# Извлечение информации из разноструктурированных данных и ее приведение к целевой схеме

© Д. О. Брюхов © С. А. Ступников © Л. А. Калиниченко © А. Е. Вовченко
Институт проблем информатики ФИЦ ИУ РАН,
Москва
dbriukhov@ipiran.ru sstupnikov@ipiran.ru leonidandk@gmail.com
alexey.vovchenko@gmail.com

## Аннотация

Согласно 4-й парадигме, сформулированной Джимом Греем в 2007 г., одной из основных движущих сил развития науки в настоящее время являются данные, получаемые в результате наблюдений, измерений различными высокотехнологичными инструментами, а также накапливаемые в процессе деятельности людей в экономике, промышленности, социальной сфере, и пр. Собственно научное знание образуется в процессе интенсивного анализа накапливаемых данных, приводящего в конечном счете к извлечению знаний из данных. Рост объема и разнообразия данных в различных областях с интенсивным использованием данных приводит к развитию методов и средств анализа массивных, разноструктурированных данных и управления ими. В настоящей статье суммирован опыт, накопленный в процессе применения методов, средств программирования и инфраструктур, предназначенных для извлечения и интеграции информации из разноструктурированных данных, соответствующей потребностям конкретных задач, выраженных целевой структурированной схемой.

## 1 Введение

Интенсивное использование данных становится доминирующим фактором практически во всех научных областях, экономике и бизнесе, государственных организациях [24]. Данные могут быть структурированными (например, логи информационных систем, данные с сенсоров), частично структурированными (например, данные из социальных сетей), неструктурированными (тексты). Извлечение структурированной информации из разноструктурированных данных и ее интеграция становится одной из самых важных задач, стоящих перед разработчиками информационных систем. Это обусловлено тем, что существующие средства анализа данных, например, анализ многомерных кубов (OLAP) или интеллектуальный анализ данных (data mining), оперируют только структурированной информацией.

Особенно остро проблема извлечения информации встает при работе с неструктурированными текстовыми данными, содержащимися в публикациях электронных СМИ, новостных лентах, веб-страницах, записях в социальных сетях, коротких сообщениях микроблогов, электронных письмах, отчетах. Данные такого рода содержат полезную информацию - сущности, взаимосвязи между сущностями, факты, тональность текста (sentiments).

Целью данной работы является суммирование опыта исследования и применения различных методов и средств извлечения информации из разноструктурированных данных и ее приведения к целевой схеме. *Целевой схемой* называется схема структуры данных, в которой должна быть представлена необходимая для решения задачи информация, извлеченная из различных неоднородных информационных ресурсов.

Статья является логическим развитием работы [6], в которой была предложена архитектура комбинированной виртуально-материализованной среды интеграции неоднородных коллекций разноструктурированных данных. Материализованную интеграцию предлагалось реализовать с использованием свободно распространяемой платформы распределенного хранения и обработки данных Hadoop [21]; а также системы организации реляционных хранилищ данных над Hadoop. Были проиллюстрированы методы и средства преодоления *модельной неоднородности* данных: данные могут быть представлены в различных нетрадиционных

**Труды XVII Международной конференции DAMDID/RCDL'2015 «Аналитика и управление данными в областях с интенсивным использованием данных», Обнинск, 13-16 октября 2015**

моделях (например, графовых или RDF), и для последующей интеграции необходимо их приведение к единой (реляционной) модели.

В данной статье внимание сосредоточено на извлечении информации из неструктурированных данных и проблеме преодоления неоднородности на более низком уровне – уровне *схем данных*. Данные из различных информационных ресурсов могут быть представлены в различных схемах. Для финального этапа интеграции (разрешения и слияния сущностей [4]) данные должны быть приведены к единой (целевой) схеме.

Статья организована следующим образом. В разделе 2 рассматриваются основные идеи процесса извлечения информации из разноструктурированных данных и ее приведения к целевой схеме. В разделе 3 рассматривается вариант архитектуры программных средств для реализации предлагаемого подхода. В разделе 4 процесс извлечения информации и ее трансформации иллюстрируется на примере задачи социально-политического мониторинга СМИ и социальных сетей. В разделе 5 представлен краткий обзор родственных работ по сопоставлению схем и трансформации данных. В заключении кратко охарактеризован опыт, полученный в ходе работы.

## 2 Процесс извлечения информации из разноструктурированных данных и ее приведения к целевой схеме

На рис. 1 изображены основные этапы процесса извлечения сущностей из исходных разноструктурированных коллекций данных, их интеграции для последующего анализа полученной информации для анализа при решении прикладной задачи (класса задач).

Процесс начинается с *поиска информационных ресурсов*, релевантных задаче, и *извлечения* из них *исходных коллекций данных*. Информационные ресурсы могут содержать структурированные данные (базы данных, представленные в различных моделях [6]), слабоструктурированные данные (например, данные из социальных сетей), неструктурированные данные (тексты).

На следующем этапе неструктурированные данные (тексты) пропускаются через *средства анализа текстов* [2], например, Pullenti [26], Метафраз [27], AQL [7], SystemT [18]. При этом из текстов *извлекаются сущности*, например, персоны, организации, территориальные образования и т.д. Сущности, извлекаемые из текстов одним конкретным инструментальным средством, всегда соответствуют одинаковой структуре, определяемой выходным форматом средств анализа текстов. Такая структура называется *исходной схемой данных* (см. примеры в разделе 4). В случае, если для анализа текстов необходимо одновременное применение нескольких различных средств (например, если эти средства обладают разными возможностями, извлекают различные сущности), то исходная схема будет представлять собой объединение структур, определяемых выходными форматами этих средств.


