ИДЕНТИФИКАЦИЯ ОБЪЕКТОВ В ЗАДАЧЕ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ДОКУМЕНТОВ

А. С. Серый (32112.alien@gmail.com)

Е. А. Сидорова (lena@iis.nsk.su)

Институт систем информатики им. А. П. Ершова СО РАН, Новосибирск, Россия

Предлагается подход к автоматизации наполнения информационной системы данными, полученными в результате автоматической обработки текстов. Учитывается устаревание информации, появление неточных и дублирующихся данных, противоречия с уже имеющейся информацией.

Ключевые слова: автоматическая обработка, автоматизация, информационная система, идентификация объектов.

OBJECT IDENTIFICATION IN PROBLEM OF AUTOMATIC DOCUMENT PROCESSING

A. S. Seryi (32112.alien@gmail.com)

E. A. Sidorova (lena@iis.nsk.su)

Institute of Informatics, Systems SB, Russian Academy of Sciences, Novosibirsk, Russian Federation

The paper presents an approach to automation of filling of an information system with the data obtained as a result of automatic document processing. The extracted data must be standardized as a network of information objects of a certain format. The backbone of such technique is to build so called focus set for every information object found in a text. Focus set for a single information object consists of all of the relations between this object and other input entities. There are several separate data processing stages: the search for duplicates, direct search, the search for similars and the search via the focus sets technique. A degree of data reliability is also provided. Thus an obsolescence of information, occurrence of the inexact and duplicated data, and conflict of new data with legacy information is taking into consideration.

Key words: automatic processing, automatization, information system, object identification.

Введение

В связи с быстрым развитием Интернет-технологий стремительно увеличивается количество накапливаемой неструктурированной текстовой информации. Вследствие этого возрос интерес к системам, которые позволяют автоматически извлекать знания из представленных документов и преобразовывать её в такую форму, с которой будет удобнее работать конечному пользователю. Таким образом, одной из основных задач, решаемых разработчиками информационных систем, является сканирование корпуса документов, написанных на естественном языке, и наполнение базы данных выделенной из текста полезной информацией. Для решения этой задачи существуют различные инструменты: достаточно известной является линейка продукции компании RCO (http://www.rco.ru/), также можно упомянуть и систему ИСИДА-Т [1,2] разработки Исследовательского Центра Искусственного Интеллекта.

Современные подходы извлечения информации не предусматривают проверки полученных данных при наполнении БД. Однако это может затруднить поиск конкретной информации в огромных архивах информационных систем. Информация устаревает, появляются копии имеющихся данных, возможно появление противоречий. Чтобы избежать таких ситуаций, снизить загруженность баз данных недостоверной и избыточной информацией, необходима предварительная обработка извлекаемых из текста фактов.

В данной статье предлагается подход к автоматизации наполнения информационной системы данными, полученными в результате автоматической обработки естественно-языковых ресурсов. Извлекаемые из текста данные должны быть унифицированы в виде сети информационных объектов[1] определенного формата. В частности, нами для извлечения данных используется текстовый анализатор, разрабатываемый в ИСИ СО РАН [3]. Формат ИО, в котором он инкапсулирует извлеченные данные, и был взят нами за основу. В процессе добавления в базу данных полученные ИО идентифицируются. Под идентификацией понимается однозначное разрешение *контекстной омонимии*, возникающей в том случае, когда одному входному объекту по его набору атрибутов можно сопоставить несколько объектов из базы данных.

1. Знания и данные в информационной системе

В дальнейшем под информационными системами будем понимать так называемые *информационные системы под управлением онтологии* [3], т. е. системы, предметная область которых ограничена и явно описана на определенном языке (считается, что описание доступно как конечным пользователям, так и внешним программным сервисам системы). Каждый ИО соответствует некоторому понятию онтологии и имеет заданную им структуру. Между

[1] *Информационный объект (ИО)* — описание некоторого объекта предметной области. Наборы разнотипных ИО составляют информационное наполнение системы.

