Интеграция библиографических данных в Linked Open Data

© Д. А. Малахов © В. А. Серебряков © К. Б. Теймуразов © О. Н. Шорин
Вычислительный центр им. А. А. Дородницына РАН,
Москва
malahov@iqbuzz.ru seerebr@ccas.ru kbt@ccas.ru shorin@nlr.ru

Аннотация

В данной работе рассматривается проблема интеграции библиографических записей. Была поставлена задача объединить данные Российской национальной библиотеки и Британской национальной библиотеки в рамках пространства Linked Open Data. В ходе решения задачи была построена прототипная система, с помощью которой данные формата RUSMARC могут быть опубликованы в Linked Open Data и связаны с другими библиотечными данными. Были проведены опыты и получены оценки работы подсистемы связывания данных разных источников.

1 Введение

1.1 Семантическая паутина

Интернет с самого начала представлял собой множество разрозненных сайтов, никак не связанных друг с другом по смыслу. С течением времени появилось все больше и больше ресурсов, посвященных одним и тем же проблемам. Поиск нужной информации становился все более затруднительным, в то же время росли требования к качеству поиска.

Появилась потребность в семантическом поиске, поиске не по словам, а по смыслу, а также в связывании данных близких по смыслу, но находящихся в разных ресурсах. Стало ясно, что существующие стандарты не в состоянии удовлетворить потребности людей, необходимо было создавать новые стандарты создания структурированных данных, по которым возможен семантический поиск.

Термин Semantic Web был впервые введён сэром Тимом Бернерсом-ли в журнале «Scientific American», и называется им «следующим шагом в развитии Всемирной паутины». Концепция Semantic Web была принята и продвигается W3C (Консорциумом Всемирной паутины).

Идея этой концепции - создать общепринятый способ совместного использования данных различными приложениями, организациями и сообществами, и предоставление возможности получать данные, как вручную, так и автоматическими средствами [1].

Для поддержки этой концепции W3C создал стандарты, понятия, технологии и форматы. В них входят URI (Uniform Resource Identifier), RDF(Resource Description Framework), OWL(Web Ontology Language), SPARQL (Protocol and RDF Query Language).

URI (Uniform Resource Identifier) – последовательность символов, идентифицирующая абстрактный или физический ресурс.

RDF – модель данных, служащая платформой для представления информации. Структура, лежащая в основе любых выражений в RDF, это коллекция триплетов, каждый из которых состоит из субъекта, предиката и объекта. Набор таких триплетов называется RDF-графом. По своей природе это ориентированный помеченный мульти-граф. Каждый триплет представляет объявление отношения между предметами. Выражение RDF триплета говорит о том, что некоторое отношение, указанное предикатом, связывает предметы, обозначенные как субъект и объект, в триплете. Узлами RDF-графа являются объекты и субъекты. Узлы обычно идентифицируется с помощью URI, однако бывают пустые и литеральные узлы. Дуги (предикаты) всегда идентифицируются с помощью URI [2].

OWL – это язык для определения и представления онтологий. Онтология предназначена для описания семантики данных. Она может включать описания классов, свойств, экземпляров классов, их операций. Формальная семантика OWL описывает, как получать логические следствия, имея такую онтологию, т.е. получить факты, которые не представлены в онтологии буквально, но следуют из ее семантики. При построении логических выводов используется модель открытого мира, т.е. если не может быть доказано, что некое утверждение истинно, из этого не следует, что оно ложно [3].

Труды 16-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2014, Дубна, Россия, 13–16 октября 2014 г.

Кроме OWL для описания онтологий также используют язык RDFS (Resource Description Framework Schema). Как правило OWL и RDFS используются совместно.

SPARQL – язык запросов для обращений к RDF-хранилищам, служит тем же целям, что и SQL в области реляционных баз данных. SPARQL точка доступа - сервер обрабатывающий запросы.

1.2 Linked Open Data

LOD (Linked Open Data) – проект, целью которого является наполнение сети Интернет данными в стандартных форматах Semantic Web, а также устанавливание связей между данными из различных источников [4].

Тим Бернерс-Ли сформулировал следующие четыре принципа связанных данных [5]:

– Использование URI для идентификации сущностей.

– Использование HTTP URI, чтобы эти сущности могли быть найдены людьми.

– При обращении по URI предоставлять полезную информацию о сущности, используя стандартизованные форматы (RDF, SPARQL).

– Предоставлять также другие, связанные URI, для облегчения поиска.