Рис. 1. Этапы процесса извлечения информации из разноструктурированных данных и ее интеграции для последующего анализа

Для структурированных данных (извлекаемых из баз данных) исходной схемой является схема базы данных. Источником структурированных данных также может быть онтология (для конкретного класса задач может быть использована готовая онтология или собранная экспертами вручную на основании слабоструктурированной информации из Веб), в этом случае исходной схемой является T-box онтологии.

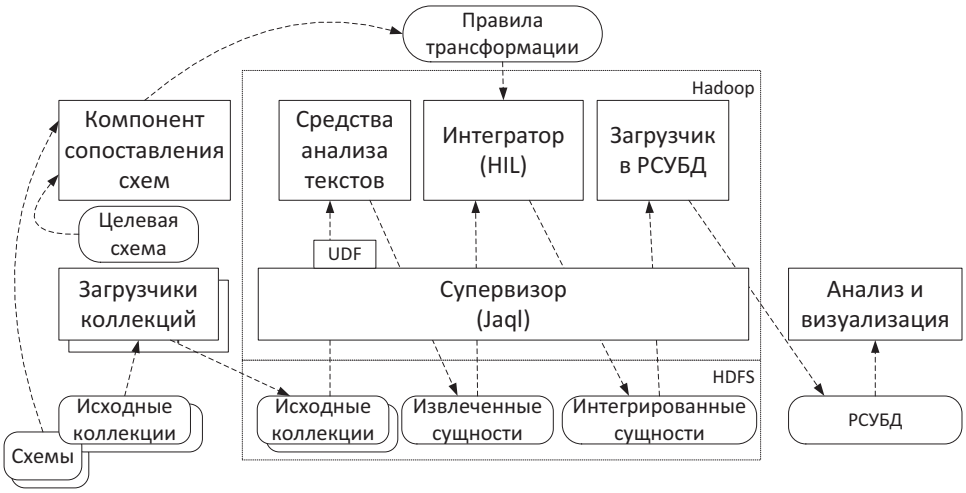



Рис. 2. Архитектура средств извлечения информации из разноструктурированных данных, ее интеграции и анализа

Для слабоструктурированных данных (например, сообщений из социальных сетей) исходная схема образуется путем объединения схемы структурированной части данных (например, профиля пользователя; кем, кому и когда было послано сообщение) и структуры информации, извлеченной из неструктурированной части (текста сообщения).

Следующим этапом является *интеграция собранных структурированных коллекций, онтологий и извлеченных сущностей из текстов в общую интегрированную коллекцию*. Интеграция информации включает несколько подэтапов.

*Сопоставление элементов* исходных схем и целевой схемы может производиться с применением различных методов и средств автоматизации (разделы 4, 5). На основании сопоставления элементов схем создаются правила преобразования данных из исходных схем в целевую (разделы 4, 5). Затем эти правила применяются для *трансформации данных*. К коллекциям данных, представленных в единой целевой схеме, применяются более тонкие методы интеграции. Для установления связей между сущностями из различных коллекций применяются методы *разрешения сущностей*, т.е. установления сходства сущностей (в том числе, обнаружение дубликатов [4]). *Слиянием данных* называется образование интегрированного представления информации об одной же сущности реального мира, полученной из различных источников данных. Слияние данных включает слияние кортежей из разных коллекций, соответствующих одной сущности, разрешение возможных конфликтов, обнаружение и удаление ошибочных данных [4].

Наконец, завершающим этапом процесса является анализ и визуализация информации интегрированной коллекции с помощью существующих средств анализа данных, например, анализа многомерных кубов или интеллектуального анализа данных.

Для обеспечения масштабирования рассматриваемого процесса по объему извлекаемых и интегрируемых данных, его необходимо реализовать на основе некоторой платформы распределенного хранения и обработки больших объемов данных. В соответствии с архитектурой среды материализованной интеграции информационных ресурсов (предложенной в работе [6]), в данной работе в качестве такой платформы рассматривается Apache Hadoop [21]. Hadoop включает, в частности, распределенную файловую систему HDFS и менеджер ресурсов YARN, позволяющий разворачивать над кластерами системы, реализующие различные программные модели параллельных распределенных вычислений (например, MapReduce [19]). В частности, при реализации такого подхода необходимо позаботиться о встраивании методов текстовой аналитики и интеграции данных в Hadoop.

В различных дистрибутивах Hadoop встроены некоторые средства анализа текстов. Так, например, в дистрибутив IBM BigInsights встроен язык разработки экстракторов текстовой аналитики AQL [7]. Если же средства текстовой аналитики разрабатывались независимо от кластерной платформы [26-27], то для их встраивания в Hadoop необходимо: обеспечить их работу под ОС Linux (например, с помощью среды Mono [20]) и разместить эти средства на каждом компьютере Hadoop-кластера для обеспечения параллельной обработки исходных коллекций данных.

Для обеспечения реализации методов интеграции данных над Hadoop (включая разрешение и слияние сущностей), в соответствии с архитектурой [6], в данной работе используется декларативный язык HIL [14], ориентированный на разрешение и слияние сущностей в Hadoop-инфраструктуре HIL может использоваться для спецификации правил трансформации данных из исходных схем в целевую, с дальнейшим выполнением этих трансформаций в среде Hadoop. HIL компилируется в язык Jaql [8], который, в свою

очередь, автоматически переписывается в MapReduce-программы.