ИО могут существовать связи, семантика которых определяется отношениями, заданными между соответствующими понятиями онтологии. Здесь, онтология — шестерка вида <C, A, T, D, R, F>, где

C — множество классов, описывающих понятия предметной области;
A — множество атрибутов понятий;
T — множество типов данных;
D — множество доменов (домен атрибута определяет множество его допустимых значений);
R — множество отношений, заданных на классах;
F — множество ограничений на значения атрибутов.

Каждый атрибут имеет, по крайней мере, имя и значение, и используется для хранения информации, специфичной для объекта и привязанной к нему. Значение атрибута может быть сложным типом данных.

Мы рассматриваем текстовый анализатор как внешний сервис информационной системы, необходимый для автоматической обработки текстовых документов и наполнение БД системы. В качестве анализатора может выступать любая программная система обработки текста, результат работы которой приводится к «понятному» для информационной системы формату (т. е. сети ИО). Рассматриваемый в статье модуль является универсальным (с указанными ограничениями) передатчиком, решающим задачу контроля данных. Однако предлагаемый подход позволяет решать не только эту задачу, но и улучшать результаты анализа текста, предоставляя доступ к глобальному контексту, т. е. знаниям не представленным непосредственно в тексте, поскольку для идентификации объектов необходимо обращение к онтологии системы и ее информационному наполнению. Информационной системой, на которой будет демонстрироваться данный подход, является портал знаний по компьютерной лингвистике (КЛ) [4]. В контексте данного портала имеются следующие основные классы извлекаемых объектов: *раздел науки, персоны, организации, географическое место, событие*, *деятельность, результат (продукт) деятельности,* а также связей между ними: *Работает-в, Направление-Исследований, Персона-Участник-События* и др.

2. Методика идентификации данных

В качестве входных данных выступают список извлеченных из документа ИО, упорядоченный по встречаемости в тексте, и список связей между этими ИО. Ключевым для предлагаемого метода идентификации данных является понятие *фокусного множества.* Фокусное множество включает все экземпляры отношений, с помощью которых текущий объект непосредственно связан с другими входными объектами. При этом множество отношений разбивается

на подмножества связей с идентифицированными и требующими идентификации объектами.

Основой метода является построение фокусных множеств для найденных в тексте объектов и сопоставление с фокусными множествами объектов, уже содержащихся в базе данных информационной системы.

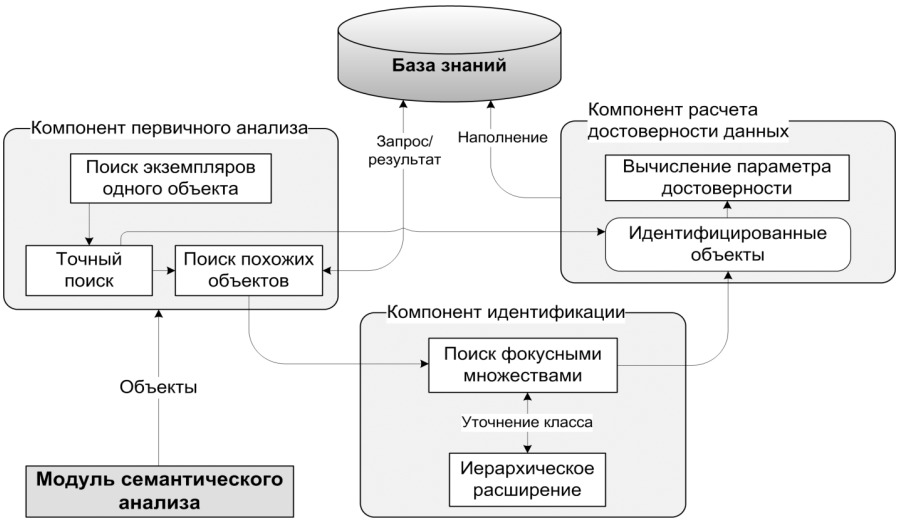

Рис. 1. Схема процедуры идентификации

На Рис.1. представлена общая схема процесса идентификации, который включает:

- Поиск дубликатов объектов[2];
- Точный поиск;
- Поиск похожих объектов;
- Поиск фокусными множествами.