На данный момент опубликовано более 40 млрд троек в рамках этого проекта. Самым крупным источником является DBPedia, более 3,5 млн сущностей, извлеченных из проекта Wikipedia.

1.3 Интеграция данных

В каждой предметной области существует много разрозненных источников. Каждая организация может оперировать только той информацией, которая у нее есть. Задача сбора информации часто бывает нетривиальной. Интеграция с пространством Linked Open Data является одним из универсальных решений данной задачи.

Linked Open Data было создано для того, чтобы в каждой предметной области интегрировать внутри себя как можно больше информации. Таким образом, публикуя данные в этом пространстве, мы с одной стороны получаем доступ ко всей информации, которая нас интересуют через свои данные, а с другой даем доступ к своей информации.

2 Постановка задачи

Целью данной работы является интеграция и обогащение библиографических записей, предоставленных РНБ (Российская Национальная Библиотека), с данными БНБ (Британская Национальная Библиотека). Набор данных РНБ насчитывает несколько миллионов библиографических записей. Для интеграции был предоставлен тестовый набор данных (около 17 тыс. единиц). Набор данных БНБ насчитывает 3,5 млн единиц, он опубликован в LOD.

Для достижения цели нужно решить задачи:

– Опубликовать данные РНБ согласно принципам LOD.

– Связать данные РНБ с опубликованными в LOD данными БНБ.

2.1 Публикация данных

Для решения этой задачи нужно решить подзадачи:

– Описать предметную область.

– Конвертировать данные РНБ в RDF.

– Настроить семантическое хранилище RDF данных РНБ.

– Предоставить доступ к данным РНБ.

2.1.1 Описание предметной области

Необходимо выбрать термины из существующих онтологий, и на их основании создать собственную онтологию. Если найдутся данные, которые нельзя представить в рамках существующих терминов, создать собственные термины и дополнить онтологию.

2.1.2 Конвертация данных

РНБ предоставила данные в формате RUSMARC. Для того, чтобы опубликовать данные согласно принципам LOD, они должны быть представлены в формате RDF с использованием терминов составленной онтологии. Нужно создать инструмент, переводящий RUSMARC формат в RDF.

2.1.3 Создание семантического хранилища

SPARQL точка доступа – сервер, принимающий запросы на языке SPARQL и выдающий данные в формате RDF.

Публикация данных в LOD подразумевает создание семантического хранилища и SPARQL точки доступа, привязанной к нему. Необходимо выбрать одно из существующих семантических хранилищ, загрузить в него данные и настроить логический вывод.

2.1.4 Предоставление доступа к данным

Нужно создать web-сервер, который выдавал бы информацию об объектах хранилища по HTTP запросу, отправленному на URI объекта.

2.2 Связывание

Согласно принципам LOD нужно задать как можно больше связей между данными РНБ и данными БНБ. Кроме того необходимо связать данные РНБ между собой.

Итак, следует разработать и реализовать алгоритм связывания разных библиографических

записей и сохранить полученные связи в семантическом хранилище. Необходимо учитывать связи при обработке запросов пользователя.

3 Описание предметной области

3.1 Общий обзор

Библиографическая запись – это элемент библиографической информации, фиксирующий сведения о документе – объекте записи, позволяющие его идентифицировать, раскрыть его состав и содержание в целях библиографического поиска [6].

Библиографическая запись включает в себя следующие части:

– заголовок;

– классификационные индексы;

– аннотация;

– язык;

– издательство;

– авторы;

– другая дополнительная информация.

3.2 Публикация библиографических записей в LOD

Одна из задач библиотеки – предоставление и обработка информации о всевозможных публикациях, а именно метаданные этих публикаций. К ним относятся: описание публикации, информация об авторе, издательстве и т.д. Поэтому интеграция данных различных библиотек является довольно актуальной проблемой.

Публикация данных в LOD вызывает огромный интерес в библиотечном сообществе, т.к. имеет ряд преимуществ по сравнению с другими способами обмена данными между библиотеками:

– В LOD для идентификации ресурса используют единое глобальное пространство имен, объекты идентифицируются с помощью URI, который является уникальным для всего LOD.

– Высокая способность к масштабированию, т.к. не обязательно хранить все данные об определенном объекте в одном хранилище или в одном источнике.

– Обмен данными можно осуществлять порциями, в виде множества законченных утверждений, необязательно единовременно передавать всю существующую информацию об объекте, т.к. ее можно получить в любой момент, когда она потребуется.