## 3 Архитектура средств извлечения информации из разноструктурированных данных, ее интеграции и анализа

На рис. 2 изображен вариант архитектуры, реализующей процесс извлечения информации из разноструктурированных данных и ее приведения к целевой схеме, рассмотренный в предыдущем разделе. Данная архитектура является развитием и уточнением архитектуры материализованной интеграции информационных ресурсов, представленной в работе [6].

Архитектура разворачивается на вычислительном кластере, на котором установлен дистрибутив Hadoop, поддерживающий языки Jaql и HIL (IBM BigInsights [16]).

В рамках архитектуры предполагается, что коллекции разноструктурированных данных, релевантных задаче, уже отобраны экспертами. Также предполагается, что под задачу создана целевая схема (в реляционной модели данных), в которую должны отображаться исходные коллекции.

Исходные коллекции могут быть сгруппированы по модели данных, в которой они представлены (например, группу образуют коллекции, развернутые на СУБД DB2). Нередко группу образует одна коллекция (например, социальная сеть или сервис микроблогов). Для каждой группы должен быть реализован компонент *Загрузчик коллекции,* осуществляющий извлечение данных из коллекции при помощи прикладного программного интерфейса и их загрузку в HDFS.

Целевая схема и схемы исходных коллекций подаются на вход *Компоненту сопоставления схем* (в качестве такого компонента в данной работе использован Harmony Schema Matcher [22] – см. раздел 4). Компонент осуществляет полуавтоматическое сопоставление элементов исходных и целевой схем. На основании соответствий элементов создаются правила трансформации данных на языке HIL. На данный момент создание правил осуществляется экспертом вручную, автоматизация генерации правил является предметом дальнейших исследований.

Компонент *Супервизор,* отвечающий за взаимодействие всех компонентов архитектуры, реализуется на языке Jaql. На вход компонент получает исходные коллекции данных. Средствами Hadoop обработка данных распараллеливается по блокам, в которых хранятся файлы с данными (или части файлов). *Супервизор* осуществляет передачу неструктурированных данных (текстов) *Средствам анализа тестов*. Каждый фрагмент текста передается соответствующему, находящемуся на том же компьютере кластера, что и этот фрагмент, компоненту анализа текстов для извлечения сущностей. Взаимодействие между компонентом *Супервизор* и средствами анализа текстов осуществляется по протоколам, поддерживаемым этими средством, и реализуется на языке Jaql с помощью пользовательских функций на языке Java. Также *Супервизор* вызывает компонент *Интегратор* для интеграции структурированных данных и сущностей, извлеченных из неструктурированных данных.

*Средства анализа текстов* производят извлечение из входных фрагментов текстов именованных сущностей различных видов на основе графематического, морфологического, семантико-синтаксического, концептуального анализа [1, 5, 26, 27]. Результатом работы средств анализа текстов является формализованное представление извлеченных сущностей. Извлеченные сущности сохраняются в файловой системе HDFS.

Компонент *Интегратор* исполняет правила трансформации данных, разрешения сущностей и слияния сущностей на языке HIL.

Компонент *Загрузчик в РСУБД* осуществляет сохранение интегрированных сущностей, полученных в результате работы компонента *Интегратор* в реляционной базе данных для дальнейшего анализа.

Анализ и визуализация интегрированных данных осуществляется с помощью существующих средств анализа данных, например, средств анализа многомерных кубов или средств интеллектуального анализа данных.

## 4 Пример извлечения информации из разноструктурированных данных и ее приведения к целевой схеме для задачи социально-политического мониторинга

Предложенный в разделе 2 подход был опробован на задаче мониторинга тональности отношения населения к экономическим и политическим вопросам в конкретном регионе на основе извлечения информации из сообщений в социальных сетях и электронных СМИ.

Архитектура, необходимая для решения задачи, была развернута на небольшом Hadoop-кластере из 5 узлов, установленном в ИПИ РАН.

### 4.1 Сбор исходных данных

В качестве релевантных задаче были выбраны следующие коллекции:

- коллекция публикаций региональных электронных СМИ;
- коллекция сообщений из социальной сети ВКонтакте, авторы которых проживают в регионе;
- коллекция сообщений, загруженных из сервиса микроблогов Twitter, авторы которых проживают в регионе;

- коллекции профилей пользователей ВКонтакте и Twitter.

Сбор сообщений региональных СМИ осуществлялся при помощи информационно-аналитической системы «Астарта».

Публикации региональных СМИ, сообщения ВКонтакте и Twitter являются слабоструктурированными данными, содержащими неструктурированную текстовую часть. Коллекции профилей пользователей содержат структурированные данные.

Для загрузки каждой из исходных коллекций в HDFS был реализован на языке Java отдельный загрузчик.