Теперь подробнее о каждом этапе.

2.1. Поиск различных экземпляров одного объекта

В процессе обработки текста в случае референции [5] к упомянутому ранее ИО могут порождаться дубликаты объекта. Чтобы объединить всю

[2] Под *дубликатами объекта* подразумеваются объекты, возникшие вследствие многократного упоминания одного и того же объекта в тексте документа. Они могут содержать в себе различные непересекающиеся части сообщаемой в тексте информации о различных свойствах объекта.

информацию об ИО в одном месте, следует установить *кореферентность* дубликатов.

Дубликаты могут быть обнаружены по нескольким внешним признакам. Во-первых — это наличие связей и неопределенность ключевых атрибутов, кроме некоторого необходимого для отсылки набора, какого именно — зависит от класса. Например, у человека это чаще всего будут имя и отчество или фамилия, у организации — название, тип или аббревиатура. Во-вторых, потенциальными дубликатом также может считаться ИО, не имеющий связей с другими объектами. Например:

(1) *ABBYY — компания, производящая электронные словари и программное обеспечение для распознавания документов. Наиболее известные продукты* ***компании*** *— система распознавания документов FineReader и электронные словари Lingvo.*

(2) *Ю. Д. Апресян — российский лингвист, академик РАН. … Юрий Дереникович Апресян родился в 1930 году в Москве.*

Пример (1) иллюстрирует случай упоминания компании для установления связи с производимыми продуктами, а пример (2) — случай дальнейшего уточнения информации после краткого вступления.

Объект, удовлетворяющий одному из признаков, считается потенциальным дубликатом и сравнивается с объектами классов того же иерархического дерева. Поиск производится и вправо и влево, но, в силу правил построения предложений и текстов в русском языке, приоритет отдается объектам, упомянутым ранее, т. е. объектам «слева».

2.2. Точный поиск

Следует отметить, что для применения основного алгоритма необходим некоторый «стартовый» список идентифицированных объектов. Этот список может быть получен с помощью процедуры точного поиска. По входному объекту в базе данных проводится поиск объектов, имеющих идентичный набор ключевых атрибутов[3]. Этот набор не обязательно должен быть полным. Если был найден лишь один объект, то входной объект идентифицирован, и дальнейший его анализ уже не требуется.

Стоит добавить, что объект может быть признан идентифицированным и без достижения однозначного соответствия с объектом БД, хотя в этом случае он не участвует в формировании фокусных множеств. Это возможно если объект имеет полностью определенный набор ключевых атрибутов

[3] Под *ключевыми атрибутами* понимается набор атрибутов, однозначно идентифицирующий объект в информационном пространстве. Значения всех ключевых атрибутов определены у каждого объекта информационного пространства.

и не достигается однозначное соответствие ни с одним из имеющихся в БД. Такой объект является новым для БД и вносится в информационную систему как есть, а не как уточнение одного из старых объектов.

Для иллюстрации дальнейшего анализа рассмотрим пример из [6], где требовалось выявлять упоминания в сообщениях известных персон и научных организаций, а также извлекать информацию о том где, когда и в какой должности эти персоны работали:

(3) *Александр Михайлович является директором Института русского языка им. В. В. Виноградова РАН с 1997 года.*

В данном примере содержатся объекты классов *Персона* и *Организация*, и имеет место локальная неоднозначность (наименование персоны vs фрагмент наименования организации). В данном случае омонимия снимается на уровне сборки лексических шаблонов объектов: подстрока *В. В. Виноградова* входит в лексическую конструкцию, реализующую шаблон наименования объекта класса *Организация*. С помощью точного поиска можно идентифицировать организацию (*Институт русского языка им. В. В. Виноградова РАН*) в БД.