– В семантических хранилищах имеется логический вывод, это упрощает интеграцию данных с разными схемами данных (достаточно определить соответствия между схемами и создать связи между объектами) и позволяет выводить новые знания на основе имеющихся (нет необходимости хранить данные, которые могут быть получены с помощью логического вывода).

Несмотря на все плюсы, не так много библиотек внедряют подобные решения. В докладе Инкубаторной группы W3C по библиотечной модели LOD опубликованы причины, мешающие развитию этой области [7]:

– Опубликованные в интернете библиотечные данные слабо связаны между собой.

– Библиотечные стандарты создавались только для библиотечного сообщества.

– Библиотечные данные слабо структурированы и преимущественно хранятся на естественном языке.

– Библиотечное сообщество и сообщество Semantic Web имеют разные терминологии для аналогичных концепций.

– Библиотеки зависят от развития систем поставщиков, и часто не могут по собственной инициативе публиковать данные в LOD.

3.3 Форматы представления библиотечных данных

Библиографические записи, как правило, представляются и хранятся в библиотеках в одном из форматов семейства MARC.

Плюсы семейства форматов MARC:

– Достаточно детальное описание записи.

– Формат внедрен повсеместно и это упрощает обмен записями.

Минусы семейства форматов MARC:

– Запись можно хранить только полностью.

– Существует несколько форматов этого семейства плохо совместимых между собой.

– Для некоторых задач форматы семейства MARC избыточны, что влечет за собой избыточную бюрократию.

– Не может быть использован для представления данных в семантических базах данных.

Существует два основных формата семейства MARC:

– MARC21(используется, как правило, в США);

– UNIMARC (международный стандарт).

RUSMARC является диалектом UNIMARC.

Записи в формате семейства MARC могут быть представлены в виде XML(MARC/XML) или в бинарной форме(MARC/bin).

Кроме MARC существуют другие способы хранения данных, такие как представление данных согласно схемам Dublin Core или MODS.

Схема Dublin Core представляет из себя набор элементов данных для описания документов и других объектов в Интернете. Благодаря своей компактности и простоте схема стала широко распространена. При разработке Dublin Core не предполагалось, что новая схема полностью заменит MARC, т.к. Dublin Core не обеспечивает такую полноту, как MARC. Но для многих задач использование Dublin Core достаточно. Кроме того, существует онтология, описывающая термины

Dublin Core. Согласно этим терминам данные могут быть представлены в виде RDF.

Схема MODS (Metadata Object Description Standard) разработана Библиотекой Конгресса, является упрощенной версией MARC. Вместо трехзначных меток полей, абстрактных идентификаторов подполей используются понятные для пользователя вербальные метки (например, «title» вместо «245»). Часть элементов MARC игнорируется, введены новые элементы. MODS создана на основе MARC21 и более детальна по сравнению с Dublin Core. На основе MODS была создана онтология, используя термины которой, можно представлять библиографическую запись в виде RDF [8].

Таким образом, используя термины Dublin Core или MODS, можно хранить библиографические записи в виде RDF, используя семантическую базу данных.

3.4 Проекты интеграции библиотечных данных

На данный момент существует несколько проектов по интеграции данных разных библиотек, одними из крупнейших являются VIAF и Europeana.

VIAF (The Virtual Internation Authority File) – виртуальная система международных стандартов для авторитетной информации, совместный проект Библиотеки Конгресса, Немецкой Национальной Библиотеки и ряда других национальных библиотек и организаций. В состав проекта входит более 20 организаций, в том числе РНБ. В рамках этого проекта планируется интегрировать информацию об авторитетных файлах из крупнейших библиотек в мире.

Europeana – европейская цифровая библиотека, цель которой – обеспечить доступ к отсканированным страницам книг, отражающих различные аспекты европейской культуры. Сейчас доступна информация на французском, немецком и английском языках. В проекте участвуют Франция, Великобритания, Испания и Германия.

3.5 Выводы

Из описанного выше можно сделать вывод, что интеграция библиотечных данных достаточно актуальная задача, а использование технологии Semantic Web для этой задачи является перспективным. Существует несколько вариантов представления библиографических записей в виде RDF. Появляется все больше и больше успешных проектов в этой области.

4 Исследование и построение решения задачи

4.1 Публикация данных

4.1.1 Описание предметной области

Для того чтобы преобразовать данные в RDF нужна схема, которая описывается с помощью RDFS и OWL. Эта схема называется онтологией. По концепции LOD онтология должна быть составлена на основе существующих в LOD онтологиях. Данные по библиографическим записям, которые предоставила РНБ, покрываются терминами Dublin Core, а информация об авторе терминами FOAF. БНБ представляет свои данные в Dublin Core и FOAF. Таким образом, имеет смысл использовать не MODS, а Dublin Core. Оно и было выбрано для дальнейшей работы.