Коллекции текстовых сообщений были представлены в виде набора файлов в формате CSV. Каждая строка файла представляет некоторое сообщение (публикацию), извлеченное из коллекции, в виде пары *<идентификатор сообщения, текст сообщения>*. Пример фрагмента файла, содержащего текстовые сообщения, выглядит следующим образом:

```
"33650_515","Политолог Александр Лобов утверждает,
что ситуация в регионе критическая."
"36147_2936","МТС подкинул проблем ) ни один номер
не работает %) сеть легла похоже у них"
```

### 4.2 Спецификация целевой схемы

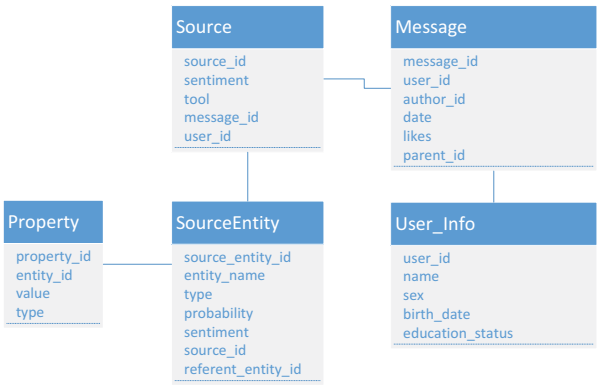



Рис. 3. Фрагмент спецификации целевой схемы

Для задачи была разработана целевая схема, фрагмент спецификации которой изображен на рис. 3. Схема включает отношения для представления извлекаемых из текстов сущностей и отношения для представления информации о сообщениях и их авторах:

- *SourceEntity* содержит информацию о сущностях, извлеченных из заданного набора неструктурированных данных;
- *Property* содержит информацию о свойствах извлеченных сущностей;
- *Source* содержит информацию об информационных ресурсах, из которых извлекались сущности;
- *Message* содержит информацию о сообщениях;
- *User_Info* содержит информацию об авторах сообщений.

### 4.3 Извлечение сущностей из текстовых сообщений

В качестве средств анализа русскоязычных текстов для решения задачи мониторинга использовался Семантико-ориентированный процессор Pullenti (Puller of Entities) [26], разработанный в ИПИ РАН. Pullenti служит для извлечения сущностей из текстовых документов и анализа тональности этих сущностей. Pullenti реализован на языке .NET. В ОС Linux в среде Hadoop Pullenti запускается с помощью кроссплатформенной среды Mono. Он может работать как в пакетном режиме, так и в режиме сервера, обрабатывающего SOAP-запросы. Pullenti производит обработку входных текстов для выделения именованных сущностей некоторых видов и тонального анализа.

Фрагмент спецификации исходной схемы извлекаемых сущностей (определенной выходным форматом Pullenti), изображен на рисунке 4.

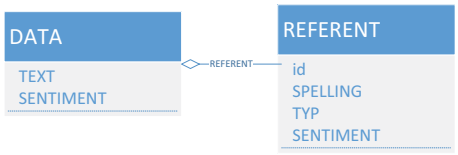



Рис. 4. Фрагмент спецификации исходной схемы извлеченных сущностей, определенной выходным форматом Pullenti

Здесь *DATA* отвечает обрабатываемому сообщению; атрибут *TEXT* – идентификатору сообщения; атрибут *SENTIMENT* – тональности текста. *REFERENT* отвечает сущности, извлекаемой из текста; атрибут *SPELLING* – фрагменту текста, соответствующему извлекаемой сущности; атрибут *id* – уникальному идентификатору сущности; атрибут *TYP* – виду извлекаемой сущности (например, *PERSON*).

Результатом работы Pullenti являются извлеченные сущности, представленные в формате XML и удовлетворяющие исходной схеме.

Например, если на вход системе подается текст

```
"33650_515","Политолог Александр Лобов утверждает,
что ситуация в регионе критическая."
```

то спецификация извлеченных сущностей для него выглядит следующим образом:

```
<DATA TEXT="33650_515" SENTIMENT="-1">
  <SENT SPELLING="КРИТИЧЕСКАЯ" COEF="-1"/>
  <REFERENT id="7" SPELLING="политолог"
     TYP="PERSONPROPERTY">
     <NAME>политолог</NAME>
  </REFERENT>
  <REFERENT id="8" SPELLING="Александр Лобов"
     TYP="PERSON" SENTIMENT="-1">
     <SEX>MALE</SEX>
     <LASTNAME>ЛОБОВ</LASTNAME>
```

```
    <FIRSTNAME>АЛЕКСАНДР</FIRSTNAME>
    <ATTRIBUTE href="#7">политолог</ATTRIBUTE>
  </REFERENT>
</DATA>
```

### 4.4 Сопоставление элементов исходной и целевой схем

Для установления соответствий между элементами схем в данной работе использовался инструментарий Harmony Schema Matcher [22]. Сопоставление элементов схем осуществляется на основе лингвистического анализа имен элементов и другой ассоциированной с ними информации. Используются несколько стратегий установления соответствий между элементами. Например, одна из стратегий сравнивает использование различных слов в описаниях элементов. Другая стратегия включает расширение имен элементов с использованием тезауруса. Harmony поддерживает различные модели данных, включая XML Schema, SQL DDL, OWL. Инструментарий предоставляет графический интерфейс для уточнения экспертом найденных соответствий.

На рис. 5 представлен графический интерфейс установления соответствий между элементами исходной схемы (подраздел 4.3) и целевой схемы (подразделе 4.2).

