

УДК 004.822 + 004.021

## МЕТОДЫ И АЛГОРИТМЫ ПОВЫШЕНИЯ ВЫРАЗИТЕЛЬНОСТИ СВЯЗАННЫХ ДАННЫХ (ОБЗОР)

О. А. Невзорова<sup>[0000-0001-8116-9446]</sup>

Казанский федеральный университет, ул. Кремлевская, 18, Казань, 420008

onevzoro@gmail.com

### **Аннотация**

В обзорной статье рассмотрены методы и алгоритмы повышения выразительности связанных данных, подготовленных для публикации в Вебе. Представлены основные подходы к обогащению онтологий, описаны методы, на которых они базируются, а также приведен инструментарий, реализующий эти подходы и инструменты применения соответствующих методов.

Основным этапом в общей схеме жизненного цикла данных в облаке открытых связанных данных является этап построения набора связанных RDF-триплетов. Для улучшения классификации данных и анализа их качества применяются различные методы повышения выразительности связанных данных. Основные идеи рассматриваемых методов связаны с обогащением существующих онтологий (расширением базовой схемы знаний) путем добавления или совершенствования терминологических аксиом. Методы обогащения опираются на методы, применяемые в различных областях, таких как представление знаний, машинное обучение, статистика, обработка текстов на естественном языке, анализ формальных понятий и теория игр.

**Ключевые слова:** *связанные данные, онтология, обогащение онтологии, семантический веб*

### **ВВЕДЕНИЕ**

Термин Linked Data, или «связанные данные», обозначает широкий набор конкретных методов, подходов и технологий для публикации структурированных данных в Вебе. Процесс публикации технологически обеспечивается: представлением данных в виде триплетов «субъект-предикат-объект» на языке RDF, иденти-

фикацией данных с помощью URI, механизмом доступа по протоколу HTTP и спецификацией контролируемых словарей, или онтологий на языках RDFS и OWL. Как правило, множества триплетов логически объединяются и хранятся в виде наборов данных (data sets) или баз знаний. Обзор содержит описание методов и алгоритмов повышения выразительности связанных данных в парадигме Linked Open Data (LOD).

Общая схема жизненного цикла данных в LOD содержит следующие этапы:

- извлечение RDF-триплетов из неструктурированных источников (Extraction);
- сохранение RDF-триплетов в постоянном хранилище с обеспечением доступа (Storage/Querying);
- интерактивное создание и редактирование данных (Manual Revision/Authoring);
- обнаружение и связывание идентичных или семантически связанных данных (Interlinking/Fusing);
- улучшение классификации данных (Classification/Enrichment);
- анализ качества данных (Quality Analysis);
- поиск и просмотр данных (Search/Browsing/Exploration).

Методы и алгоритмы повышения выразительности связанных данных используются на этапах анализа качества данных и улучшения классификации данных.

## **1. МЕТОДЫ ОБОГАЩЕНИЯ ДАННЫХ**

Онтологии Linked Data в основном наполнены экземплярами классов, в результате чего наблюдается нехватка классификации и структурной информации [1]. Данная проблема решается путем обогащения существующих онтологий высокоуровневыми структурами для облегчения задач агрегирования данных и построения запросов.

Под термином «обогащение» понимается автоматическое или полуавтоматическое расширение базовой схемы знаний [1]. Этот термин описывает процесс увеличения выразительности и семантического богатства базы знаний. Обычно это достигается путем добавления или совершенствования терминологических аксиом.

Как правило, методы обогащения могут быть массово применены для создания баз знаний. При таком подходе онтологическая структура не создается сразу напрямую, а постепенно эволюционирует вместе с данными в базе знаний. В идеале это способствует более быстрому созданию баз знаний. В частности, в контексте Web of Linked Data такой подход кажется интересной альтернативой традиционным методам разработки онтологий.

Обогащение базы знаний может быть рассмотрено как подпроблема задачи машинного обучения онтологий. Обучение онтологий отличается тем, что при этом возможно использование внешних источников информации, например, письменного текста. В то же время обогащение предполагает анализ уже существующей базы знаний для улучшения ее схемы.

Методы обогащения опираются на методы, применяемые в различных областях, таких как представление знаний, машинное обучение, статистика, обработка текстов на естественном языке, анализ формальных понятий и теория игр.

Далее будут рассмотрены основные подходы к обогащению, описаны методы, на которых они базируются, а также приведен инструментарий, реализующий данные подходы.

## **2. ОСНОВНЫЕ ПОДХОДЫ К ОБОГАЩЕНИЮ**

Обогащение онтологии использует методы машинного обучения и эвристические методы для добавления дополнительных аксиом к существующей онтологии, при этом применение методов зависит от целевого типа аксиом.

Можно выделить следующие основные направления применения методов обогащения:

- нахождение определений классов (definitions of classes);
- построение аксиом надклассов (super class axioms);
- гибридный подход с использованием жестких правил;
- изучение несвязностей (disjointness);
- завершение (completion);
- нахождение параметров свойств (properties of properties);
- построение отображений (ontology mapping).

## 2.1. Нахождение определений классов

Нахождение определений классов – одна из самых сложных задач обогащения онтологий. Решение этой задачи сильно связано с методами индуктивного логического программирования (Inductive Logic Programming) [2] и требует применения дескрипционной логики.

Разработка проблем обучения в рамках дескрипционной логики начинается с работ [3], [4], в которых использовались так называемые наименьшие общие вышестоящие (least common subsumers) для решения проблемы обучения (несколько модифицированный вариант проблемы нахождения определений классов). Позже в [5] был описан оператор усовершенствования (refinement operator) для дескрипционной логики *ALER* (модификация DL, *Attributive language*, расширенная некоторыми дополнительными ограничениями), а также предложен к использованию нисходящий (top-down) подход. Оператор усовершенствования – метод индуктивного логического программирования, который используется для поиска в пространстве выражений. Нисходящий подход предполагает применение оператора усовершенствования, начиная с наиболее общего класса, в результате чего строится описание этого класса из отображений его подклассов.

Позже оба вышеописанных подхода были объединены и осуществлены в инструменте *YINYANG* [6]. Однако эти алгоритмы имеют тенденцию производить описания классов, очень длинные и сложные для понимания. Алгоритмы, используемые в *DL-learner* [7], преодолевают эту проблему, а также значительно прорабатывают методы обучения и нисходящего усовершенствования (top-down refinement). *DL-FOIL* [8] – это похожий подход, но он основан на смеси восходящего и нисходящего методов усовершенствования описаний классов. Используются альтернативные меры оценки, которые, в том числе, принимают во внимание предположение об открытости мира, которое не было сделано в *ILP* ранее.