4.1.2 Конвертация данных

РНБ предоставляет свои данные в формате RUSMARC/bin. Общее количество записей около 17 тысяч. Необходимо преобразовать их в RDF. Эта задача состоит из двух подзадач:

– Преобразовать данные из бинарного формата RUSMARC/bin в RUSMARC/xml.

– Преобразовать данные из RUSMARC/xml в RDF с помощью онтологии.

Для решения первой подзадачи РНБ предоставила программу на C#, которую следует доработать.

Для решения второй подзадачи следует написать XSLT шаблон на основании онтологии.

4.1.3 Создание семантического хранилища

Семантическое хранилище – это набор программных средств позволяющих хранить RDF данные и манипулировать ими с помощью SPARQL запросов.

Существует два вида хранилищ семантических данных:

– Хранилище, основанное на реляционной БД, при этом эффективно используется дисковое пространство и память, но получается низкая производительность.

– Хранилище, основанное на TDB, при этом достигается высокая производительность, но дисковое пространство и память значительно расходуются.

Было выбрано TDB хранилище, так как публикуемые данные имеют большой размер, производительность в нашем случае важнее.

Также существует несколько библиотек для работы с хранилищами.

– Jena;

– Sesame;

– Virtuoso.

Все эти библиотеки похожи друг на друга, но Jena имеет возможность осуществлять логический вывод OWL, а все остальные ограничиваются логическим выводом RDFS. Кроме того для Jena существует более эффективная реализация логического вывода OWL в библиотеке Pellet.

Таким образом, были выбраны Jena + Pellet.

4.1.4 Предоставление доступа к данным

Каждая библиографическая запись, представленная в RDF имеет свой URI. Согласно принципам LOD, при HTTP запросе по этому URI пользователь должен получать полную информацию об этой записи. Это касается и авторов. Для этого нужно создать web-сервер, который будет иметь доступ к семантическому хранилищу, используя библиотеку Jena, и доставать из него всю информацию по полученному URI. Этот сервер должен быть размещен по тому адресу, куда ссылаются записи. Для этого перед наполнением семантического хранилища нужно указать в качестве базового URI в файле RDF адрес сервера.

В качестве web-сервера был выбран сервер Jetty, написанный на Java, т.к. Jena написана на Java, а сервер Jetty встраивается в приложение, и для него ничего не надо дополнительно устанавливать.

4.2 Связывание

Для того чтобы связать данные РНБ и БНБ нужно получить связи, сохранить их в RDF и создать семантическое хранилище. Кроме того на web-сервере необходимо учитывать связи, иметь доступ к ним и возвращать их пользователю при обращении к сущностям, к которым эти связи относятся.

Самый тривиальный способ создания связей – сравнить каждый элемент с каждым по какому-то правилу и получить набор связей. У этого подхода есть два минуса при большом объеме данных:

– потенциальный набор связей $\frac{n(n-1)}{2}$;

– количество сравнений $\frac{n(n-1)}{2}$.

Количество записей предоставленных БНБ около 3,5 млн. В текущей ситуации решение, описанное выше, не эффективно.

Связи могут быть созданы с помощью кластеризации. Две записи будут считаться связанными, если попадают в один кластер. В этом случае, очевидно, количество связей, которые надо хранить заметно уменьшается.

В классической кластеризации количество сравнений $\frac{n(n-1)}{2}$.

В потоковой кластеризации не нужно сравнивать каждый элемент с каждым достаточно сравнить элемент с каждым из кластеров, в итоге количество сравнений получается $O(n)$, где n – количество элементов [9]. Было решено воспользоваться именно потоковой кластеризацией.

Готовых библиотек найдено не было. В учебных целях было решено разработать и реализовать алгоритм самостоятельно.

Для получения кластеров записей РНБ и БНБ необходимо записи РНБ разбить на кластеры и по полученным кластерам распределить записи БНБ.

Основная идея алгоритма заключается в том, что запись и набор записей можно представить в виде вектора лексем, где для каждой лексемы задано некоторое число. Библиографическая запись – вектор лексем, где каждая лексема характеризуется числом, пропорционально зависимым от веса лексемы и частоты ее представления в записи. Кластер – вектор лексем, представленный суммой векторов, включенных в кластер записей.