С помощью этого интерфейса установлены, в частности, следующие соответствия между элементами целевой схемы и схемы Pullenti:

```
SourceEntity = REFERENT
SourceEntity.name = REFERENT.SPELLING
SourceEntity.type = REFERENT.TYP
SourceEntity.probability = 100
SourceEntity.sentiment = REFERENT.SENTIMENT

Source = DATA
Source.entity_id = DATA.TEXT
Source.sentiment = DATA.SENT
Source.tool = “SOLP”
Source.message_id = DATA.TEXT
Source.user_id = DATA.TEXT
```

Аналогично устанавливаются соответствия между элементами целевой схемы и схемы коллекции об авторах сообщений.

### 4.5 Трансформация данных

На основе установленных соответствий между элементами схем (раздел 4.4) созданы правила трансформации информации, представленной в исходных схемах, в информацию, представленную в целевой схеме.

В качестве примера ниже представлены правила (на языке HIL) трансформации элемента DATA исходной схемы Pullenti в элементы *SourceEntity* и *Source* целевой схемы:

```
// Разрешение конфликтов
@jaql{
 getMessageId = fn(s) substring(0, strPos(s, '_')-1)
 getUserId = fn(s) substring(strPos(s, '_')+1, strLen(s))
}

declare getMessageId: function string to string;
declare getUserId: function string to string;


// Исходная схема
declare DATA: set [ TEXT: string, SENT: string, TYP: string,
REFERENT: set [ id: string, SPELLING: string, TYP: string,
SENTIMENT: int]];

// Целевая схема
declare SourceEntity : set [ name: string, type: string,
probability: int, sentiment: int, source_id: string];
declare Source : set [ source_id: string, sentiment: int, tool:
string, message_id: string, user_id: string];

// Правила трансформации
insert into SourceEntity
select [ name: d.REFERENT.SPELLING
        , type: d.REFERENT.TYP
        , probability: 100
        , sentiment: d.REFERENT.SENTIMENT
        , source_id: d.TEXT
       ]
from DATA d
where ;

insert into Source
select [ source_id: d.TEXT
        , sentiment: d.SENT
        , tool: “SOLP”
        , message_id: getMessageId(d.TEXT)
        , user_id: getUserId(d.TEXT)
       ]
from DATA d;
```

Функции *getMessageId* и *getUserId,* используются для разрешения конфликтов значений между спецификациями схем. Они реализованы с помощью пользовательских функций (user-defined functions) на языке Jaql.

Спецификации исходной и целевой схемы определены при помощи операции *declare*. Правила трансформации определены при помощи операции *insert*.

### 4.6 Анализ и визуализация информации

Визуализация интегрированных данных (построение отчетов и диаграмм) осуществлялась на платформе Cognos BI [15]. Для этого интегрированная информация была загружена в реляционную СУБД DB2 с помощью программы, реализованной на языке Java. Пример отчета изображен на рисунке 6. Детальное обсуждение процесса построения многомерного куба и отчетов над ним выходит за рамки данной статьи.

