Ontology Integration for the Linked Open Data

Author:
Lihua Zhao

Supervisor:
Ryutaro Ichise

Doctor of Philosophy

Department of Informatics
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies

2013

A dissertation submitted to the Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Advisory Committee

Assoc. Prof.	Ryutaro Ichise	National Institute of Informatics, SOKENDAI
Prof.	Hideaki Takeda	National Institute of Informatics, SOKENDAI
Assoc. Prof.	Ikki Ohmukai	National Institute of Informatics, SOKENDAI
Prof.	Akiko Aizawa	National Institute of Informatics
Assoc. Prof.	Kouji Kozaki	Osaka University

Abstract

The Linked Open Data (LOD) cloud contains tremendous amounts of interlinked instances, from where we can retrieve abundant knowledge. In order to access to the linked data, we have to be familiar with the ontology of each data set. However, because of the heterogeneous and big ontologies, it is time consuming to learn all the ontologies manually and it is difficult to observe which properties are important for describing instances of a specific class. In order to construct an ontology that can help users easily access to various data sets, we propose the **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) that can reduce the heterogeneity of the ontologies, retrieve core ontology schemas, and construct easily understandable integrated ontology.

FITON solves three main problems: ontology heterogeneity problem, difficulty in identifying core ontology schemas, and missing domain or range information problem. The three main components of FITON solve each problem, which are graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. The graph-based ontology integration approach solves the ontology heterogeneity problem by analyzing the graph patterns of the interlinked instances and integrates heterogeneous ontologies by retrieving related classes and properties that are critical to link the same instances in different data sets. The machine-learning-based approach retrieves core ontology schemas (top-level classes and frequent core properties) by applying Decision Table and Apriori, that can help Semantic Web application developers easily understand the ontology schemas of the data sets. Furthermore, the integrated ontology constructor automatically adds missing domain, range, and annotations that can provide us rich information about the ontology. The integrated ontology constructed by FITON can help us discover missing links, detect misused properties, recommend standard ontology schemas for the instances, and improve the information retrieval with simple SPARQL queries.

To my family and friends.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Ryutaro Ichise for the continuous support of my Ph.D study and research, for his patience, motivation, and immense knowledge. Without his guidance and persistent help this thesis would not have been possible. I appreciate all his contributions of time, ideas, and supervision to make my Ph.D. experience productive and stimulating.

I would like to thank my committee members, Prof. Hideaki Takeda, Prof. Akiko Aizawa, Prof. Ikki Ohmukai, and Prof. Kouji Kozaki, for their encouraging, insightful, and constructive comments. In addition, thank Prof. Keizo Oyama, Prof. Atsuhiro Takasu, Prof. Nigel Collier, and Prof. Ken Satoh, who provided valuable and thoughtful feedback.

I appreciate my lab members, Lankesh, Mizanur, Raul, and all the internship students, who supported me and helped me in my research. This thesis would not be possible without the support and encouragement of my friends: Osamnia, Raghvendra, Hamza, Mohamed, Yi-Ju, Dan, Simo, Xuping, Yunlong, Zhi, Yanlin, Yu, Lei, Ruijian, Tao, Jeffrey, Vanessa, Khanh, Hua, Lihong, Masakazu, and all my friends from NII and SOKENDAI.

Most importantly, none of this would have been possible without the love and patience of my family. I would like to express my heart-felt gratitude to my parents and sister, who have been a constant source of love, concern, and support all these years.

Lastly, I thank the educational and financial support from National Institute of Informatics and the Graduate University for Advanced Studies (SOKENDAI) during my doctoral studies. I appreciate all the incredible opportunities and memorable experience they offered during my Ph.D. study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the background knowledge about the Linked Open Data, OWL, and SPARQL (Section 1.1). Then we present the motivation of our thesis in Section 1.2. The problem statement and contributions in our thesis are discussed in Section 1.3 and Section 1.4, respectively. In the last Section 1.5, we introduce each chapter of our thesis.

1.1 Background

The Semantic Web is a Web of Data, where the related data are linked so that the machine can explore the web of data by crawling the links. This collection of interrelated data sets on the Web are usually referred to as Linked Data. Linked Open Data (LOD) is Linked Data which is released under an open license, which does not impede its reuse for free [1].

A five-star rating scheme for the linked open data is introduced by Tim Berners-Lee as follows [1, 2]:

1. ★ Data is available on the web with an open license.
2. ★★ Data is available as machine-readable structured data (e.g. excel instead of image scan of a table).
3. ★★★ Data is available as (2), plus non-proprietary format (e.g. CSV instead of excel).
4. ★★★★ All the above, plus use open standards from W3C (RDF and SPARQL) to identify things, so that people can link to it.
5. ★★★★★ All the above, plus link data to other people′s data to provide context.

The Semantic Web research community and particularly the W3C Linking Open Data (LOD) project aimed to bootstrap the Web of Data by identifying existing data sets available under open licenses, convert them to RDF according to the Linked Data principles, and to publish them on the Web. As a point of principle, the project has always been open to anyone who publishes data according to the Linked Data principles.

The Linked Open Data (LOD) cloud is a set of data sets that have been published and interlinked by the project so far. The LOD cloud has been growing rapidly over the past years, that consists of 295 machine-readable data sets with over 31 billion Resource Description Framework (RDF) triples in the form of <subject, predicate, object> (as of Sep. 2011) [3]. The data sets in the LOD cloud are mainly categorized into seven domains: cross-domain, geographic, media, life sciences, government, user-generated content, and publications as shown in FIGURE 1.1.