При представлении записи в виде лексем не учитываются стоп-слова и окончания слов.

Кроме того, необходима функция кластеризации, сравнивающая вектор кластера и вектор записи, а также векторы записей между собой и векторы кластеров между собой.

4.2.1 Кластеризация записей РНБ

Первоначально имеется набор векторов записей. Последовательно проходя по ним, строится набор кластеров:

1) если кластеров нет, то вектор записи становится первым вектором кластера;

2) для всех кластеров определяем близость записи к ним, если запись не попала ни в один кластер, то создается новый кластер на основе записи, иначе запись добавляется во все релевантные кластеры.

Такая кластеризация зависима от порядка обхода записей. Чтобы уменьшить влияние, необходимо последовательно для каждой записи удалить ее из всех кластеров, затем снова проверить на соответствие всем существующим кластерам. Следует добавить запись в каждый релевантный кластер. Такую процедуру можно применять несколько раз, но, как показывает практика, 2-3 раз достаточно.

После описанных процедур может образоваться достаточно много кластеров, векторы которых почти равны векторам других кластеров. От таких дублей следует избавляться. Кластер признается дублем другого кластера, если в нем встречается 90% лексем другого кластера. Оба кластера удаляются из множества, вместо них добавляется кластер, содержащий записи, представленные в обоих кластерах.

В конечном наборе кластеров могут присутствовать пары кластеров, которые схожи относительно функции кластеризации. Такие кластеры удаляются из множества, а вместо них добавляется кластер, содержащий записи обоих кластеров.

4.2.2 Кластеризация записей БНБ

После получения кластеров записей РНБ следует распределить по ним записи БНБ. Этот процесс не зависит от порядка обработки записей БНБ. Кластеризация производится за один проход по

записям БНБ. Каждая запись БНБ сравнивается с кластером РНБ через функцию кластеризации. Если для записи существует хотя бы один релевантный кластер, она записывается в кластер с большей релевантностью.

5 Описание практической части

5.1 Подготовка данных

Была создана онтология, схема онтологии отражена на рисунке 1.

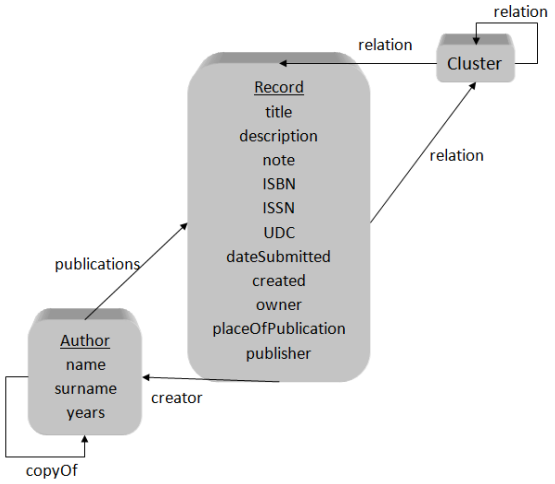

Рис. 1. Сконструированная онтология

Для описания авторов публикаций используется класс Author, который через предикат copyOf может ссылаться на свою копию. Множество объектов, связанных через copyOf, задают полное описание некоторого автора, которого они описывают по отдельности.

Для описания связей используется класс Cluster, если две записи связаны предикатом relation с одним объектом класса Cluster или с разными объектами, но связанными предикатом relation, то они считаются связанным.Для описания записи используется класс Record, имеющий множество предикатов для описания состояний своих объектов.

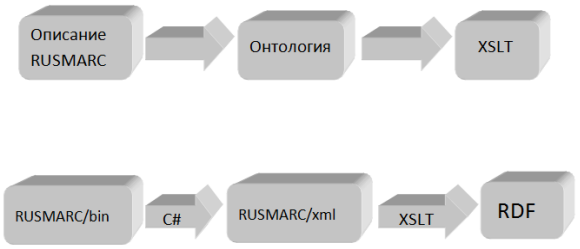

Рис. 2. Процесс получения RDF из данных РНБ

На рисунке 2 приведена схема получения XSLT преобразования из RUSMARC/xml в RDF и получения RDF из RUSMARC/bin.

Полученный RDF был вставлен в хранилище TDB, а поверх него была настроена SPARQL точка доступа fiseki, входящая в состав Jena, с логическим выводом OWL от библиотеки pellet. Логический вывод pellet оказался в десятки раз быстрее стандартного логического вывода OWL, реализованного в Jena, и смог обрабатывать большие объемы данных (300 000 троек). Стандартный вывод зациклился на таком объеме данных.