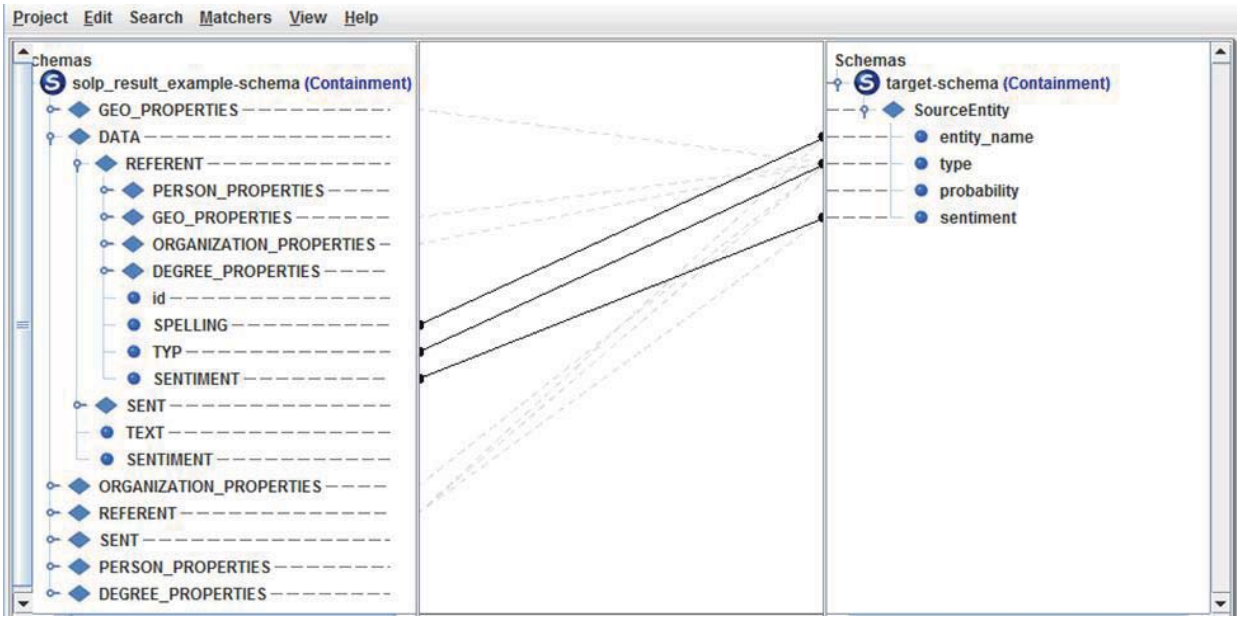
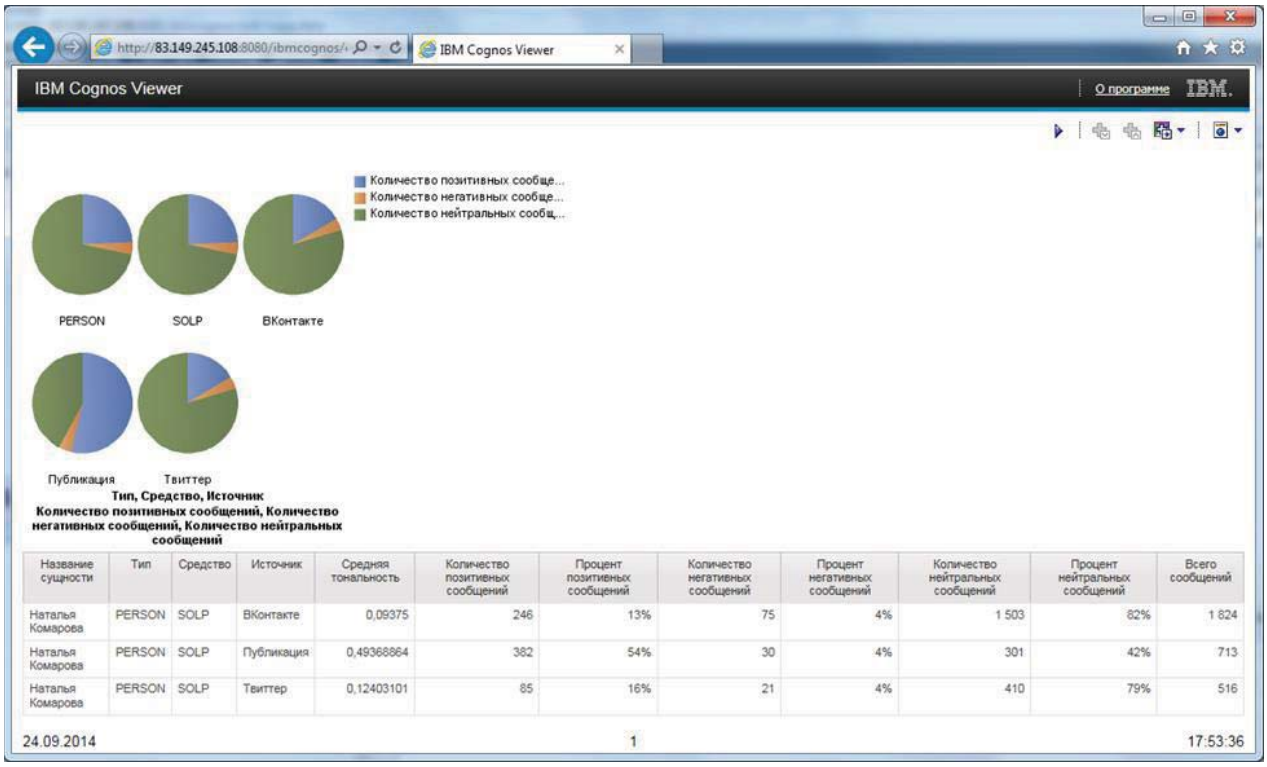



Рис. 5. Пример установления соответствий между элементами схем с использованием графического интерфейса Harmony




| Название сущности | Тип | Средство | Источник | Средняя тональность | Количество позитивных сообщений | Процент позитивных сообщений | Количество негативных сообщений | Процент негативных сообщений | Количество нейтральных сообщений | Процент нейтральных сообщений | Всего сообщений |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Наталья Комарова | PERSON | SOLP | ВКонтакте | 0,09375 | 246 | 13% | 75 | 4% | 1 503 | 82% | 1 824 |
| Наталья Комарова | PERSON | SOLP | Публикация | 0,49368864 | 382 | 54% | 30 | 4% | 301 | 42% | 713 |
| Наталья Комарова | PERSON | SOLP | Твиттер | 0,12403101 | 85 | 16% | 21 | 4% | 410 | 79% | 516 |

Рис. 6. Пример отчета о распределении тональности сообщений для сущности типа *Персона*

## 5 Родственные работы

*Сопоставление схем (schema matching)* обычно является первым этапом процесса интеграции данных. Эта задача достаточно сложна, поэтому многие системы до сих пор требуют ручной спецификации соответствий элементов схем (с помощью графического интерфейса пользователя). Однако, ручная спецификация семантических соответствий между элементами схем слишком трудоемка и подвержена ошибкам в том случае, когда схемы и/или их количество достаточно большие. В связи с этим, развиваются методы автоматического или полуавтоматического поиска соответствий [10, 11, 22].

Среди разнообразных алгоритмов установления соответствий между элементами схем, наиболее часто применяемыми являются алгоритмы, основанные на метаданных, которые используют характеристики элементов схем, такие, как имена,

типы данных, комментарии [10, 22]. Алгоритмы, основанные на экземплярах, устанавливают соответствие между элементами схем на основе близости их экземпляров [17, 25]. Для улучшения работы алгоритмов может использоваться дополнительная информация, например, для установления соответствий по именам элементов, они могут быть расширены с помощью общих тезаурусов (таких как Wordnet), или могут использоваться списки и/или тезаурусы синонимов.

В работе [3] установление соответствий между элементами схем основано на онтологических спецификациях этих элементов. Элемент спецификации одной схемы онтологически релевантен элементу спецификации другой схемы, если между соответствующими им онтологическими понятиями установлена позитивная ассоциация, или ассоциация обобщения/специализации.