Персона *Александр Михайлович* задана недостаточно точно (отсутствует ключевой атрибут *Фамилия*), поэтому запускается поиск похожих объектов. Обозначим организацию $\boldsymbol{a}$, а персону — $\boldsymbol{b}$.

2.3. Поиск похожих объектов

Алгоритм предназначен для поиска объектов базы данных, наиболее похожих на объект, найденный в тексте. При построении списка похожих объектов участвуют только атрибуты. Список можно рассматривать как нулевой шаг последовательности фильтраций, осуществляемых алгоритмом поиска фокусными множествами.

Список строится путем сравнения текущего объекта из входного списка с объектами базы данных по различным подмножествам атрибутов. Так i-й шаг представляет собой выбор объектов БД, имеющих совпадения по любому набору из i атрибутов с анализируемым объектом.

Мощность списка найденных объектов БД может становиться только меньше от шага к шагу. За окончательный принимается результат i-го шага при условии $i < n$ и результат шага $(i+1)$ — пустое множество. Если этого не происходит, то после n-го шага выбирается список минимальной положительной мощности. Это и есть наиболее похожие объекты. Если на любом шаге список похожих объектов сузился до одного элемента, то анализируемый объект объявляется идентифицированным и его дальнейшее рассмотрение прекращается. Отметим также, что поиск похожих объектов производится только среди экземпляров одного класса (либо среди экземпляров классов одного иерархического дерева в случае учета иерархии).

Вернемся к примеру (3). Допустим, что в базе данных несколько человек с именем и отчеством *Александр Михайлович*, возможно работающих в той же организации.

Построение списка для объекта ***b:***

Шаг 1: выбираются персоны с именем *Александр* или отчеством *Михайлович***.**

Шаг 2: выбираются персоны с именем *Александр* и отчеством *Михайлович***.**

Согласно предположению, в системе существует больше одной персоны с такими значениями атрибутов. Они и сформируют список наиболее похожих объектов.

2.4. Поиск фокусными множествами

Основной алгоритм процедуры идентификации. Здесь основными фигурантами выступают уже связи объектов. Общий принцип работы кратко описывается следующими шагами:

a. Входной список объектов делится на два подсписка, **A** и **B** — идентифицированных и неидентифицированных объектов соответственно.
b. Для каждого объекта $\mathbf{b}_i \in \mathbf{B}$ строится фокусное множество $\boldsymbol{F_i} = \langle \boldsymbol{b_i^I}, \boldsymbol{b_j^{II}} \rangle$ — пара множеств отношений, связывающих $\mathbf{b}_i$ с объектами подписков **A** и **B** соответственно.
c. Из списка похожих объектов поочередно удаляются объекты, имеющие l связей из множества $\boldsymbol{b_i^I}$, до тех пор, пока мощность его не станет равной 1 ($l = 0, 1, 2, \ldots, |\boldsymbol{b_i^I}|$). В случае если этого не произошло, объект bi не может быть идентифицирован по имеющейся о нем информации.
d. Пусть объект $\mathbf{b}_i$ был идентифицирован на предыдущем шаге. В этом случае он переносится в подсписок **A**, все связи вида $\langle \boldsymbol{b_i}, \boldsymbol{b_j} \rangle \in \boldsymbol{b_j^{II}}$, $\mathbf{b}_j \in \mathbf{B}$, переносятся во множество $\boldsymbol{b_j^I}$, а объекты $\boldsymbol{b_j}$ анализируются даже в том случае, если ранее уже были отброшены за недостатком информации.

Теперь посмотрим, как это будет выглядеть применительно к нашему примеру (3).