Был настроен сервер jetty для предоставления информации по HTTP. При получении запроса сервер обращается в хранилище данных РНБ и хранилище связей с БНР с помощью SPARQL запросов, получает нужную информацию и отправляет пользователю.

Согласно алгоритмам кластеризации, описанным выше, заголовки и описания записей РНБ были переведены на английский язык, разбиты на кластеры, и связи с БНБ были получены. Было создано хранилище TDB с SPARQL точкой доступа fiseki, был настроен логический вывод RDFS.

Для алгоритма кластеризации был проведен ряд экспериментов. Были скачаны группы новостей, распознанные системой «Yandex Новости», переведены на английский язык и кластеризованы.

Пусть *Drel* – множество связей распознанных яндексом.

Пусть *Dretr* – множество связей распознанных системой.

Тогда точность определяется формулой (1), а полнота – формулой (2):

$$\frac{|Drel \cap Dretr|}{|Dretr|}, \quad (1)$$

$$\frac{|Drel \cap Dretr|}{|Drel|}. \quad (2)$$

В результате эксперименты была получена точность равная 80% и полнота равная 60%. Эксперимент повторялся несколько раз с разными группами новостей, отклонение составило ±10%.

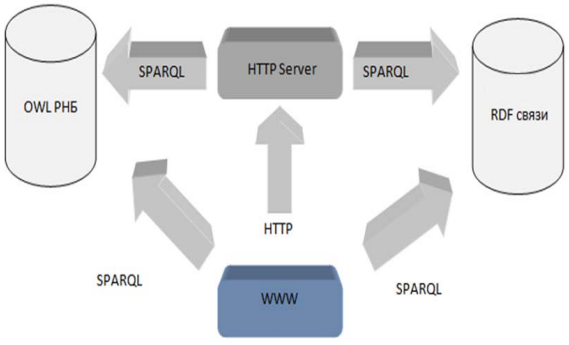

Рис. 3. Схема пользовательского приложения

Общая схема пользовательского приложения, позволяющего просматривать данные о библиографических записях РНБ и их связи между собой, авторами и библиографическими записями БНР, отображена на рисунке 3.

6 Заключение

В данной работе были разработаны программные решения, осуществляющие публикацию библиографических записей в пространство LOD и

интеграцию с библиографическими записями других источников. Описана онтология библиографических записей. Разработана процедура преобразования данных из формата RUSMARC/bin в RDF. Создано семантическое хранилище и SPARQL точка доступа. Настроен HTTP сервер для доступа к семантическим данным. Разработаны и реализованы алгоритмы потоковой кластеризации для получения связей записей РНБ и БНБ. Получены оценки качества алгоритма кластеризации.

Дальнейшие работы могут вестись по направлениям:

– полнотекстовый поиск по заголовкам и описаниям;

– создание распределенного хранилища;

– поиск по классификаторам UDC и BDC;

– поиск по issn и isbn.

Литература

[1] Т. Бернерс-Ли, Д. Хендлер, О. Лассила. Семантическая сеть. http://ezolin.pisem.net/logic/semantic_web_rus.html

[2] Спецификация зыка RDF. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210

[3] Спецификация языка OWL. http://www.w3.org/TR/2012/REC-owl2-syntax-20121211

[4] T. Heath, C. Bizer. Linked Data: Evolving the Web into a Global Data Space. California : Morgan & Claypool, 2011. 136 c.

[5] T. Berners-Lee. Linked Data – Design Issues http://www.w3.org/DesignIssues/LinkedData.html

[6] ГОСТ 7.9–2003. Москва : Изд-во стандартов, 2004. 6 с.

[7] Library Linked Data Incubator Group Final Report http://www.w3.org/2005/Incubator/lld/XGR-lld-20111025

[8] О.Н. Жлобинская. MARC-форматы в современной информационной среде. http://www.rusmarc.ru/publish/MARC_now.pdf

[9] П. Воляк. Проблемы кластеризации новостного потока. http://nlpseminar.ru/lecture50/

Semantic Integration of Bibliographic Records

D. Malakhov, V. Serebriakov,
K. Teymurazov, O. Shorin

The paper deals with the problems of integration of bibliographic records in frame of the task of integration of data from the Russian National Library and the Britain National Library as a part of the Linked Open Data space. In course of solving the problem a prototype system has been constructed, through which the data format RUSMARC may be published in the Linked Open Data and linked to other library data. Experiments were carried out and quality estimates were obtained for for the subsystem linking data from different sources.