Другим подходом к обучению определениям именованных классов является вычисление так называемого наиболее определенного понятия (most specific concept) для всех экземпляров класса. Наиболее определенное понятие – это самое определенное описание класса, такого, что объект есть экземпляр описания класса. В таком случае можно вычислить наименьшее общее вышестоящее (least common subsumer) [9] этих выражений, чтобы получить описание класса. Однако

в выразительной дескрипционной логике наиболее определенное понятие может не существовать, и тогда наименьшее общее вышестоящее – просто дизъюнкция всех выражений. Для простых логик, таких как, например, *EL*, подход кажется многообещающим.

### Инструментарий для обучения определениям классов

**DL-Learner** [8]. DL-Learner – приложение обучения концептам дескрипционной логики на основе примеров, что также позволяет обучаться классам в OWL-онтологиях. Приложение предоставляет инструментарий с основами машинного обучения для DL/OWL для решения задач обучения с помощью экспертов и изучения уже имеющихся данных. Возможности:

- Обучение определениям классов: на основе существующих экземпляров OWL-классов DL-Learner делает предположения для определений классов о включении в них `owl:equivalentClass` или `rdfs:subClassOf` Axiom. Так как алгоритм смещен в сторону коротких и легко понимаемых человеком предложений, то эксперту может быть оказана помощь при редактировании Tbox онтологии;
- Поиск похожих экземпляров: описания классов, предложенные DL-Learner, могут быть использованы для нахождения похожих экземпляров с помощью поиска. Масштабируемость методов позволяет проводить генерацию рекомендаций на лету, например, в веб-сценарии;
- Классификация экземпляров: полученные во время обучения описания классов могут быть использованы для определения, к какому конкретному классу принадлежат новые добавляемые экземпляры.

**DL-FOIL**. Это адаптация алгоритма FOIL (First Order Inductive Learner) [10] для обучения DL-представлениям, с поддержкой языка OWL-DL. В основе системы лежит множество операторов улучшения, позаимствованных из аналогичных систем [11], [12] и с различными целевыми функциями, принимающими во внимание предположение об открытости мира.

**YINYANG (Yet another INduction Yields to ANother Generalization)** [6]. Эта система содержит реализацию методологии управляемого обучения концептам и дескрипционной логики. Система позволяет перепроверить результаты, рассматривая часть из них как новое обучающее множество.

**ORE (Ontology Repair and Enrichment)** [13]. ORE – инструментарий, позволяющий инженерам по знаниям улучшать OWL-онтологии путем исправления противоречий и добавления новых аксиом. ORE использует фреймворк DL-Learner для добавления новых определений классов и надклассов для классов, уже существующих в базе знаний.

## **2.2. Построение аксиом надклассов**

В [14] применены приспособляющиеся эвристики и осуществлена адаптация методов обучения к определениям в онтологиях. Работа направлена на достижение максимальной эффективности и применение методов обучения. Эта статья описывает плагины для двух редакторов онтологий (Protege и OntoWiki), а также стохастические методы, улучшающие результаты работы методов машинного обучения на порядок. Кроме того, алгоритмы, представленные в статье, могут быть применены для построения аксиом надклассов.

## **2.3. Гибридный подход с использованием жестких правил**

Ряд подходов, например, [15], разработан для обучения на гибридных базах знаний, объединяющих онтологии и правила. Обычно гибридные подходы – это обобщение стандартных методов обучения, которые допускают использование жестких правил за счет потери эффективности (из-за большей области поиска). Подобно ситуации с представлением знания требуется очень тщательный выбор баланса между выразительностью языка и эффективностью алгоритмов обучения.

## **Инструментарий для полуавтоматического обогащения онтологий**

**Blomqvist** [16]. Blomqvist предоставляет фреймворк для полуавтоматического построения и обогащения онтологий. Применяется подход, основанный на шаблонах. Система предназначена для обогащения «легких» онтологий путем работы с логической сложностью и выразительностью, а не для получения богатой аксиоматизации. По этой причине шаблоны используются не для нахождения новых аксиом, а для автоматического размещения новых концептов в контексте уже имеющихся более общих сущностей.

#### 2.4. Изучение несвязностей

Исследование, проведенное в работе [17], направлено на изучение несвязностей между классами в онтологии, чтобы более тщательно проверить рассуждения и выводы. Для достижения этой цели могут использоваться как непосредственно сама онтология, так и внешние тексты, например, статьи Wikipedia, соответствующие рассматриваемому понятию. Статья включает обширное исследование, которое показывает, что задача обнаружения несвязностей является трудной, но может быть упрощена путем обогащения онтологии.

#### 2.5. Завершение

Другая задача обогащения – завершение базы знаний. Целью решения такой задачи является приведение базы знаний к законченной форме (в некотором заданном смысле). Например, целью может быть возможность гарантировать, что все отношения включения как подкласса между именованными классами могут быть выведены. В работах [18], [9] исследованы возможности применения формального анализа понятий (FCA) для завершения баз знаний. Этот подход выглядит многообещающим, хотя, возможно, не в состоянии обрабатывать шумы так же хорошо, как методы машинного обучения. Доступно соответствующее дополнение к Protege [19]. Подход, описанный в [20], предлагает обогащать базы знаний путем исследования взаимосвязей в них, что было реализовано в RELExO framework [21]. Подход работает с простыми отношениями и задает эксперту серии вопросов. Эксперт или должен положительно ответить на вопрос, или обеспечить контрпример.

#### Инструментарий для задачи завершения

**RELExO (Relational Exploration for Learning Expressive Ontologies)** [21]. RELExO – инструментарий для расширения связей в OWL/DL-онтологиях, который был разработан для поддержки дополнения и совершенствования сложных описаний классов. RELExO комбинирует обучение сложным описаниям классов из текстовых источников с подходом к обработке связей, основанным на FCA, с целью нахождения связей нижестоящих классов для концептов онтологии.

Генерируются гипотезы расширения связей классов, которые не могут быть

выведены или опровергнуты, исходя из аксиом, уже имеющихся в онтологии. После этого осуществляется поиск контрпримеров среди экземпляров онтологии. В случае, если это не удастся, запрашивается помощь эксперта, который должен предоставить контрпример или подтвердить предложенную гипотезу.

**RoLExO** [20]. RoLExO опирается на тот же тип верификации гипотез и взаимодействия с пользователем, что и RELExO, но генерирует гипотезы об области ограничений для данных.