$\boldsymbol{A = \{a\}}$, $\boldsymbol{B = \{b\}}$, $\boldsymbol{F_i = \langle b^I, b^{II} \rangle}$ — *фокусное множество объекта* ***b****.*

$\boldsymbol{b^I = \{\langle a,b \rangle\}}$ — *отношение «работает-в» со значением «директор» атрибута «должность».*

Из списка похожих объектов удаляются все, не связанные с ***a*** и имеющие другую должность. Остается либо один, либо ни одного объекта, соответствующего объекту ***b*.** В нашем случае это был *Александр Михайлович Молдован.*

Выполнение алгоритма продолжается до тех пор, пока в подсписке **B** есть хоть один объект, доступный для анализа, после чего база данных редактируется в соответствии с полученными результатами.

$\boldsymbol{b^{II} = \varnothing}$

2.5. Использование иерархических отношений

Понятия онтологии находятся в иерархической связи друг с другом («общее-частное»). Если объект не был идентифицирован, то можно сделать предположение о неточности указания его онтологического класса и расширить ареал поиска на экземпляры всех классов его иерархического дерева, Необходимо определить как классы-наследники, так и классы-родители. Глубина поиска по иерархии понятий может регулироваться в зависимости от количества входных объектов и требований производительности.

Использование иерархии по отношению «часть-целое» возможно в случае, когда объект подчинен другому объекту и имеет сложную структуру, представленную линейными цепочками наименований, совокупность которых образует дерево (множество деревьев) информационных объектов. Для идентификации такого объекта нужно восстановить иерархию вложенности объектов.

3. Наполнение базы данных

При редактировании объекта в системе могут возникать противоречия между старыми и новыми значениями его атрибутов. Это не относится к ключевым атрибутам[4]. Существуют несколько способов разрешения подобных противоречий:

1. Замена старых значений атрибутов на новые. Считается, что новая информация более достоверна.
2. Сохранение старых и новых значений с указанием даты внесения. Потерявшие актуальность данные удаляются экспертом вручную.
3. Введение параметра достоверности значений. По сути, это автоматизация предыдущего способа. Для этого требуется сохранение старых и новых данных, но по мере изменения параметра достоверности, система автоматически избавляется от недостоверных данных.

Как было сказано, третий способ предполагает введение специального параметра, количественно выражающего достоверность того или иного значения или связи. В данной работе предлагается следующая формула расчета такого параметра, выражающая зависимость от трех основных факторов, могущих послужить причинами противоречий с информацией, хранимой в базе данных: времени, рейтинга (авторитета) документа, из которого получены данные, и вероятности ошибки семантического анализатора, обрабатывавшего документ:

$$a\,(\boldsymbol{a}\colon v) = s(\boldsymbol{D}) \cdot h(T) \cdot G$$

где $\boldsymbol{a}\colon v$ — атрибут $\boldsymbol{a}$ в значении v, $\boldsymbol{D}(\boldsymbol{a}\colon v)$ — документ, в котором встретилось $\boldsymbol{a}\colon v$, T –текущая дата, как дата встречи $\boldsymbol{a}\colon v$.

[4] В противном случае, объект не мог бы быть идентифицирован.

Коэффициент $s(\boldsymbol{D})$ отражает зависимость $\boldsymbol{a}$ от степени доверия к документу. Документ — первый из источников противоречий с БД. Чем более авторитетный документ — тем больше доверия к данным, добытым из него.

$$s(\boldsymbol{D}) = \frac{R(\boldsymbol{D}(\boldsymbol{a}{:}\,v))}{\sum_j R(\boldsymbol{D}(\boldsymbol{a}{:}\,v_j))}$$

В числителе стоит суммарный рейтинг документов, в которых встретилось значение v, а в знаменателе — сумма рейтингов документов, в которых встретилось любое из имеющихся значений данного атрибута. Таким образом, $s(\boldsymbol{D})$ представляет собой «удельный вес» документов, упоминающих значение v, в массе всех документов, упоминающих данный атрибут (или связь). $s(\boldsymbol{D}) \leq 1$.