Tim Berners-Lee, the inventor of the World Wide Web, introduced the Linked Data principles as follows [1]:

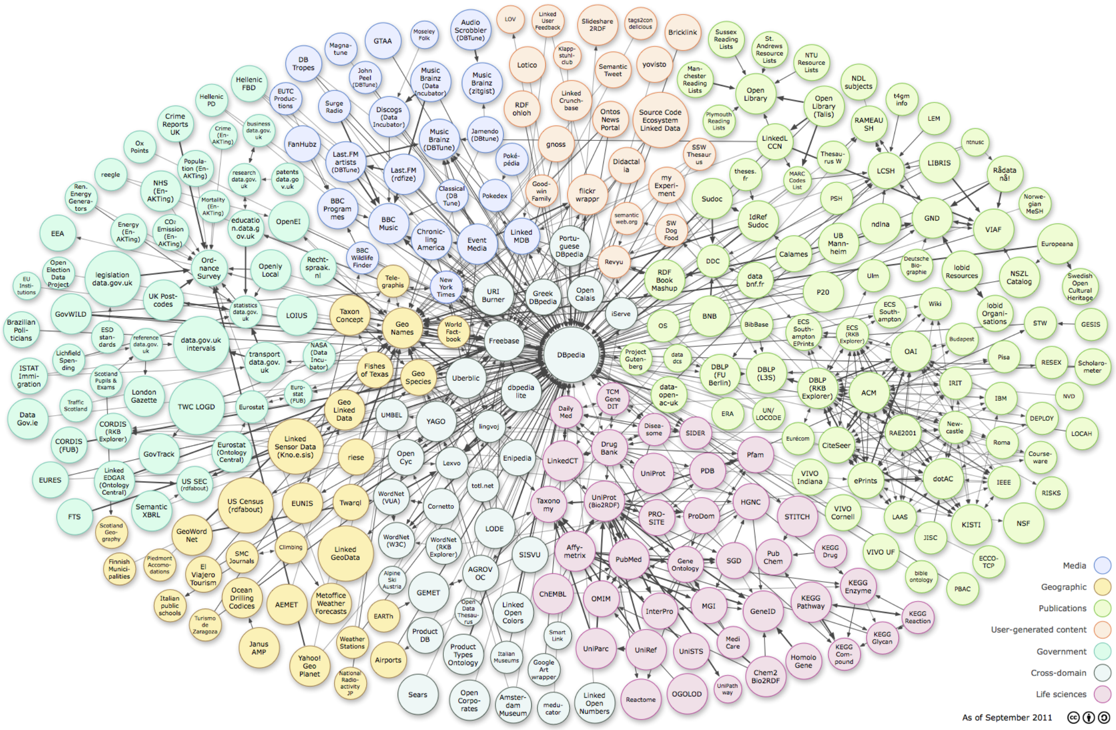

FIGURE 1.1: Linked Open Data Cloud Diagram.

- Use URIs to denote things.
- Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents.
- Provide useful information about the thing when its URI is dereferenced, leveraging standards such as RDF* and SPARQL.
- Include links to other URIs, so that they can discover more things.

The links are essential in the Linked Open Data that interconnect instances from different data sets. The same instances in different data sets are interlinked with *owl:sameAs*, which is a built-in OWL property [1]. Currently, there are around 504 million *owl:sameAs* links in the LOD cloud. The Web Ontology Language (OWL) is a semantic markup language developed as a vocabulary extension of the RDF with more vocabularies for describing classes and properties [4]. RDF is a general-purpose language for representing information on the Web. RDF Schema is a semantic extension of RDF, that provides mechanisms for describing groups of related resources and the relationships between these resources [5]. The OWL 2 Web Ontology Language [6] provides classes and properties as the old OWL 1 [4], but with richer data types, data ranges, and disjoint properties, etc.

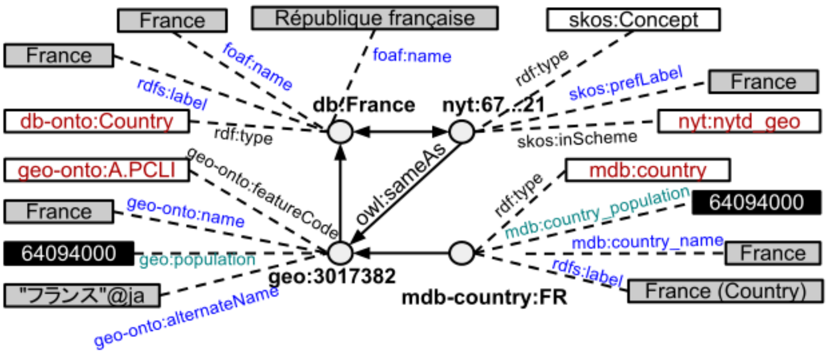

FIGURE 1.2: Interlinked Instances of "France".

SPARQL Protocol and RDF Query Language (SPARQL) is a powerful RDF query language that enables Semantic Web users to access to linked data [7]. However, the users have to understand the ontology schemas of the data sets to construct SPARQL queries [8]. SPARQL can be used to express queries across diverse data sources, where the data is stored natively as RDF or viewed as RDF via middleware [9].

1.2 Motivation

Many Semantic Web applications have been developed by accessing to the data sets in the LOD cloud, such as linked data browsers [10], semantic search engines [11], and some domain specific applications [12]. Although many Semantic Web applications have been developed that demand access to the linked data sets, integrating ontology schemas or data sets from diverse domains remains a challenging problem [2]. Furthermore, not all the ontology schemas are necessary for accessing to different data sets. For instance, when we want to link the ontology from a publication data set to a cross-domain data set, we only need to know the ontology schemas related to the publication in the cross-domain ontologies.

