программа может использовать в качестве примеров правильной классификации также объек-

ты, классификацию которых пользователь явно не указывал, но, исправив ошибки анализа ниже в тексте, подтвердил правильность выполнения их анализа.

Найденный способ построения алгоритма классификации лексем при выделении атрибутов вопросов отличается способом исследования пространства правил, поиск по-прежнему начинается на верхних уровнях и перемещается вниз, что позволяет задать вероятно, более важные свойства. Однако в этом случае свойства нижнего уровня могут быть важнее, поиск не обязательно должен начинаться на самом верхнем уровне, и может пропускать уровни.

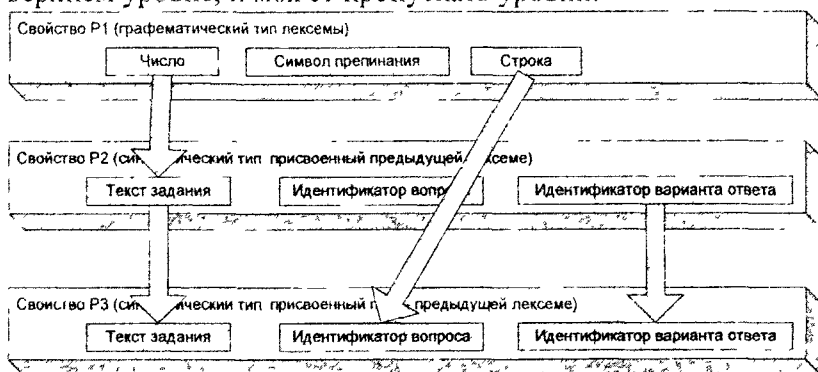


Рисунок 9. Пространство и способ поиска правил выделения атрибутов вопросов

Для классификации используется набор свойств, составленный из графематических типов лексемы и её контекста, синтаксических типов лексем, предшествующих рассматриваемой, номера строки, на которой расположена лексема в вопросе, и текста лексемы.

Найденный способ составления правил:

1. Фиксируется первый образец,  $i=1$ .
2. Выполняется попытка построить самые общие правила, используя только одно свойство объекта,  $n=1$ .
3. Выбирается подмножество свойств объекта  $Conditions = \{V_{imk}\}$ , из множества свойств образца такое, что

$\| \text{Conditions} \| = n$ . Здесь  $m_k$  – набор индексов, такой, что  $m_k < j_0$ , а  $\| m_k \| = n$ .

- a. Подмножества для каждого фиксированного значения  $n$  и  $i$  перебираются последовательно.
  - b. Если все возможные подмножества уже перебраны, то правило утоняется,  $n=n+1$ . Если  $n=j_0$ , то алгоритм переходит к следующей лексеме ( $i=i+1$ ), к шагу 2.
4. Проверяется на противоречивость правило:

$$b \in (B_{ij} = \{b: P_{mk} = V_{imk}\} \Rightarrow C_i$$

- a. Если найдётся такой объект  $\{b_p, C_p\}$ , что  $V_{pmk} = V_{imk}$ , для всех  $m_k$ , используемых в Conditions, но  $C_p \neq C_i$ , то правило объявляется неверным. Алгоритм переходит к шагу 3 (рассматривается следующее подмножество, или правило уточняется).
- b. Если такого объекта нет, правило считается правильным, оно добавляется в список сгенерированных правил, а все объекты, которые ему удовлетворяют, исключаются из рассмотрения. Алгоритм переходит к следующему образцу ( $i=i+1$ ), к шагу 2.

Сложность обучения программы алгоритму выделения атрибутов вопросов можно оценить сверху следующим образом

$$O\left(i_0 \sum_{k=1}^{j_0} C_{j_0}^k\right). \text{ Потому для разумной производительности не-}$$

обходимо при обучении использовать лишь явно указанные примеры. Неявно заданные примеры выполнения классификации, которые указывает пользователь, исправляя ошибки анализа в тексте после них, учитывать нельзя, поскольку это приведёт к недопустимым потерям производительности.

Разработан метод автоматического определения ошибок анализа, основанный на генерировании регулярной грамматики, которой должны удовлетворять правильно выделенные в тексте вопросы. На следующем рисунке представлен пример такой грамматики.

(номер вопроса) (текст вопроса)  
 (([1] варианта) (текст варианта))+  
 (маркер правильного ответа) (правильный ответ)  
 (маркер подсказки) (подсказка)  
 ((название доп. атрибута) (значение доп. атрибута))+

Рисунок 10. Регулярная грамматика атрибутов вопроса

Определив, удовлетворяют ли ещё не проверенные вопросы автоматически сгенерированной грамматике, можно определить, правильно ли они были проанализированы, должен ли пользователь внести исправления в их анализ. К автоматически обнаруженным ошибкам анализа программа привлекает внимание пользователя.

Глава 4 представляет разработанный в рамках работы инструмент анализа документов с вопросами. Все приведённые в работе технологии и алгоритмы были реализованы и экспериментально исследованы с помощью этого инструмента.

Подробно описано взаимодействие пользователя с инструментом, из которого следует, что предложенный способ анализа текста не требует от пользователей каких-либо дополнительных действий: пользователь всего лишь размечает текст, как это он делал бы без помощи модуля обучения, а программа анализирует его действия и пытается их повторить. Кроме того, программа может автоматически указать возможные ошибки в предложенном совместно пользователем и программой варианте анализа.