## **2.6. Нахождение параметров свойств**

Существуют более «легковесные» методы обогащения онтологий, чем описанные ранее. Например, параметры свойств могут быть получены путем простого статистического анализа. Под этим подразумевается возможность того или иного свойства быть симметричным, функциональным, рефлексивным, обратно функциональным и т. д. Точно так же области и диапазоны значений свойств могут быть определены по существующим данным.

## **2.7. Построение отображений онтологий**

Весьма популярным подходом к обогащению онтологий является построение онтологических отображений [1]. Часто задачи, связанные с онтологиями, требуют доступа сразу к нескольким из них. Распределенная природа разработки онтологий привела к тому, что существуют различные онтологии одной или нескольких пересекающихся предметных областей. В таком случае значительно затрудняется понимание одной онтологии с точки зрения другой. Именно для решения данной проблемы строятся отображения между онтологиями.

Выделяются три основных вида онтологических отображений [22]. Одним из ключевых различий между ними является то, как данное отображение строится и поддерживается. Каждое отображение имеет свои преимущества и недостатки.

**Построение отображения между интегрированной и частной онтологиями.** В этом случае производится отображение некоторого концепта из одной онтологии в представление или запрос над другими онтологиями. Это отображение используется для поддержки интеграции онтологий в Semantic Web. Из локальных онтологий извлекается информация в интегрированную онтологию Semantic



Web, обеспечивая единое представление, которое позволяет пользователям получать результаты на запросы к локальным онтологиям. Также возможно создание специализированных интегрированных онтологий из нескольких локальных онтологий, например, для нужд управления знаниями крупного предприятия.

### **Инструментарий:**

**LSD (Learning Source Description)** [23]. Полуавтоматически создаются семантические отображения с применением подхода множественных стратегий для машинного обучения. Этот подход предполагает использование нескольких модулей обучения, при этом каждый модуль использует различную информацию в исходных схемах данных.

LSD использует следующие базовые модули:

- Обучение по именам: сопоставляет XML-элементы, имеющие соответствующий тэг;
- Обучение по содержимому: сопоставляет XML-элементы по данным в схеме (хорошо работает для текстовых данных);
- Наивная байесовская модель: использует значения данных, но не работает для коротких и числовых полей;
- XML-обучение: работает с иерархической структурой входных сущностей.

Множественная стратегия обучения имеет два этапа: подготовка и согласование. На фазе подготовки небольшое количество исходных данных вручную отображаются в промежуточную схему, и эта информация используется для обучения модулей. На фазе согласования система автоматически строит отображения на основе информации, полученной на подготовительной фазе. Также LSD учитывает соответствие ограничений, использует обратную связь с пользователем, вложенные структуры в XML-данных для улучшения точности отображений.

**MOMIS (Mediator Environment for Multiple Information Sources)** [24]. MOMIS создает глобальные виртуальные представления источников информации, независимо от их происхождения или разнородности данных. Онтология строится за 5 фаз:

- Извлечение схемы из локального источника программным модулем

(wrapper);

- Аннотация локального источника с помощью WordNet;
- Генерация общего тезауруса: отношения внутри и между схемами знаний о классах и атрибутах;
  - Генерация глобального виртуального представления: создание глобальной схемы и отображений между атрибутами глобальной и исходных схем с использованием общего тезауруса и схем, построенных на первой фазе;
  - Построение аннотаций глобального виртуального представления с использованием аннотации исходных схем и отображения между глобальной и локальными схемами.

MOMIS строит онтологию, которая более точно описывает домены и предоставляет легко понимаемый смысл содержимого, а также способ расширения ранее созданных концепций путем добавления новых источников.

**A Framework for OIS (Ontology Integration System) [25].** Отображения между интегрированной онтологией и локальными онтологиями представляются как запросы, а онтологии – как дескрипционные логики. Предлагается два подхода к построению отображений:

- Концепты глобальной онтологии отображаются в запросы к локальным онтологиям;
- Концепты локальных онтологий отображаются в запросы к глобальной онтологии.

Отличием этого подхода является возможность построения отображений глобальной онтологии на локальную, в отличие от систем, описанных выше, которые строят отображение локальной онтологии в представление глобальной.

**Построение отображения между двумя частными онтологиями [26].** Сущности одной онтологии преобразуются в сущности другой онтологии, базируясь на семантических отношениях. Исходная и целевая онтологии должны быть связаны на концептуальном уровне.

Данное отображение позволяет онтологиям оставаться привязанными к некоторому контексту, при этом, по-прежнему, оставаясь локальными. Возможно построение связей между онтологиями, когда они не могут быть объединены по причине несовместимости информации, заложенной в них.

Нахождение связей между локальными онтологиями может оказаться не проще, чем между локальной и интегрированной онтологией по причине недостатка общих словарей.

#### **Инструментарий:**

**Context OWL (Contextualizing Ontologies)** [27]. Используются расширенные синтаксис и семантика OWL. Онтологии не могут быть связаны или объединены, если две из них содержат взаимно несовместимые понятия. Однако эти две онтологии могут быть связаны через специальные промежуточные правила, которые содержат базовые понятия о контексте отображения. Такое отображение представляется как множество промежуточных правил, использующих операнды  $\supseteq$ ,  $\subseteq$ ,  $\equiv$ ,  $*$  (связаны) и  $\perp$  (не связаны), например, если  $k$  является более общим понятием, чем  $k_1$  ( $k \supseteq k_1$ ),  $k$  – менее общее понятие  $k_1$  ( $k \subseteq k_1$ ),  $k$  эквивалентно  $k_1$  ( $k \equiv k_1$ ),  $k$  сравнимо с  $k_1$  ( $k * k_1$ ),  $k$  несравнимо с  $k_1$  ( $k \perp k_1$ ).

**СТХМАТЧН** [28]. СТХМАТЧН – это алгоритм обнаружения семантических связей всей иерархической классификации с помощью правил логического вывода. СТХМАТЧН на входе получает классы из иерархической классификации и для каждой пары концептов этих классов (узлы соответствующих знаний, включая смысл в иерархической классификации) возвращает их семантическую связь ( $\supseteq$ ,  $\subseteq$ ,  $\equiv$ ,  $*$ , или  $\perp$ ).

Достоинством СТХМАТЧН является то, что отображениям могут быть приписаны четко определенные модели семантики, и эта структурная, лексическая и контекстная информация принимается в расчет.