The data providers use different ontology schemas for publishing their data sets in the LOD cloud, which causes the ontology heterogeneity problem. As we can see from FIGURE 1.2, the "France" instances in different data sets are described using different ontology schemas. Furthermore, multiple data sets provide different values for the same predicate of an object [13]. Hence, in order to improve the interoperability of the linked data sets, we have to understand their heterogeneous ontologies in advance and the alignments between different ontologies.

Ontology matching tools such as AROMA [14], S-Match [15], RiMOM [16], Agreement-Maker [17], and BLOOMS [18] have been developed for finding alignments between two different ontologies.

However, all these tools can only use two ontologies as inputs and cannot find mappings for the rudimentary ontologies. Furthermore, since they don't analyze the contents of the data sets or the links between instances in the real data, these ontology matching tools cannot identify the actual ontology schemas used in the real data.

Integrating heterogeneous ontologies can help linked data sets integration and missing links discovery. Furthermore, integrating only essential parts of the ontologies and the alignments among various ontologies can improve the interoperability of the data sets and make it easier for the Semantic Web developers to understand how the instances are interlinked.

1.3 Problem Statement

Our research mainly focuses on integrating ontologies from various data sets that can help Semantic Web application developers easily understand the ontologies and access to the data sets. In this paper, we try to solve the following three problems:

1. **Ontology Heterogeneity Problem**
   The data sets are published according to the Linked Data principles and also provide links to other data resources [2]. However, there are all different kinds of ontologies that cause the ontology heterogeneity problem. In [19], they categorized the ontology heterogeneity problem into four different types: syntactic heterogeneity, terminological heterogeneity, conceptual heterogeneity, and semiotic heterogeneity. We mainly focus on the terminological and conceptual heterogeneity problems.

   - The terminological heterogeneity problem occurs when the same entities in different ontologies are represented differently. For instance, openingDate vs. establishedDate and shortDescription vs. abstract.
   - The conceptual heterogeneity is also called semantic heterogeneity in [20] and logical mismatch in [21], which occurs due to the use of different axioms for defining concepts or due to the use of totally different concepts. For instance, most of the airport instances are described as the type of db-onto:Airport

that is subClass of db-onto:Infrastructure (db-onto:Infrastructure is subClass of db-onto:ArchitecturalStructure), but some airports are described using db-onto:Building that is subClass of db-onto:ArchitecturalStructure.

Fig. 1.2 shows the interlinked instances of "France", where all the properties (labeled on the dotted line) connected with the grey boxes (objects) represent the name of "France" and the properties connected to the black boxes represent the population. In order to access to various data sets simultaneously, we have to understand their heterogeneous ontologies in advance to achieve semantic interoperability.

2. **Difficulty in Identifying Core Ontology Schemas**
The instances of each class are described by partial of the ontology properties. When the ontology is large, it is time consuming to identify important properties that are used for describing instances of a specific class. Retrieving frequently used core classes and properties from various data sets can help the Semantic Web developers easily understand the ontology schemas used for describing instances in each data. The core ontology schemas can assist us to construct SPARQL queries and to discover missing information in the data sets, which is defined as top-level classes and frequently used core properties.

    - Top-level class: If the data set is ontology-based, top-level classes are all the direct subClasses of the owl:Thing. Otherwise, we use top categories as top-level classes. For example, in DBpedia, the top-level classes are db-onto:Agent, db-onto:Place, etc., and in NYTimes are nyt:nytd_geo, nyt:nytd_org, and etc.
    - Frequent core property: The frequently used properties that are used for describing instances in the data sets are considered as frequent core properties. For example, the properties db-onto:kingdom, db-onto:class, and db-onto:family are frequently used for describing the instances that are defined with the class of db-onto:Species.

3. **Missing Domain or Range Information**
The relations between the ontology classes and properties are described with the property rdfs:domain, which indicates that the properties are designed to be used for the instances of a specific class. Furthermore, the range information of the values can help users better understand the data sets. However, in the real data sets, there are many ontologies missing domain or range information. Besides the domain and range information, we should also retrieve the description of each ontology schema to construct an easily understandable integrated ontology.

TABLE 1.1: Problems and corresponding components.

Problem	Component in **FITON**
Ontology Heterogeneity Problem	Graph-Based Ontology Integration
Difficulty in Identifying Core Ontology Schemas	Machine-Learning-Based Approach
Missing Domain or Range Information	Integrated Ontology Constructor

1.4 Contributions

We propose the **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) that consists of three main components solving the above problems. The problems and corresponding components in **FITON** are listed in Table 1.1. In the following, we describe the main contributions of each component.

- **Graph-Based Ontology Integration**
  Ontology integration is defined as the process that generates a single ontology from different existing ontologies [22]. However, there are only few links at a class or property level that makes it difficult to directly retrieve equivalent classes and properties for the ontology integration. Ontology alignment, or ontology matching is commonly used to find correspondences between ontologies to solve the ontology heterogeneity problem [23]. We combined the String-based and Knowledge-based ontology matching methods on the predicates and objects to discover similar concepts.

  Since the same instances are linked by *owl:sameAs*, we can create undirected graphs with the linked instances and analyze the graphs to retrieve related classes and properties. By analyzing the graph patterns, we can observe how the same classes and properties are represented differently in various data sets. In order to reduce the ontology heterogeneity problem, we perform different similarity matching methods on the SameAs graph patterns to integrate various ontologies.