$$R(\boldsymbol{D}(\boldsymbol{a}{:}\,v)) = \sum_{i=1}^{n} r(\boldsymbol{D}_i(\boldsymbol{a}{:}\,v))$$

где $r(\boldsymbol{D}_i(\boldsymbol{a}{:}\ v))$ — рейтинг i-го документа, в котором встретился атрибут $\boldsymbol{a}$ в значении v. Общая формула рейтинга отдельного документа выглядит следующим образом:

$$r(\boldsymbol{D}) = R_{res} \cdot \left(1 - \frac{\sum_{j=1}^{N} \sum_{v_j^i \in D} k_j^i}{\sum_{j=1}^{N} \sum_{i=1}^{n_j} k_j^i}\right)$$

где R_{res} — сумма рейтингов ресурсов документа $\boldsymbol{D}$, N — количество атрибутов, встретившихся в документе $\boldsymbol{D}$ и имеющих больше одного альтернативного значения[5], $\boldsymbol{v}_j^i$ — i-е значение j-го атрибута, $\boldsymbol{k}_j^i$ — количество встречаемостей i-го значения j-го атрибута. Задача классификации ресурсов нами не рассматривается. Предполагается, что либо рейтинги ресурсов, из которых получены документы, вычислены и представлены в информационной системе, либо рейтинг каждого ресурса полагается равным 1.

Следующий коэффициент $h(T)$ отвечает за зависимость достоверности от времени, в том числе и за «старение» информации со временем. Изменения в реальном мире — второй источник противоречий с БД.

$$h(T) = h(T(\boldsymbol{a}{:}\,v)) = 1 + \ln\left((T - t_{last}) + 1\right)$$

Здесь t_{last} — ближайшая к T дата встречи отличного от v значения атрибута $\boldsymbol{a}$.

Третий коэффициент введен для учета ошибки анализатора при извлечении фактов. Ошибка при извлечении фактов — третий источник противоречий с БД.

[5] Учитываются только атрибуты, значения которых не могут быть множественными.

Коэффициент G применяется в случае наличия информации о принципе работы текстового анализатора, в частности, веса, выставленные экспертом схемам сборки фактов [3]. Обозначим эти веса w_l, $l = 1, 2, ..., F$. Также, ошибка при извлечении фактов может возникнуть вследствие неверного толкования понятий, т.е. омонимии. Поэтому G зависит еще и от количества альтернативных в данной позиции значений.

$$G = \frac{c \sum w_l}{L}$$

Здесь c — константа, L — количество альтернативных (омонимичных) значений, $\sum w_l$ — сумма весов схем фактов, участвовавших в формировании $\boldsymbol{a}: v$.

4. Данные эксперимента

В таблице 1 представлены результаты анализа трех документов, предназначенных для портала КЛ.

Таблица 1. Время работы алгоритмов на разных типах документов

К-во слов в документе	К-во ИО	К-во связей	Макс. к-во связей одного ИО	Макс. к-во атрибутов в одном ИО
413	17	14	12	4
65	6	4	1	2
532	11	5	5	5

К-во дубликатов	Поиск дубликатов (мс)	Поиск наиболее похожих объектов (мс)	Общее время работы (мс)	Идентифицировано объектов
3	312	2140	6125	4
0	63	105	394	4
8	453	179	850	4

Как можно видеть, документы имеют довольно небольшой размер. Как правило, это заметки, новостные сообщения или короткие статьи.

В первом случае длительное время обработки обусловлено в основном не количеством извлеченных ИО, а числом связей между ними, что увеличивает время построения и прогонки фокусных множеств, а также количеством ИО с наибольшим числом атрибутов, для которых требуется построить список наиболее похожих объектов. Как видно из таблицы 1, поиск похожих объектов занял более трети общего времени.

Второй документ является обыкновенным новостным сообщением с сайта компании ABBYY и содержит небольшое количество ИО.