**GLUE** [29]. GLUE полуавтоматически создает связи между онтологиями, используя методы машинного обучения. GLUE находит наиболее схожие концепты между онтологиями и просчитывает распределение совместной вероятности этих концептов, используя обучение с множественными стратегиями для оценки степени близости. Первая стратегия основывается на частоте слов в контексте рассматриваемой сущности (например, объединение всех атрибутов сущности) и использует наивный Байесовский подход. Вторая стратегия основана на полном имени входной сущности, а третья комбинирует предсказания первых двух, назначая им веса в соответствии с оценкой доверия.

**MAFRA (Ontology MAapping FRAmework for distributed ontologies in the Semantic Web)** [30]. MAFRA обеспечивает распределенный процесс построения связей, состоящий из пяти вертикальных и четырех горизонтальных модулей.

Горизонтальные модули:

- Нормализация: работает с языковой и лексической разнородностями между исходной и целевой онтологиями;
- Обнаружение общего: находит и устанавливает похожие сущности обеих онтологий;
- Семантическое связывание: строит отображения для изменяемых сущностей исходной онтологии в наиболее близкие сущности целевой онтологии;
- Исполнение: трансформирует сущности исходной онтологии в целевую, основываясь на ранее построенных семантических связях;
- Пост-обработка: перепроверяет результаты трансформации для улучшения качества результатов.

Вертикальные модули:

- Эволюция: обслуживает семантические связи, синхронизируя связи в обеих онтологиях;
- Совместное достижение консенсуса: отвечает за установление консенсуса в семантических связях между обеими онтологиями;
- Ограничения предметной области: улучшает семантические связи, используя тезаурус WordNet соответствующей предметной области;
- Интерфейс: позволяет вмешательство человека для достижения лучших результатов.

**LOM (Lexicon-based Ontology Mapping)** [31]. LOM находит автоморфизмы между словарями онтологий с целью уменьшения вклада человеческого труда в процесс построения отображения. Применяются методы, основанные на использовании: всего термина, частей слова, соответствующего синсета WordNet и типа соответствия. LOM не гарантирует точности и аккуратности построенного отображения и имеет существенные ограничения при работе с абстрактными символами или обозначениями в математике, химии или медицине. Преимуществом данного подхода является почти полная автоматизированность.

**QOM (Quick Ontology Mapping)** [32]. QOM – эффективный метод обнаружения связей между онтологиями, так как имеет очень низкую временную сложность. QOM использует подход, основанный на динамическом программировании. Строятся структуры, рассматривающие отображения-кандидаты, классифицируются по перспективности, в результате чего отсекаются наихудшие. Система применяется в тех случаях, когда критично время работы.

**ONION (ONTology compositiON system)** [33]. ONION разрешает терминологическую гетерогенность и предоставляет соединяющие правила для отображений. Лингвистический модуль определяет все возможные пары терминов в онтологиях и приписывает им соответствующие веса близости. Если вес близости превышает определенный порог, то такая пара принимается, и генерируется соответствующее соединяющее правило. После этого дополнительные соответствия ищутся структурным модулем. Построение связей между онтологиями проводится за несколько циклов. Эксперт может удалять, добавлять или изменять соединяющие правила, предложенные системой через специальный интерфейс.

**OKMS (Ontology-based knowledge management system)** [34]. В OKMS отображения используются для комбинирования либо распределенных, либо гетерогенных онтологий. Процесс осуществляется в 5 шагов:

- Нормализация: если исходная информация не является онтологией, то осуществляется соответствующее преобразование программным модулем;
- Нахождение близких сущностей: генерируется матрица, содержащая меры близостей сущностей разных онтологий;
- Построение семантического отображения: на этом шаге генерируется отображение, описывающее преобразование концептов исходной онтологии в концепты целевой;
- Исполнение: выполнение преобразований семантического отображения;
- Пост-процессинг: дополнительная обработка для улучшения качества результатов.

**OMEN (Ontology Mapping Enhancer)** [35]. OMEN – это вероятностный инструмент для построения отображений онтологий, который улучшает качество уже

построенного отображения, используя Байесовские сети. Байесовские сети используют множество мета-правил, представляющих то, как каждое отображение онтологии влияет на другие отображения, базируясь на структуре онтологии и семантике онтологических связей. Существующие отображения между двумя концептами могут быть использованы для улучшения отображений между соседними концептами.

**P2P ontology mapping (Peer-to-peer ontology mapping)** [36]. Данный фреймворк создает эффективное взаимодействие между онтологиями, участвующими в построении отображения, базируясь на динамических отображениях только части онтологии, релевантной данному взаимодействию. Работа осуществляется в три шага:

- Генерация гипотез;
- Фильтрация гипотез;
- Выбор лучшей гипотезы.

**Отображение онтологий в их объединение** [37]. Устанавливается соответствие между исходными онтологиями для объединения или согласования. При этом определяются пересекающиеся концепты, синонимы, аналоги и уникальные записи в этих двух онтологиях. Выявляются общие структуры и конфликты локальных онтологий перед их сведением или объединением.

#### **Инструментарий:**

**SMART** [38]. SMART – это инструментарий полуавтоматического объединения и сведения онтологий. Он находит лингвистически близкие имена классов и, основываясь на этих эквивалентностях, создает список начальной лингвистической близости (синонимия, единая подстрока, общий суффикс или префикс). Изучается структура отношений между объединенными концептами и строятся эквивалентности между именами свойств и их типами. Инструментарий проверяет результат работы на возможные конфликты, а также предлагает возможные пути их разрешения пользователю.

**OntoMorph** [39]. OntoMorph предоставляет мощный язык правил для определения отображений, способствуя слиянию онтологий и быстрой генерации

базы знаний преобразования. Он комбинирует два мощных механизма для трансформации баз знаний: синтаксическое и семантическое преобразования. Синтаксическое преобразование осуществляется по специальным шаблонам на уровне предложений, семантическое – на основе семантических моделей и правил логического вывода.

**HICAL (Hierarchical Concept Alignment system)** [40]. HICAL предоставляет управление иерархией концептов для объединения/сведения онтологий (одна иерархия концептов сводится к другой), используя методы машинного обучения для сведения множества иерархий концептов, а также использует данные о сущностях в пересечении таксономий для выведения отображений. Иерархия используется для категоризации и получения синтаксической информации, а не вычисления близости между словами, что позволяет категорировать различные слова в рамках одного концепта.

**Anchor-PROMPT** [41]. Система Anchor-PROMPT берет пары связанных терминов из исходных онтологий и проходит по кратчайшему пути, связывающему эти пары. Проводится сравнение терминов на этом пути с целью определения одинаковых терминов и последующей генерации множества пар семантически идентичных концептов.