- **Machine-Learning-Based Approach**
  Machine learning methods such as association rule learning and rule-based classification can be applied to discover core properties for describing instances in the data sets. Apriori is a well-known algorithm for learning association rules in a big database [24], while the rule-based learning method - Decision Table can retrieve a subset of properties that leads to high prediction accuracy with cross-validation

[25]. By applying machine learning approaches to the linked data sets, we can retrieve core ontology schemas that are important for describing instances in the data sets.

- **Integrated Ontology Constructor**

  The domain and range information in the ontologies are critical for the users to understand the relations between ontology schemas. However, there are many missing domain and range information in the published LOD cloud. Hence, we propose the integrated ontology constructor that automatically enrich the integrated ontology by adding missing domain, range, and default annotations.

  We randomly select some samples of the instances that contain the retrieved classes and properties and analyze the usage of the domain information and values. From the contents of the sample instances, we automatically retrieve the domain information to link the properties and classes. We also retrieve the default range information of the properties and analyze the values of the properties from sample instances, which are mainly categorized into String and Resource. By analyzing samples of the instances, we can reduce the analysis time and also be able to retrieve the domain and range information. The default annotations are also added to make the integrated ontology easily understandable.

In this thesis, we propose the **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) that consists of three main components: graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. **FITON** solves the problems described in Section 1.3 that analyzes graph patterns of the interlinked instances and integrates heterogeneous ontologies by retrieving related classes and properties that are critical to link the same instances in different data sets. **FITON** also retrieves core ontology schemas (top-level classes and frequent core properties) by applying machine learning methods, that can help Semantic Web application developers easily understand the ontology schemas of the data sets. Furthermore, **FITON** enriches the integrated ontology by adding domain, range, and annotations that can provide us rich information about the ontology. The integrated ontology can help us discover missing links, detect misused properties, recommend standard ontology schemas for the instances, and improve the information retrieval with simple SPARQL queries.

1.5 Outline

This thesis consists of 6 Chapters. In the following, we introduce the remaining chapters:

- **Chapter 2: Related Work**
  We discuss some related work such as ontology matching, analysis of SameAs links, and concept extraction. The string-based and WordNet-based ontology matching methods are introduced and some ontology matching approaches for the LOD cloud are discussed. We also present some research that analyze the SameAs network, find alignments on the SameAs network, and discover SameAs links from the LOD cloud. Some concept extraction methods that can help retrieving key concepts from ontologies are also introduced in this chapter. At last, we discuss the unsolved problems and limitations of the related work.

- **Chapter 3: Integrating Heterogeneous Ontologies**
  We introduce the Mid-Ontology learning approach and graph-based ontology integration approach that solve the ontology heterogeneity problem.

  The Mid-Ontology learning framework finds alignments between different ontologies in the LOD data sets. This approach collects interlinked instances and apply ontology similarity matching methods on the collected triples to find related ontology predicates and integrate them into one ontology. After evaluating the quality of the integrated ontology and the performance with a SPARQL query, we analyze the characteristics of the integrated ontology and it's possible applications.

  The graph-based ontology integration approach overcomes some limitations of the Mid-Ontology learning approach by performing ontology matching methods on the SameAs graph patterns. This approach integrates both related classes and properties, and does not require a hub data set to collect data. Furthermore, different similarity measures are applied to the categorized <Predicate, Object> pairs and also default domain information is added to the integrated ontology for better understanding. The comparison results with the Mid-Ontology learning approach is also discussed to show the advantages of the graph-based ontology integration approach.

- **Chapter 4: Enriching the Integrated Ontology**
  Although the integrated ontology constructed using graph-based ontology integration approach successfully integrates related ontology schemas from various data

sets, the integrated ontology may miss some core ontology schemas that are important for describing instances of the data sets. Furthermore, because of the missing domain or range information in the definitions of the ontology schemas, it is difficult to observe the relations between properties and classes or the ranges of property values. In order to solve these two problems: the difficulty in identifying core ontology schemas and missing domain or range information, we propose two approaches to enrich the integrated ontology. The former problem is solved using machine-leaning-based approach and the latter one is solved by the integrated ontology constructor. Detailed experimental results and discussions are also included in this chapter.

- **Chapter 5: Framework for InTegrating ONtologies - FITON**
  This chapter introduces **FITON** which consists of three main components: graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. By combining these three components, we can solve the three proposed main problems. We compare the ontology alignment results with other ontology matching tools and also evaluate the integrated ontology with manually created ontology alignments. At last, we discuss the performance on different data sets, possible applications on link discovery and the effectiveness in information retrieval with the integrated ontology using some SPARQL examples.

- **Chapter 6: Conclusion**
  In the last chapter, we discuss the advantages of our approach by comparing with related work and also discuss the remaining problems. Then we conclude our research work that integrates heterogeneous ontologies by analyzing graph patterns of the interlinked instances, identifies core ontology schemas by applying machine learning methods, and adds missing domain or range information using the integrated ontology constructor. At last, we propose future work to achieve our vision and propose possible further research by using the integrated ontology constructed with **FITON**.

Chapter 2

Related Work

In this chapter, we introduce some ontology matching methods that are commonly used in solving ontology heterogeneity problem (Section 2.1). The ontology matching methods include string-based ontology similarity measures and WordNet-based (knowledge based) ontology matching methods. Then we discuss some research work about analyzing the SameAs networks that can assist finding ontology alignments from interlinked instances and discover SameAs links between instances (Section 2.2). We also introduce some approaches that extract concepts from data or ontologies using machine learning methods in Section 2.3. At last, we list some unsolved problems (Section 2.4) in the related work that can be solved with our approach and summarize this chapter.