Установленные соответствия используются для генерации правил преобразования данных из исходной схемы в целевую [3, 12].

Существует ряд как коммерческих, так и исследовательских программных средств для сопоставления схем. В коммерческих средствах, таких, как IBM Infosphere Data Architect, Microsoft Biztalk server, SAP Netweaver Process Integration, или Altova MapForce, сопоставление схем является первым этапом генерации отображений схем (например, для трансформации данных). Из некоммерческих средств можно выделить, например, прототип Comma++ [10] для сопоставления схем и онтологий; прототип Harmony [22], являющийся одним из компонентов в проекте Open Information Integration [23] по разработке открытой инфраструктуры для интеграции информации.

В данной работе для сопоставления схем был использован инструментарий Harmony, однако, предлагаемая архитектура допускает применение любого другого удобного инструментария.

*Трансформация данных* представляет собой процесс преобразования данных заданных в исходной схеме в данные представляемые в целевой схеме. При этом, целевая схема может быть, как заранее заданной, так и генерируемой на основе интеграции нескольких исходных схем. Заданная целевая схема позволяет фиксировать схему под решение конкретных задач, и не зависеть от дальнейшего подключения новых источников данных. Но при этом она может достаточно сильно отличаться от исходных схем, что затрудняет задачу трансформации данных.

Для генерации правил трансформации данных используются зависимости между элементами схем, найденные на этапе сопоставления схем. Правила трансформации данных могут задаваться на различных языках трансформации. Обычно средства трансформации данных используют свои собственные языки спецификации трансформаций.

Одним из первых средств для генерации кода (запросов) для трансформации данных была система Clio [12], разработанная IBM. Clio предоставляет алгоритмы и средства установления соответствий между данными в разных схемах, и автоматической генерации запросов для трансформации данных.

Проект OpenII (Open Information Integration [23]) предоставляет набор open-source программных средств для интеграции информации, включая трансформацию данных.

В настоящее время средства трансформации данных активно используются при создании хранилищ данных (data warehouse). Существует ряд коммерческих средств, таких, как IBM InfoSphere DataStage, Microsoft Integration Services, Oracle Warehouse Builder, Informatica PowerCenter, Clover ETL. Существует некоммерческий, open-source вариант продукта Clover ETL [9]. Он включает в себя средство CloverETL Designer для создания и выполнения ETL-графов, и приложения CloverETL Server для администрирования, запуска и мониторинга ETL графов. Clover ETL имеет средства для работы с данными, хранимыми в Hadoop.

В данной работе для реализации трансформаций данных используется разработанный IBM декларативный язык HIL [14], ориентированный на разрешение и слияние сущностей в Hadoop-инфраструктуре. Язык HIL позволяет реализовать:

- методы трансформации данных из исходной схемы в целевую. При этом функции разрешения различных конфликтов между этими схемами могут быть описаны также с помощью пользовательских функций (user-defined functions), реализуемых на языках Jaql или Java;
- методы извлечения, сопоставления и группирования, разбора, связывания, устранения дублирования различных разноструктурированных представлений информации об одних и тех же сущностях реального мира;
- методы и операции слияния данных об одних и тех же сущностях реального мира и их связей, представленных в разных коллекциях, образованных в процессе разрешения сущностей.

## 6 Заключение

В ходе исследований получен опыт построения систем для извлечения информации из разноструктурированных данных, проверенный при решении практической задачи в одной из областей с интенсивным использованием данных – области социально-политического мониторинга - с использованием стека технологий обработки и управления большими данными.

Структурирован процесс извлечения информации из разноструктурированных данных, ее интеграции и приведения к целевой схеме. Разработана и реализована архитектура,

поддерживающая данный процесс. Получен опыт встраивания средств текстовой аналитики в Hadoop-инфраструктуру, их применения для извлечения информации из неструктурированных данных. Также получен опыт использования декларативного языка высокого уровня HIL для трансформации исходных коллекций данных в целевую схему хранилища данных.

Всего из сервиса микроблогов Twitter было извлечено более 500 тыс. сообщений от 30 тыс. пользователей, из социальной сети ВКонтакте – боле 4 млн. сообщений от 600 тыс. пользователей, из региональных СМИ – около 7 тыс. публикаций. Сообщения из Twitter и ВКонтакте были извлечены за 2011-2014 гг., публикации СМИ – за 2 месяца 2014 г.

Работа выполнена при поддержке РФФИ (гранты 13-07-00579, 14-07-00548) и Института проблем информатики ФИЦ ИУ РАН (проект 38.25) ,

## Литература


[1] Белоногов Г. Г., Гиляревский Р. С., Хорошилов Ал-др А., Хорошилов Ал-ей А. Развитие систем автоматической обработки текстовой информации // Нейрокомпьютеры: разработка, применение. – 2010, №8. – С. 4–13.

[2] Брюхов Д. О., Скворцов Н. А. Извлечение информации из больших коллекций русскоязычных текстовых документов в среде Hadoop // Электронные библиотеки: перспективные методы и технологии, электроные коллекции (RCDL 2014): Тр. 16-й Всеросс. науч. конф. – Дубна: ОИЯИ, 2014. С. 391-398.

[3] Брюхов Д. О. Конструирование информационных систем на основе интероперабельных сред информационных ресурсов. Дис. канд. техн. наук: 05.13.11. — М.: ИПИ РАН, 2003. — 158 с.