**CMS (CROSI Mapping System)** [42]. CMS – это система сведения онтологий, основанная на структурном сравнении и использующая богатую семантику OWL-конструкций. Модульная архитектура системы позволяет использовать внешние лингвистические ресурсы, а также множественные стратегии при нахождении меры близости сущностей онтологий.

**FCA-Merge** [43]. FCA-Merge – это метод слияния онтологий, базирующийся на анализе формальных концептов Гантера и Вилли [44]. Весь процесс состоит из трех шагов:

- Извлечение сущностей и генерация формального контекста для каждой онтологии;
- Вычисление решетки концептов с помощью алгоритма TITANIC [45];
- Ручная генерация единой онтологии путем взаимодействия с пользователем, базирующаяся на ранее построенной решетке концептов.

**CHIMAERA** [46]. CHIMAERA – это интерактивный инструментарий для объединения онтологий, а также онтолого-лингвистический редактор. Он позволяет вмешательство пользователя в любой момент в течение процесса объединения. Если находятся лингвистические соответствия, то они обрабатываются автоматически, в противном же случае – принять решение предлагается пользователю.

### **ЗАКЛЮЧЕНИЕ**

Обогащение онтологий является широко изученной проблемой. Ведется разработка многочисленных подходов, направленных на обогащение различных видов аксиом онтологий. Наиболее разработанными направлениями являются нахождение определений классов и построение отображений онтологий.

Выбор того или иного инструментария в значительной степени зависит от того, какую именно задачу предполагается решать.

Для задачи нахождения определений классов наиболее разработанным инструментом является ORE. Дополнительным преимуществом этой системы является наличие плагинов к системам Protege и OntoWiki.

Для выбора инструментария для решения задачи построения отображения между интегрированной и частной онтологиями требуются дополнительные исследования. Представленные системы используют различные подходы, и их эффективность зависит от онтологий, между которыми строится отображение.

Рассматривая задачу построения отображения между двумя частными онтологиями, можно выделить LOM как систему, требующую меньшего вклада труда экспертов, и QOM по причине низкой ресурсоемкости по времени используемого алгоритма.

### **Благодарности**

Исследование выполнено за счет гранта Российского научного фонда, проект № 19-71-10056.



## СПИСОК ЛИТЕРАТУРЫ

1. Auer S., Lehmann J., Ngonga-Ngomo A.-C. Introduction to Linked Data and Its Lifecycle on the Web // Reasoning Web 2011. Lecture Notes in Computer Science. 2011. V. 6848. Springer, Heidelberg. P. 1–75.
2. Nienhuys-Cheng S.-H., de Wolf R. Foundations of Inductive Logic Programming // Lecture Notes in Computer Science. V. 1228. Springer, Heidelberg, 1997. 248 p.
3. Cohen W.W., Borgida A., Hirsh H. // Computing Least Common Subsumers in Expressive Description Logics. In: Foo N. (eds). Advanced Topics in Artificial Intelligence. AI 1999. Lecture Notes in Computer Science. Vol. 1747. Springer, Berlin, Heidelberg. P. 754–760.
4. Cohen W.W., Hirsh H. Learning the CLASSIC description logic: Theoretical and experimental results // Proceedings of the 4<sup>th</sup> International Conference on the Principles of Knowledge Representation and Reasoning (KR-94). San Francisco: Morgan Kaufmann, 1994. P. 121–133.
5. Badea L., Nienhuys-Cheng S.-H. A refinement operator for description logics // In: Cussens J., Frisch A.M. (eds). Inductive Logic Programming: 10th International Conference, ILP 2000. Lecture Notes in Computer Science (LNAI). 2000. Vol. 1866. Springer, Heidelberg. P. 40–59.
6. Esposito F., Fanizzi N., Iannone L. Knowledge-intensive induction of terminologies from metadata // Proceedings of The Third International Semantic Web Conference Proceedings, ISWC 2004. Lecture Notes in Computer Science. 2004. V. 3298. Springer-Verlag, Heidelberg. P. 411–426.
7. Lehmann J., Hitzler P. Foundations of Refinement Operators for Description Logics // In: Blockeel H., Ramon J., Shavlik J., Tadepalli P. (eds). Inductive Logic Programming: 17th International Conference, ILP 2007. Lecture Notes in Computer Science (LNAI). 2008. Vol. 4894. Springer, Heidelberg. P. 161–174.
8. Fanizzi N., d'Amato C., Esposito F. DL-FOIL Concept Learning in Description Logics // Inductive Logic Programming: 18th International Conference, ILP 2008. Lecture Notes in Computer Science. 2008. V. 5194. Springer, Heidelberg. P. 107–121.
9. Baader F., Sertkaya B., Turhan A.Y. Computing the Least Common Subsumer w.r.t. a Background Terminology. In: Alferes J.J., Leite J. (eds). Logics in

Artificial Intelligence. JELIA 2004. Lecture Notes in Computer Science. 2004. Vol. 3229. Springer, Berlin, Heidelberg. P. 400–412.

10. *Quinlan J.R.* Learning Logical Definitions from Relations // Machine Learning. 1990. V. 5. P. 239–266.

11. *Iannone L., Palmisano I., Fanizzi N.* An algorithm based on counterfactuals for concept learning in the Semantic Web // Applied Intelligence. 2007. Vol. 26. P. 139–159.

12. *Lehmann J., Hitzler P.* A refinement operator based learning algorithm for the ALC description logic // In: Blockeel H., Ramon J., Shavlik J., Tadepalli P. (eds). Inductive Logic Programming: 17th International Conference, ILP 2007. Lecture Notes in Computer Science (LNAI). 2008. Vol. 4894. Springer, Heidelberg. P. 147–160.

13. *Lehmann J., Bühmann L.* ORE – A Tool for Repairing and Enriching Knowledge Bases // In: Patel-Schneider P.F. et al. (eds) The Semantic Web – ISWC 2010. ISWC 2010. Lecture Notes in Computer Science. 2010. Vol. 6497. Springer, Berlin, Heidelberg. P. 177–193.

14. *Lehmann J., Auer S., Bühmann L.* Class expression learning for ontology engineering // Journal of Web Semantics. 2011. Vol. 9. P. 71–81.

15. *Lisi F.A.* Building rules on top of ontologies for the semantic web with inductive logic programming // Theory and Practice of Logic Programming. 2008. Vol. 8(3). P. 271–300.

16. *Blomqvist E.* Semi-automatic Ontology Construction based on Patterns. Ph.D. thesis. Linköping University. 2009.