2.1 Ontology Matching

The ontology heterogeneity problem is commonly solved by ontology matching methods, which aim at finding correspondences between semantically related entities of different ontologies. These correspondences may stand for equivalence as well as other relations, such as consequence, subsumption, or disjointness between ontology entities. Ontology entities, in turn, usually denote the named entities of ontologies, such as classes, properties or individuals. Ontology matching has taken a critical place for helping heterogeneous resources to inter-operate [19].

An ontology is an explicit specification of conceptualization [26]. The term of ontology is originated from philosophy, where an ontology is a systematic account of existence. Ontologies are the structural frameworks for organizing information and are used in artificial intelligence, Semantic Web, biomedical informatics, and information architecture as a form of knowledge representation about the world or some part of it. For practical usage of ontologies, they should support large-scale interoperability, to be well-founded and axiomatized to be generally understood [27].

Ontologies are expressed in an ontology language, which usually deal with the following kinds of entities [19]:

- **Classes** are interpreted as a set of individuals in the domain defined by the owl:Class. Classes are also called as concepts that are the main entities of an ontology.
- **Individuals** (objects or instances) are interpreted as particular individual of a domain, which is defined by owl:Thing.
- **Relations** are interpreted as a subset of the product of the domain that are introduced in OWL by the owl:ObjectProperty or owl:DatatypeProperty construct.
- **Data Types** are particular parts of the domain, which specify values as opposed to individuals. Typical data types are String, Integer, Date, Number, and etc.
- **Data Values** are simple values.

Interoperability of various data sets might be affected by the heterogeneity of the data sets, which is already well known within the distributed database systems community: structural heterogeneity (schematic heterogeneity) and semantic heterogeneity (data heterogeneity) [28]. Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the content of an information item and its intended meaning. In order to achieve semantic interoperability in a heterogeneous information system, the meaning of the information that is interchanged has to be understood across the systems [29].

Jérôme Euzenat and Pavel Shvaiko introduced four most obvious types of the ontology heterogeneity [19]:

- **Syntactic heterogeneity**: When two ontologies are not expressed in the same ontology language, it causes the syntactic heterogeneity problem. This kind of mismatch can be tackled at the theoretical level by establishing equivalences between constructs of different languages.
- **Terminological heterogeneity**: This kind of heterogeneity problem occurs when the same entities in different ontologies are represented differently. This can be caused by the use of different natural languages, different technical sub-languages, or the use of synonyms.
- **Conceptual heterogeneity**: It is also called semantic heterogeneity in [20] and logical mismatch in [21]. This conceptual heterogeneity happens due to the use of different axioms for defining concepts or due to the use of totally different concepts.
- **Semiotic heterogeneity**: This problem is concerned with how entities are interpreted by people, which is difficult for computers to detect and solve.

In this section, we will introduce some commonly used ontology similarity matching methods that find the relations between entities expressed in different ontologies, where the relations are commonly equivalence relations. The equivalence relations are discovered by measuring the similarity between the entities of the ontologies. Then, we will introduce some ontology matching tools for the LOD ontologies.

2.1.1 String-based Similarity Matching

- **Substring (prefix and suffix) similarity** measures the ratio of the common subpart between two ontologies.

  Definition 2.1. The substring similarity is a similarity σ: $\mathbb{S} \times \mathbb{S} \rightarrow [0, 1]$ such that $\forall x, y \in \mathbb{S}$, and let t be the longest common substring of x and y:

$$\sigma(x, y) = \frac{2|t|}{|x| + |y|}$$

This equation is used for calculating the longest common prefix or longest common suffix.

- n**-gram similarity** is often used in comparing strings, which computes the number of common n-grams. For example, the 3-grams for the string "paper" are: pap, ape, and per.

  Definition 2.2. Let $ngram(s, n)$ be the set of substring of s of length n. The n-gram similarity is a similarity σ: $\mathbb{S} \times \mathbb{S} \to \mathbb{R}$ such that:

  $$\sigma(s,t) = |ngram(s,n) \cap ngram(t,n)|$$

  The normalized version of this function is defined as follows:

  $$\sigma(s,t) = \frac{|ngram(s,n) \cap ngram(t,n)|}{min(|s|,|t|) - n + 1}$$

  This function is efficient when only some parts of characters are missing and it is possible to add extra characters at the beginning or the end of strings if the string is too short.

- **Levenstein Distance** is the edit distance with all costs equal to 1, which is designed for measuring the similarity between strings that may contain misspelling. In information theory and computer science, the Levenshtein distance is a string metric for measuring the difference between two sequences. The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. It is named after Vladimir Levenshtein, who considered this distance in 1965 [30].

  Definition 2.3. Given a set Op of string operations (op: $\mathbb{S} \to \mathbb{S}$), and a cost function w: $Op \to \mathbb{R}$, such that for any pair of strings there exists a sequence of operations which transforms the first one into the second one (and vice versa), the edit distance is a dissimilarity δ: $\mathbb{S} \times \mathbb{S} \to [0, 1]$ where $\delta(s, t)$, is the cost of the less costly sequence of operations which transforms s to t.

  $$\delta(s,t) = \min_{(op_i)_I; op_n(\ldots op_1(s))=t} (\sum_{i \in I} w_{op_i})$$

- **Jaro-Winkler Distance** Jaro measure is for matching proper names that may contain similar spelling mistakes [31, 32], which is based on the number and proximity of the common characters between two strings.