[4] А. Е. Вовченко, Л.А. Калиниченко, Д.Ю. Ковалев. Методы разрешения сущностей и слияния данных в ETL-процессе и их реализация в среде Hadoop. Информатика и ее применения, 2014, т.8, вып.4, с. 94-109

[5] И.П. Кузнецов, А.Г. Мацкевич. Семантико-ориентированные системы на основе баз знаний: монография. – М.: Связьиздат, 2007. – 173 с.

[6] Ступников С. А., Вовченко А. Е. Комбинированная виртуально-материализованная среда интеграции больших неоднородных коллекций данных // Электронные библиотеки: перспективные методы и технологии, электронные коллекции (RCDL 2014): Тр. 16-й Всеросс. науч. конф. – CEUR Workshop Proceedings 1297:201-210. http://ceur-ws.org/Vol-1297/

[7] Annotation Query Language. https://www-01.ibm.com/support/knowledgecenter/SSPT3X_3.0.0/com.ibm.swg.im.infosphere.biginsights.aqlref.doc/doc/aql-overview.html

[8] Kevin S. Beyer, Vuk Ercegovac, Rainer Gemulla, Andrey Balmin, Mohamed Eltabakh, Carl-Christian Kanne, Fatma Ozcan, Eugene J. Shekita. Jaql: A Scripting Language for Large Scale Semistructured Data Analysis. VLDB 2011.

[9] Clover ETL. http://www.cloveretl.com/

[10] Do H. H., Rahm E. (2002) COMA – A system for flexible combination of schema matching approaches. In: VLDB. VLDB Endowment, pp 610–621.

[11] Euzenat J., et al. (2004) State of the art on ontology matching. Tech. Rep. KWEB/2004/D2.2.3/v1.2, Knowledge Web

[12] Fagin R., Haas L. M., Hernandez M. A., Miller R. J., Popa L., Velegrakis Y. (2009) Clio: Schema mapping creation and data exchange. In: Conceptual modeling: Foundations and applications. LNCS 5600. Springer, Heidelberg.

[13] He B., Chang K. C. (2006) Automatic complex schema matching across Web query interfaces: A correlation mining approach. ACM Trans. Database Syst 31(1):346–395.

[14] Hernandez. M., G. Koutrika, R. Krishnamurthy, L. Popa, and R. Wisnesky. 2013. HIL: A high-level scripting language for entity integration. 16th Conference (International) on Extending Database Technology (EDBT'13) Proceedings. Genoa. 549–560.

[15] IBM Cognos Business Intelligence 10.2.0 documentation. 2014. - http://goo.gl/XbBT8Z

[16] IBM InfoSphere BigInsights Information Center. 2014. - http://pic.dhe.ibm.com/infocenter/bigins/v2r1/index.jsp

[17] Kirsten T., Thor A., Rahm E. (2007) Instance-based matching of large life science ontologies. In: Proceedings of data integration in the life sciences (DILS). LNCS, vol 4544. Springer, Heidelberg, pp 172–187.

[18] Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, Shivakumar Vaithyanathan, Huaiyu Zhu. SystemT: a system for declarative information extraction, ACM SIGMOD Record 37(4), 7–13, ACM, 2009.

[19] Donald Miner. MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems. O'Reilly Media, 2012.

[20] Mono - cross platform, open source .NET framework. 2014. - http://www.mono-project.com

[21] White T. 2012. Hadoop: The definitive guide. 3rd ed. O'Reilly Media. 688 p.

[22] P. Mork, L. Seligman, A. Rosenthal, J. Korb, C. Wolf. The Harmony Integration Workbench. International Journal of Data Semantics. December 2008.

[23] Seligman L., Mork P., Halevy A. Y. et al. (2010). OpenII: An open source information integration

toolkit. In: Proceedings of ACM SIGMOD conference. ACM, NY, pp. 1057–1060.

[24] The Forth Paradigm: Data-Intensive Scientific Discovery. Eds. Tony Hey, Stewart Tansley, and Kristin Tolle. Redmond: Microsoft Research, 2009. - http://goo.gl/GqkDX1

[25] Thor A., Kirsten T., Rahm E. (2007) Instance-based matching of hierarchical ontologies. In: Proceedings of 12th BTW conference (Database systems for business, technology and web). Lecture Notes in Informatics 103, pp 436–448.

[26] Семантико-ориентированный процессор Pullenti. 2015. - http://pullenti.ru/

[27] Лингвистическое ПО МетаФраз R10. 2015. - http://www.metafraz.ru/index/0-4

## Information Extraction from Multistructured Data and its Transformation into a Target Schema

Dmitry Briukhov, Sergey Stupnikov,
Leonid Kalinichenko, Alexey Vovchenko

According to the 4th paradigm formulated by Jim Gray in 2007, data are now one of the main driving force for progressing of science. Such data are obtained in the result of observations carried out by high-tech instruments or accumulated in the process of human activity in economy, industry, social environment, etc. Actually, the scientific knowledge is generated in process of the data intensive analysis resulted in knowledge extraction from these data. Fast growth of the data volume and diversity in various data intensive domains causes development of new methods and facilities for analysis and management of massive multistructured data. In this paper the experience is summed up that has been accumulated in the process of exploring of methods, infrastructures and programming facilities intended for extraction and integration of information out of multistructured data. The extracted information should correspond to the needs of the specific problems. Such needs are defined by the target structured schema.