17. *Volker J., Vrandečić D., Sure Y.* Learning Disjointness // In: Franconi E., Kifer M., May W. (eds). 4th European Semantic Web Conference, ESWC 2007. Lecture Notes in Computer Science (LNAI). 2007. Vol. 4894. Springer, Heidelberg. P. 175–189.

18. *Rudolph S.* Exploring relational structures via FLE // In: Wolff K.E., Pfeiffer H.D., Delugach H.S. (eds) // 12th International Conference on Conceptual Structures, ICCS 2004. Lecture Notes in Computer Science (LNAI). 2004. Vol. 3127. Springer, Heidelberg. P. 196–212.

19. *Sertkaya B.* OntocomP system description // In: Grau B.C., Horrocks I., Motik B., Sattler U. (eds.) Proceedings of the 22nd International Workshop on

Description Logics (DL 2009), Oxford, UK, July 27-30. CEUR Workshop Proceedings. Vol. 477. CEUR-WS.org (2009).

20. *Völker J., Rudolph S.* Fostering Web Intelligence by Semi-automatic OWL Ontology Refinement // IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW. 2008. P. 454–460. doi: 10.1109/WIIAT.2008.36.

21. RELExO (Relational Exploration for Learning Expressive Ontologies). URL: <http://code.google.com/p/relexo/>.

22. *Choi N., Song I.-Y., Han H.* A survey on ontology mapping // SIGMOD Record. 2006. Vol. 35(3). P. 34–41.

23. *Doan A., Domingos P., Halevy A.* Learning to Match the Schemas of Data Sources: A Multistrategy Approach // Machine Learning. 2003. Vol. 50 (3). P. 279–301.

24. *Beneventano D., Bergamaschi S., Guerra F.* Synthesizing an Integrated Ontology // IEEE Internet Computing. 2003. Vol. 7(5). P.42–51.

25. *Calvanese D., De Giacomo G., Lenzerini M.* A Framework for Ontology Integration // Proceedings of the 1st International Semantic Web Working Symposium (SWWS). 2001. P. 303–317.

26. *Silva N., Rocha J.* Ontology Mapping for Interoperability in Semantic Web // Proceedings of the IADIS International Conference WWW/Internet 2003, ICWI 2003. 2003. P. 603–610.

27. *Bouquet P., Giunchiglia F., van Harmelen F.* C-OWL: Contextualizing Ontologies // The Semantic Web – ISWC 2003, Second International Semantic Web Conference, 2003. Lecture Notes in Computer Science. 2003. Vol. 2870. P. 164–179.

28. *Bouquet P., Serafini L., Zanobini S.* Semantic Coordination: A New Approach and an Application // The Semantic Web - ISWC 2003, Second International Semantic Web Conference, 2003. Lecture Notes in Computer Science. 2003. Vol. 2870. P.130–145.

29. *Doan A., Madhavan J., Dhamankar R. et al.* Learning to match ontologies on the Semantic Web // VLDB. 2003. Vol. 12. P. 303–319. <https://doi.org/10.1007/s00778-003-0104-2>.

30. *Silva N., Rocha J.* Semantic Web complex ontology mapping // Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003).2003. P. 82-88. doi: 10.1109/WI.2003.1241177.

31. *Li J.* LOM: A Lexicon-based Ontology Mapping Tool // Proceedings of the Performance Metrics for Intelligent Systems. 2004. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.5379&rep=rep1&type=pdf>
32. *Ehrig M., Staab S.* QOM – Quick Ontology Mapping // Proceedings of The Third International Semantic Web Conference Proceedings, ISWC 2004. Lecture Notes in Computer Science. 2004. Vol. 3298. P. 683–697.
33. *Mitra P., Wiederhold G.* Resolving Terminological Heterogeneity in Ontologies // Proceedings of the ECAI'02 workshop on Ontologies and Semantic Interoperability. 2002. P. 45–50.
34. *Maedche A., Motik B., Stojanovic L., Studer R., Volz R.* Ontologies for Enterprise Knowledge Management // IEEE Intelligent Systems. 2003. Vol. 18(2). P. 26–33.
35. *Mitra P., Noy N. F., Jaiswals A.* OMEN: A Probabilistic Ontology Mapping Tool // Proceedings of International Semantic Web Conference, ISWC- 2005. 2005. P. 537–547.
36. *Besana P., Robertson D., Rovatsos M.* Exploiting interaction contexts in P2P ontology mapping // Proceedings of 2nd International Workshop on Peer to Peer Knowledge Management. 2005. CEUR Workshop Proceedings. P. 1613–1673. [CEUR-WS.org/Vol-139/2.pdf](http://ceur-ws.org/Vol-139/2.pdf).
37. *Noy N.F., Musen M.A.* PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment // Proceedings of the National Conference on Artificial Intelligence (AAAI/IAAI). 2000. P.450–455.
38. *Noy N.F., Musen M.A.* Smart: Automated Support for Ontology Merging and Alignment // Proceedings of the 12th Workshop on Knowledge, Acquisition Modeling and Management (IKAW 1999). 1999. P. 1–20.
39. *Chalupsky H.* Ontomorph: A Translation System for Symbolic Knowledge // Proceedings of the 7th international Conference on Principles of Knowledge Representation and Reasoning. 2000. P. 471–482.
40. *Ichise R., Takeda H., Honiden S.* Rule Induction for Concept Hierarchy Alignment // Proceedings of the Workshop on Ontology Learning at the 17th International Joint Conference on Artificial Intelligence (IJCAI). 2001. [http://ceur-ws.org/Vol-38/ichise\\_IJCAI-OL.pdf](http://ceur-ws.org/Vol-38/ichise_IJCAI-OL.pdf).

41. *Noy N.F., Musen M.A.* Anchor-PROMPT: Using Non-Local Context for Semantic Matching // Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI). 2001. <http://dit.unitn.it/~accord/RelatedWork/Matching/noy.pdf>.

42. *Kalfoglou Y., Hu B.* CROSI Mapping System (CMS) Results of the 2005 Ontology Alignment Contest // Proceedings of the K-CAP 2005 Workshop of Integrating Ontologies. 2005. P. 77–85.

43. *Stumme G., Maedche A.* FCA-Merge: Bottom-Up Merging of Ontologies // Proceeding of the International Joint Conference on Artificial Intelligence IJCAI-01. Seattle: 2001. P. 205–234.

44. *Ganter B., Wille R.* Formal Concept Analysis. Mathematical Foundations. Berlin: Springer, 1999. 284 p.