  Definition 2.4. The Jaro distance $\delta(s,t)$ of two given strings s and t is calculated as follows: (δ: $\mathbb{S} \times \mathbb{S} \rightarrow [0,1]$)

$$\delta(s,t) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \times (\frac{|m|}{|s|} + \frac{|m|}{|t|} + \frac{m-t}{m}) & \text{otherwise} \end{cases}$$

  where:

  - m is the number of matching characters;
  - t is half the number of transpositions.

  The Jaro-Winkler measure is introduced in [33], which has been improved by favouring matches between strings with longer common prefixes.

  Definition 2.5. Jaro-Winkler distance $\delta_{jaro}(s,t)$ is:

$$\delta_{jaro}(s,t) = \delta(s,t) + (\ell_p(1 - \delta(s,t)))$$

  where:

  - $\delta(s,t)$ is the Jaro distance for string s and t.
  - ℓ is the length of common prefix at the start of the string up to a maximum of 4 characters.
  - p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. $p \leq 0.25$, otherwise the distance become larger than 1. The standard p for Winkler's work is 0.1.

  JaroWinkler distance is a measure of similarity between two string, which is mainly used in the area of record linkage (duplicate detection). The higher the JaroWinkler distance for two strings is, the more similar the strings are. The JaroWinkler distance metric is designed and best suited for short strings such as person names. The score is normalized such that 0 equates to no similarity and 1 is an exact match.

2.1.2 WordNet-based Similarity Matching

WordNet::Similarity[1] provides six measures of similarity and three measures of relatedness based on the lexical database WordNet [34]. We will introduce each measure.

Three of the six WordNet-based similarity measures are based on the information content of the least common subsumer (LCS) of concepts. These three similarity measures are Resnik [35], Lin [36], and Jiang and Conrath (JCN) [37].

- **Resnik** semantic similarity measure is based on the concept of information content that defines the generality or specificity of a concept in a certain topic. The information content (IC) of a concept c is defined as follows:

$$IC(c) = -\log\left(\frac{freq(c)}{freq(root)}\right)$$

  where

  - $freq(c)$ is the frequency of the concept c.
  - $freq(root)$ is the frequency of the root concept, which counts the occurrences of all the concepts in its taxonomy.

  The Resnik semantic similarity is defined as follows:

$$sim_{Resnik}(c_1, c_2) = IC(lcs(c_1, c_2))$$

  where

  - IC is the information content of a concept.
  - $lcs(c_1, c_2)$ is the lowest common subsumer of concept c_1 and c_2.

- **Lin** considers the information content of lowest common subsumer (lcs) and the two compared concepts. Lin measure is based on the Resnik measure defined as follows:

$$sim_{Lin}(c_1, c_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

- **Jiang and Conrath (JCN)** measure incorporates both information content of the concepts and the information content of their lowest common subsumer. This

[1] http://wn-similarity.sourceforge.net/

measure is also based on the Resnik measure defined as follows:

$$sim_{JCN}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \times IC(lcs(c_1, c_2))}$$

The other three similarity measures are based on path lengths between a pair of concepts, which are Leacock and Chodorow(LCH) [38], Wu and Palmer(WUP) [39], and PATH.

- **Leacock and Chodorow(LCH)** semantic similarity measure is based on counting the number of links between two input systems [38]. This measure finds the shortest path between two concepts and scales it by the maximum path length found in the *is-a* hierarchy in which they occur. The following definition is for computing the scaled semantic similarity between two concepts c_1 and c_2 in WordNet:

$$sim_{LCH}(c_1, c_2) = -\log \frac{len(c_1, c_2)}{2 \times \max_{c \in WordNet} depth(c)}.$$

  where

  - $len(c_1, c_2)$ is the shortest path in WordNet from synset c_1 to c_2.
  - $depth(c)$ is the length of the path from the global root.

- **Wu and Palmer(WUP)** measure finds the depth of the LCS of the concepts and then scales it by the sum of the depths of the individual concepts. The WUP similarity is defined as follows:

$$sim_{WUP}(c_1, c_2) = \frac{2 \times depth(lcs(c_1, c_2))}{len(c_1, lcs(c_1, c_2)) + len(c_2, lcs(c_1, c_2)) + 2 \times depth(lcs(c_1, c_2))}$$

  where

  - $depth(lcs(c_1, c_2))$ is the global depth in the hierarchy, which is used as a scaling factor.

- **PATH** measure is a baseline measure that is equal to the inverse of the shortest path between two concepts as defined bellow:

$$sim_{PATH}(c_1, c_2) = \frac{1}{len(c_1, c_2)}$$

  where

  - $len(c_1, c_2)$ is the shortest path between two concepts.

The other three methods measure relatedness between concepts, which are Hirst and StOnge(HSO) [40], LESK [41], and VECTOR [42].

- **Hirst and StOnge(HSO)** measure classifies relations in WordNet as having direction, and then establishes the relatedness between two concepts by finding a path that is neither too long nor that changes directions too often. Hirst and St.Onge classified links in WorNet into three different categories: *upward*, *downward*, and *horizontal*. They define the strong relation between two words if one of the following holds:

  - They both belong to the same synset.
  - They belong to synsets connected by a *horizontal* link.
  - One of the words is a compound word that includes the second word. The synsets of two words are connected by an *is-a* relation.