Глава 5 содержит описание экспериментов, иллюстрирующих эффективность предложенного в работе решения. Так, например, для анализа документа относительно простого формата необходимо указать лишь один пример полного разбора вопроса. Далее в документе с 200 вопросами требуется исправить лишь 15 ошибок алгоритма анализа, которые были автоматически идентифицированы. Для анализа документа

другого формата, в котором чётко выполнялись соглашения о форматировании, для выделения информации о более чем 100 вопросах понадобилось указать один пример полного разбора, и затем исправить лишь одну ошибку.

Также приведён пример действительно сложного формата документа. Поскольку формат сложный и существенно изменяется от вопроса к вопросу, для правильного анализа текста, содержащего 100 вопросов, после указания примера полного анализа первого вопроса, для правильного анализа всего документа необходимо внести 43 исправления. Несмотря на относительно большое количество исправлений, именно при анализе подобных документов созданный инструмент имеет наибольшую ценность, поскольку полуавтоматически составленный алгоритм обладает существенной сложностью. Традиционное кодирование экспертом указанной с помощью примеров связи между разметкой и логической структурой информации весьма сложно, и вряд ли осуществимо за разумное время.

В заключении приведены основные результаты диссертационной работы.

### **Основные результаты работы:**

1. Разработана технология создания обучаемых инструментов анализа слабо структурированных текстовых данных. Такие инструменты способны работать с произвольными форматами текстов и строить структурированную модель информации, содержащейся в них.
2. Разработан статистический метод сравнения возможных способов выполнения этапа графематического анализа.
3. Предложен способ выполнения этапа синтаксического анализа с помощью решения задач классификации, аналогичный его выполнению в задачах анализа текстов на естественном языке. При использовании такой

модели задача синтаксического анализа решается как определение класса объекта на основе его свойств.

4. Предложен способ автоматического выявления ошибок анализа, основанный на регулярных грамматиках.
5. Решена прикладная задача анализа текстов документов с вопросами. Разработан инструмент, позволяющий выполнять анализ документов, содержащих сотни вопросов с помощью указания единственного примера полного анализа, и устранения нескольких ошибок, автоматически выявленных программой. Приведены результаты экспериментов, наглядно иллюстрирующие эффективность его использования в комплексе программ систем дистанционного образования.
6. Для реализации этапа синтаксического анализа документов с вопросами разработаны и исследованы два эффективных способа составления алгоритма классификации, формирующих правила выполнения анализа методом нисходящей индукции.

По теме диссертации опубликованы следующие работы:

1. Уразлин Ю.К. Применение прецедентов для построения систем поиска в неструктурированных данных. // Современные проблемы фундаментальных и прикладных наук. Часть VII. Прикладная математика и экономика: Труды XLV научной конференции. /Моск.физ. – техн. ин-т. – М. - Долгопрудный, 2002. - С. 65.

2. Уразлин Ю.К. Анализ слабо структурированных текстовых данных. // Моделирование и обработка информации: сб.ст. /Моск.физ.-тех. ин-т. – М., 2003. – С. 108-118.

3. Уразлин Ю.К. Анализ слабо структурированных текстовых данных. //Современные проблемы фундаментальных и прикладных наук. Часть VII. Прикладная математика и экономика: Труды XLVI научной конференции. /Моск. физ. – техн. ин-т. – М. – Долгопрудный, 2003. – С. 62-64.

4.Уразлин Ю.К. Выделение групп лексем при анализе слабо структурированных текстовых данных. // Моделирование процессов управления: сб.ст./Моск.физ.-тех. ин-т. – М., 2004. – С. 95-105.

5.Уразлин Ю.К. Автоматическое выявление ошибок при анализе слабо структурированных текстовых данных. //Современные проблемы фундаментальных и прикладных наук. Часть VII. Прикладная математика и экономика: Труды XLVII научной конференции. /Моск. физ. – техн. ин-т. – М. – Долгопрудный, 2004. – С. 105-106.

6.Уразлин Ю.К. Анализ слабо структурированных текстовых данных в системах дистанционного образования. // Процессы и методы обработки информации: Сб.ст./Моск.физ.-тех. ин-т. – М., 2005. – С. 150-157.

7.Уразлин Ю.К. Автоматизация формирования наборов тестов в системах дистанционного образования. // Компьютерное моделирование 2005: Труды VI международной конференции. / СПб: Издательство Санкт-Петербургского государственного политехнического университета, 2005. - с. 566-573.

8.Уразлин Ю.К. Использование методов обработки текстов на естественном языке для анализа слабо структурированных текстовых данных. //Современные проблемы фундаментальных и прикладных наук. Часть VII. Прикладная математика и экономика: Труды XLVIII научной конференции. /Моск. физ. – техн. ин-т. – М. – Долгопрудный, 2005. –С. 84-85.

**05 - 2 2 7 0 9**

**РНБ Русский фонд**

**2006-4**

**23185**

**Уразлин Юрий Климентович**

**Анализ слабо структурированных текстовых данных**

**Автореферат**

**Подписано в печать 14.11.2005.**

**Усл. печ. л. 1.5. Тираж 80 экз. Заказ № 435.**

**Московский физико-технический институт  
(государственный университет)**

**Печать на аппаратуре Rex-Rotary Copy Printer 1280.**

---

**141700, Московская обл., г. Долгопрудный, Институтский пер., 9**