Третий документ является отрывком из биографии, он был взят для иллюстрации работы на наборе ИО с высоким содержанием дубликатов, поскольку герой биографии упоминается в ней постоянно. Тем не менее, время анализа

невелико. Это обусловлено тем, что основной ИО — персона, являющаяся предметом статьи, был идентифицирован алгоритмом прямого поиска. В итоге большую часть времени занял поиск дубликатов.

Заключение

Описанный метод применяется при разработке сервисов анализа документов для информационного ресурса «Хроники СО АН» [6] и портала знаний по компьютерной лингвистике [4]. Проведенные эксперименты показали, что алгоритм поиска наиболее похожих объектов дает ощутимые расходы по времени, в случае объектов с десятью и более атрибутами. Это происходит вследствие значительного усложнения генерируемых запросов к базе данных, что приводит к увеличению временных затрат на их оптимизацию и выполнение. Для решения этой проблемы планируется ввести фильтрацию рассматриваемых атрибутов (например, не учитывать полнотекстовые атрибуты типа *Комментарии*, *Толкование, Описание* и др.), и осуществлять их упорядочивание по значимости. Также, возможно искусственно ограничивать время работы алгоритма, а за результат принимать список, построенный к моменту остановки (например, в случае большого количества входных объектов). Будут продолжаться работы по оптимизации механизма расчета коэффициента достоверности и его параметров, вполне вероятно увеличение доли участия эксперта в этой процедуре.

References

1. *Aleksandrovskii D. A., Kormalev D. A., Kormaleva M. S., Kurshev E. P., Suleimanova E. A., Trofimov I. V.* 2006. The Development of Means of Text Analytic Processing in the System ISIDA-T [Razvitie Sredstv Analiticheskoi Obrabotki Teska v Sisteme ISIDA-T]. Trudy 10 Natsional'noi Konferentsii po Iskusstvennomu Intellektu (Proc. of the X National Conference on Artificial Intellect), 2 : 555–563.
2. *Borovikova O. I., Zagorul'ko Iu. A., Zagorul'ko G. B., Kononenko I. S., Sokolova E. G.* 2008. Designing of a Portal of Knowledge on Computational Linguistics [Razrabotka Portala Znanii po Komp'iuternoi Lingvistike]. Trudy 11 Natsional'noi Konferentsii po Iskusstvennomu Intellektu (Proc. of the XI National Conference on Artificial Intellect), 3: 380–388.
3. *Kononenko I. S., Sidorova E. A.* 2009. An Ontology-Based Facts Extraction Approach [Podkhod k Izvlecheniiu Faktov iz Teksta na osnove Ontologii]. Komp'iuternaia Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoi Konferentsii "Dialog 2009" (Computational Linguistics and Intelligent Technologies: Proceedings of the International Conference "Dialog 2009") : 451–457.
4. *Kormalev D. A., Kurshev E. P.* 2006. The Development of Language for Infromation Extraction Rules in the System ISIDA-T [Razvitie Iazyka Pravil Izvlecheniia

Infromatsii v Sisteme ISIDA-T]. Trudy Mezhdunarodnoi Konferentsii “Programmnye Sistemy: Teoriia I Prilozheniia” (Proc. of National Conference "Program Systems: Theory and Applications"), 1 : 365–377.

5. *Lebedev M. V., Cherniak A. Z.* 2001. Ontological Problems of Reference [Ontologicheskie Problemy Referentsii].
6. *Sidorova E. A., Zagorul’ko Iu. A., Kononenko I. S.* 2006. Semantic Approach to Document Analysis basing on the Ontology of Object Area [Semanticheskii Podkhod k Analizu Dokumentov na osnove Ontologii Predmetnoi Oblasti]. Komp’iuternaia Lingvistika i Intellektual’nye Tekhnologii: Trudy Mezhdunarodnoi Konferentsii “Dialog 2006” (Computational Linguistics and Intelligent Technologies: Proceedings of the International Conference “Dialog 2006”) : 468–473.