  For strongly related words, the relatedness is $2{\cdot}C$, where C is a constant value used for *medium-strong* relations. In practice, $C = 8$, so that the coefficient is 16. A *medium-strong* relation exists if the two synsets are connected by a valid path in WordNet. The similarity (relatedness) in HSO measure is calculated as the following formula:

  $$sim_{HSO}(c_1, c_2) = C - len(c_1, c_2) - k \times turns(c_1, c_2)$$

  where C and k are constants (in practice, they use $C = 8$ and $k = 1$, and $turns(c_1, c_2)$ is the number of times the path between c_1 and c_2 changes direction.

- **LESK** measure incorporates information from WordNet glosses and finds overlaps between the glosses of two concepts, as well as concepts that are directly linked to the concepts.

- **VECTOR** measure creates a co-occurrence matrix for each word used in the WordNet glosses from a given corpus, and then presents each gloss or concept with a vector that is the average of these co-occurrence vectors.

Five of the above measures were evaluated in [43]. The evaluation results show that the information-content-based measure JCN is superior to the other measures HSO, LCH, Lin, and RES.

2.1.3 Aggregation of Ontology Matching Methods

Many ontology matching methods utilize string-based and WordNet-based ontology similarity matching methods for mapping source and target ontologies. For instance, an ontology matching method based on the Ordered Weighted Average(OWA) for combining similarity measures is introduced [44] and a machine learning approach that aggregates 40 similarity measures is introduced [45].

An aggregation approach of similarity measures called n-Harmony is introduced in [46]. The n-Harmony measure identifies the top-n highest values in each similarity matrix to assign a weight to the corresponding similarity measure for aggregation. This approach is based on the Harmony[47] measure that assigns a higher weight to reliable and important similarity measure and a lower weight to those fail to map similar ontologies. In total, 13 different similarity measures are used for aggregation, including 4 string-based, 1 structure-based and 8 WordNet-based similarity measures. The final aggregated similarity matrix *FinalSimMatrix* is calculated as follows:

$$FinalSimMatrix = \frac{\sum_k (nH_k \times SMatrix_k(O_s, O_t))}{|SMatrix|}$$

where nH_k is n-Harmony weight and SMatrix is the similarity matrix of each similarity measure between source ontology O_s and target ontology O_t. Before combining the similarity matrices, min(L-1, $nH \times L$) lowest values in each row and column of similarity matrix are removed as the data cleaning process. Here L is the minimum number of concepts in two ontologies and nH represents harmony weight of corresponding similarity matrix. Furthermore, only those similarity matrices with a high harmony weight are aggregated for the final similarity matrix. The final decision of whether an ontology pair is matching or not depends on the final similarity matrix and manually tuned threshold.

2.1.4 Ontology Matching for the LOD Ontologies

The data sets in the LOD cloud are interlinked at the instance-level using equivalent statements. However, few of the ontologies behind these data sets are interlinked. There are many research work that map the ontologies of the LOD to improve the interoperability and discover missing links. We will introduce some research that find alignments between ontologies of the LOD cloud.

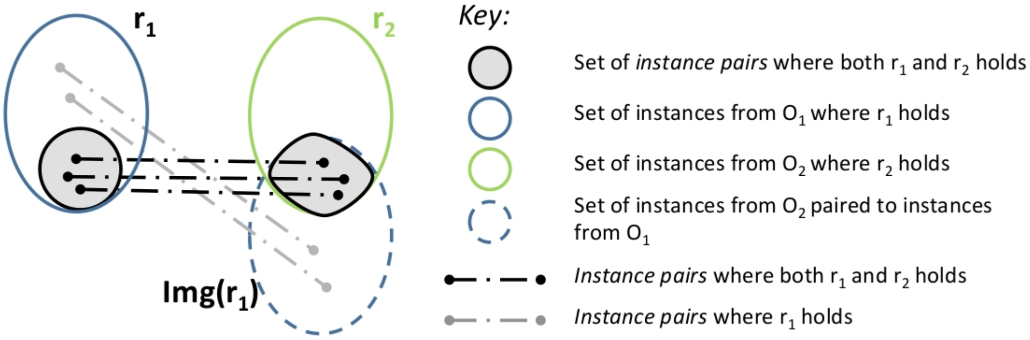

FIGURE 2.1: Measure the Alignment Score between Two Ontologies.

Set Representation	Relation	$P = \frac{\lvert I(r_1) \cap r_2 \rvert}{\lvert r_2 \rvert}$	$R = \frac{\lvert I(r_1) \cap r_2 \rvert}{\lvert r_1 \rvert}$	P′	R′
	Disjoint	= 0	= 0	≤ 0.01	≤ 0.01
	$r_1 \subset r_2$	< 1	= 1	> 0.01	≥ 0.90
	$r_2 \subset r_1$	= 1	< 1	≥ 0.90	> 0.01
	$r_1 = r_2$	= 1	= 1	≥ 0.90	≥ 0.90
	Not enough support	0 < P < 1	0 < R < 1	0.01 < P′ < 0.90	0.01 < R′ < 0.90

FIGURE 2.2: Summary of the Metrics.

- **Aligning Ontologies of the Linked Data**

Some approaches that find ontology alignments from linked instances are introduced in [48–50]. At first, they use the *owl:sameAs* property to link LinkedGeoData with DBpedia, and use *skos:closeMatch* property to link Geospecies with DBpedia. During the data preprocessing, they only considered linked instances and removed unimportant properties to reduce the search space. Their algorithm discovers equivalent and subsumption relations and models one Linked Data source through ontology alignment. They introduced restriction classes, which are the classes defined by the conjunctions of property value restrictions. Their approaches automatically generate alignments from linked instances in the Linked Data, especially geospatial, zoology, and genetics data sources [48].