45. Titanic: Machine Learning Algorithms. URL: <https://www.kaggle.com/berhag/titanic-machine-learning-algorithms>.

46. *McGuinness D., Fikes R., Rice J., Wilder S.* The Chimaera Ontology Environment // Proceedings of the 17th National Conference on Artificial Intelligence (AAAI). 2000. P. 1123–1124.

---

## METHODS AND ALGORITHMS FOR INCREASING LINKED DATA EXPRESSIVENESS (OVERVIEW)

O. A. Nevzorova

Kazan Federal University

onevzoro@gmail.com

### **Abstract**

This review discusses methods and algorithms for increasing linked data expressiveness which are prepared for Web publication. The main approaches to the enrichment of ontologies are considered, the methods on which they are based and the tools for implementing the corresponding methods are described.

The main stage in the general scheme of the related data life cycle in a cloud of Linked Open Data is the stage of building a set of related RDF- triples. To improve the classification of data and the analysis of their quality, various methods are used to increase the expressiveness of related data. The main ideas of these methods are concerned with the enrichment of existing ontologies (an expansion of the basic scheme of knowledge) by adding or improving terminological axioms. Enrichment methods are based on methods used in various fields, such as knowledge representation, machine learning, statistics, natural language processing, analysis of formal concepts, and game theory.

**Keywords:** *linked data, ontology, ontology enrichment, semantic web*

### **REFERENCES**

1. Auer S., Lehmann J., Ngonga-Ngomo A.-C. Introduction to Linked Data and Its Lifecycle on the Web // Reasoning Web 2011. Lecture Notes in Computer Science. 2011. V. 6848. Springer, Heidelberg. P. 1–75.
2. Nienhuys-Cheng S.-H., de Wolf R. Foundations of Inductive Logic Programming // Lecture Notes in Computer Science. V. 1228. Springer, Heidelberg, 1997. 248 p.
3. Cohen W.W., Borgida A., Hirsh H. // Computing Least Common Subsumers in Expressive Description Logics. In: Foo N. (eds). Advanced Topics in Artificial

Intelligence. AI 1999. Lecture Notes in Computer Science. Vol. 1747. Springer, Berlin, Heidelberg. P. 754–760.

4. *Cohen W.W., Hirsh H.* Learning the CLASSIC description logic: Theoretical and experimental results // Proceedings of the 4<sup>th</sup> International Conference on the Principles of Knowledge Representation and Reasoning (KR-94). San Francisco: Morgan Kaufmann, 1994. P. 121–133.

5. *Badea L., Nienhuys-Cheng S.-H.* A refinement operator for description logics // In: Cussens J., Frisch A.M. (eds). Inductive Logic Programming: 10th International Conference, ILP 2000. Lecture Notes in Computer Science (LNAI). 2000. Vol. 1866. Springer, Heidelberg. P. 40–59.

6. *Esposito F., Fanizzi N., Iannone L.* Knowledge-intensive induction of terminologies from metadata // Proceedings of The Third International Semantic Web Conference Proceedings, ISWC 2004. Lecture Notes in Computer Science. 2004. V. 3298. Springer-Verlag, Heidelberg. P. 411–426.

7. *Lehmann J., Hitzler P.* Foundations of Refinement Operators for Description Logics // In: Blockeel H., Ramon J., Shavlik J., Tadepalli P. (eds). Inductive Logic Programming: 17th International Conference, ILP 2007. Lecture Notes in Computer Science (LNAI). 2008. Vol. 4894. Springer, Heidelberg. P. 161–174.

8. *Fanizzi N., d'Amato C., Esposito F.* DL-FOIL Concept Learning in Description Logics // Inductive Logic Programming: 18th International Conference, ILP 2008. Lecture Notes in Computer Science. 2008. V. 5194. Springer, Heidelberg. P. 107–121.

9. *Baader F., Sertkaya B., Turhan A.Y.* Computing the Least Common Subsumer w.r.t. a Background Terminology. In: Alferes J.J., Leite J. (eds). Logics in Artificial Intelligence. JELIA 2004. Lecture Notes in Computer Science. 2004. Vol. 3229. Springer, Berlin, Heidelberg. P. 400–412.

10. *Quinlan J.R.* Learning Logical Definitions from Relations // Machine Learning. 1990. V. 5. P. 239–266.

11. *Iannone L., Palmisano I., Fanizzi N.* An algorithm based on counterfactuals for concept learning in the Semantic Web // Applied Intelligence. 2007. Vol. 26. P. 139–159.

12. *Lehmann J., Hitzler P.* A refinement operator based learning algorithm for the ALC description logic // In: Blockeel H., Ramon J., Shavlik J., Tadepalli P. (eds).

Inductive Logic Programming: 17th International Conference, ILP 2007. Lecture Notes in Computer Science (LNAI). 2008. Vol. 4894. Springer, Heidelberg. P. 147–160.

13. *Lehmann J., Bühmann L.* ORE – A Tool for Repairing and Enriching Knowledge Bases // In: Patel-Schneider P.F. et al. (eds) The Semantic Web – ISWC 2010. ISWC 2010. Lecture Notes in Computer Science. 2010. Vol. 6497. Springer, Berlin, Heidelberg. P. 177–193.

14. *Lehmann J., Auer S., Bühmann L.* Class expression learning for ontology engineering // Journal of Web Semantics. 2011. Vol. 9. P. 71–81.

15. *Lisi F.A.* Building rules on top of ontologies for the semantic web with inductive logic programming // Theory and Practice of Logic Programming. 2008. Vol. 8(3). P. 271–300.

16. *Blomqvist E.* Semi-automatic Ontology Construction based on Patterns. Ph.D. thesis. Linköping University. 2009.

17. *Volker J., Vrandečić D., Sure Y.* Learning Disjointness // In: Franconi E., Kifer M., May W. (eds). 4th European Semantic Web Conference, ESWC 2007. Lecture Notes in Computer Science (LNAI). 2007. Vol. 4894. Springer, Heidelberg. P. 175–189.

18. *Rudolph S.* Exploring relational structures via FLE // In: Wolff K.E., Pfeiffer H.D., Delugach H.S. (eds) // 12th International Conference on Conceptual Structures, ICCS 2004. Lecture Notes in Computer Science (LNAI). 2004. Vol. 3127. Springer, Heidelberg. P. 196–212.

19. *Sertkaya B.* OntocomP system description // In: Grau B.C., Horrocks I., Motik B., Sattler U. (eds.) Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30. CEUR Workshop Proceedings. Vol. 477. CEUR-WS.org (2009).

20. *Völker J., Rudolph S.* Fostering Web Intelligence by Semi-automatic OWL Ontology Refinement // IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW. 2008. P. 454–460. doi: 10.1109/WIIAT.2008.36.

21. RELExO (Relational Exploration for Learning Expressive Ontologies). URL: <http://code.google.com/p/relexo/>.

22. *Choi N., Song I.-Y., Han H.* A survey on ontology mapping // SIGMOD Record. 2006. Vol. 35(3). P. 34–41.



23. *Doan A., Domingos P., Halevy A.* Learning to Match the Schemas of Data Sources: A Multistrategy Approach // *Machine Learning*. 2003. Vol. 50 (3). P. 279–301.
24. *Beneventano D., Bergamaschi S., Guerra F.* Synthesizing an Integrated Ontology // *IEEE Internet Computing*. 2003. Vol. 7(5). P.42–51.
25. *Calvanese D., De Giacomo G., Lenzerini M.* A Framework for Ontology Integration // *Proceedings of the 1st International Semantic Web Working Symposium (SWWS)*. 2001. P. 303–317.
26. *Silva N., Rocha J.* Ontology Mapping for Interoperability in Semantic Web // *Proceedings of the IADIS International Conference WWW/Internet 2003, ICWI 2003*. 2003. P. 603–610.
27. *Bouquet P., Giunchiglia F., van Harmelen F.* C-OWL: Contextualizing Ontologies // *The Semantic Web – ISWC 2003, Second International Semantic Web Conference, 2003. Lecture Notes in Computer Science*. 2003. Vol. 2870. P. 164–179.
28. *Bouquet P., Serafini L., Zanobini S.* Semantic Coordination: A New Approach and an Application // *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, 2003. Lecture Notes in Computer Science*. 2003. Vol. 2870. P.130–145.
29. *Doan A., Madhavan J., Dhamankar R. et al.* Learning to match ontologies on the Semantic Web // *VLDB*. 2003. Vol. 12. P. 303–319. <https://doi.org/10.1007/s00778-003-0104-2>.
30. *Silva N., Rocha J.* Semantic Web complex ontology mapping // *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*.2003. P. 82-88. doi: 10.1109/WI.2003.1241177.
31. *Li J.* LOM: A Lexicon-based Ontology Mapping Tool // *Proceedings of the Performance Metrics for Intelligent Systems*. 2004. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.5379&rep=rep1&type=pdf>
32. *Ehrig M., Staab S.* QOM – Quick OntologyMapping // *Proceedings of The Third International Semantic Web Conference Proceedings, ISWC 2004. Lecture Notes in Computer Science*. 2004. Vol. 3298. P. 683–697.
33. *Mitra P., Wiederhold G.* Resolving Terminological Heterogeneity in Ontologies // *Proceedings of the ECAI'02 workshop on Ontologies and Semantic Interoperability*. 2002. P. 45–50.

34. *Maedche A., Motik B., Stojanovic L., Studer R., Volz R.* Ontologies for Enterprise Knowledge Management // IEEE Intelligent Systems. 2003. Vol. 18(2). P. 26–33.
35. *Mitra P., Noy N. F., Jaiswals A.* OMEN: A Probabilistic Ontology Mapping Tool // Proceedings of International Semantic Web Conference, ISWC- 2005. 2005. P. 537–547.
36. *Besana P., Robertson D., Rovatsos M.* Exploiting interaction contexts in P2P ontology mapping // Proceedings of 2nd International Workshop on Peer to Peer Knowledge Management. 2005. CEUR Workshop Proceedings. P. 1613–1673. CEUR-WS.org/Vol-139/2.pdf.
37. *Noy N.F., Musen M.A.* PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment // Proceedings of the National Conference on Artificial Intelligence (AAAI/IAAI). 2000. P.450–455.
38. *Noy N.F., Musen M.A.* Smart: Automated Support for Ontology Merging and Alignment // Proceedings of the 12th Workshop on Knowledge, Acquisition Modeling and Management (IKAW 1999). 1999. P. 1–20.
39. *Chalupsky H.* Ontomorph: A Translation System for Symbolic Knowledge // Proceedings of the 7th international Conference on Principles of Knowledge Representation and Reasoning. 2000. P. 471–482.
40. *Ichise R., Takeda H., Honiden S.* Rule Induction for Concept Hierarchy Alignment // Proceedings of the Workshop on Ontology Learning at the 17th International Joint Conference on Artificial Intelligence (IJCAI). 2001. [http://ceur-ws.org/Vol-38/ichise\\_IJCAI-OL.pdf](http://ceur-ws.org/Vol-38/ichise_IJCAI-OL.pdf).
41. *Noy N.F., Musen M.A.* Anchor-PROMPT: Using Non-Local Context for Semantic Matching // Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI). 2001. <http://dit.unitn.it/~accord/RelatedWork/Matching/noy.pdf>.
42. *Kalfoglou Y., Hu B.* CROSI Mapping System (CMS) Results of the 2005 Ontology Alignment Contest // Proceedings of the K-CAP 2005 Workshop of Integrating Ontologies. 2005. P. 77–85.

43. *Stumme G., Maedche A.* FCA-Merge: Bottom-Up Merging of Ontologies // Proceeding of the International Joint Conference on Artificial Intelligence IJCAI-01. Seattle: 2001. P. 205–234.

44. *Ganter B., Wille R.* Formal Concept Analysis. Mathematical Foundations. Berlin: Springer, 1999. 284 p.

45. Titanic: Machine Learning Algorithms. URL: <https://www.kaggle.com/berhag/titanic-machine-learning-algorithms>.

46. *McGuinness D., Fikes R., Rice J., Wilder S.* The Chimaera Ontology Environment // Proceedings of the 17th National Conference on Artificial Intelligence (AAAI). 2000. P. 1123–1124.

### **СВЕДЕНИЯ ОБ АВТОРЕ**



**НЕВЗОРОВА Ольга Авенировна** – доцент кафедры информационных систем Института вычислительной математики и информационных технологий Казанского федерального университета, к. т. н. Основные направления научных исследований: обработка естественного языка, искусственный интеллект.

**Olga Avenirovna NEVZOROVA** – Kazan Federal University, Institute of Computer Mathematics and Information Technologies, Associated Professor the Department of Information Systems, PhD. Major Fields of Scientific Research: Natural language processing, Artificial intelligence.

e-mail: [onevzoro@gmail.com](mailto:onevzoro@gmail.com)

*Материал поступил в редакцию 27 апреля 2020 года*