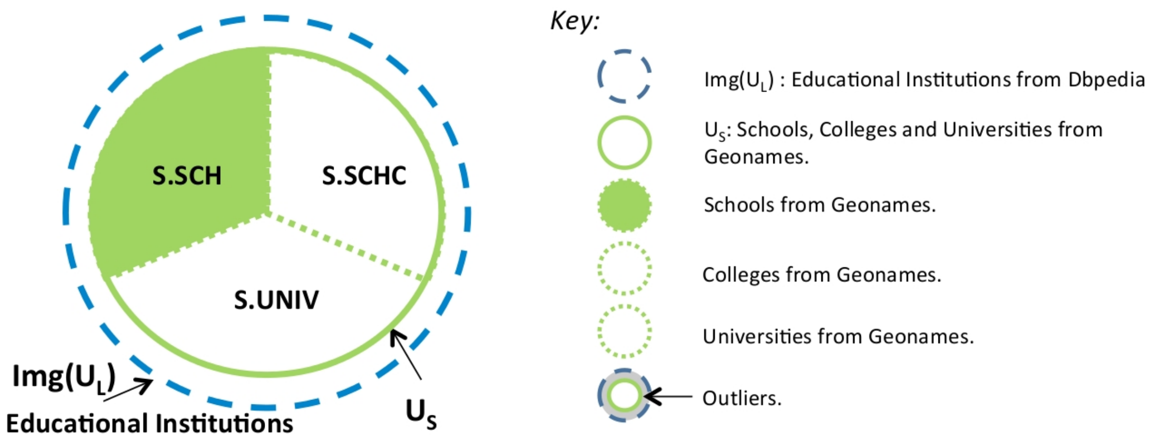

FIGURE 2.3: Spatial Covering of Educational Institutions from DBpedia.

They compare the score using the linked instances from two different ontologies as in FIGURE. 2.1. At first, they find the restriction class r_1 from the first source and r_2 from the second source. Then, compute $I(r_1)$ that denotes the image of r_1: the set of instances from the second source linked to instances in r_1. Next, we compare r_2 with the intersection of $I(r_1)$ and r_2 (shaded region). The two metrics P and R are defined as follows:

$$P = \frac{|I(r_1) \bigcap r_2|}{|I(r_1)|}, \quad R = \frac{|I(r_1) \bigcap r_2|}{|I(r_2)|}$$

In practice, they consider two classes are equivalent if the metric scores are higher than 0.9. The metrics in different cases of intersection are listed in FIGURE 2.2.

The authors improved their method by exploring disjunctions of restriction classes [49, 50]. Their approach produces coverings where concepts at different levels in the ontologies can be mapped even there is no direct equivalence. Furthermore, they map a union of smaller concepts to a larger concept in the other source. For example, as shown in FIGURE 2.3, the union of three concepts $\{$*geonames:featureCode* $\in$ *S.SCH, S.SCHC, S.UNIV*$\}$ is mapped to $\{$*rdf:type* $=$ *dbpedia:EducationalInstitution*$\}$. The outliers occur because of incorrect instance alignments, missing instance alignments, or incorrect values for properties.

- **Debugging Mappings**
  A debugging method for mapping lightweight ontologies is proposed in [51] . They applied a machine learning method to learn a classifier that can determine disjointness of two classes, with the features of the taxonomic overlap, semantic distance, object properties, label similarity, and WordNet similarity. In the following, we describe each feature that are used for determining disjointness of two classes:

Taxonomic Overlap: Two classes are disjoint *iff* their taxonomic overlap is empty. The taxonomic overlap is calculated based on the Jaccard similarity coefficient [52]:

$$f_{overlap_i}(c_1, c_2) = \frac{|i \in I|c_1(i) \wedge c_2(i)|}{|i \in I|c_1(i) \vee c_2(i)|} \quad f_{overlap_c}(c_1, c_2) = \frac{|c \in C|c \subseteq c_1 \bigcap c_2|}{|c \in C|c \subseteq c_1 \bigcup c_2|}$$

Semantic Distance: Minimum length of a path consisting of subsumption relationships between atomic classes that connects two classes c_1 and c_2.

Object Properties: The number of object properties that two classes c_1 and c_2 share.

Label Similarity: Three lexical similarity measures - Levenshtein, n-gram and Jaro-Winkler.

WordNet Similarity: Computes the cosine similarity between vector-based representations of the glosses, that are associated with the two synsets.

- **FactForge**

  In [53], the authors introduced the FactForge system that constructs an intermediate-level ontology to connect general ontologies and an upper ontology. They introduced a method to construct an intermediate-level ontology by mapping an upper ontology PROTON, to the concepts of DBpedia, Geonames, and Freebase described in the FactForge. The mains steps of the matching method are as follows:

  1. Mapping the concepts from PROTON to the concepts described in the datasets of FactForge (DBpedia, Freebase, Geonames).
  2. Assigning subsumption relations between entities and properties from FactForge to PROTON.
  3. Extending PROTON with classes and properties to obtain mapping at a conceptual level with FactForge.
  4. Using OWL class and property construction capabilities to represent classes and properties from FactForge and map them to PROTON classes.
  5. Extending FactForge with instances to account for the conceptual representations of the matching.

  They enriched the upper ontology by adding 166 new classes and 73 new properties; the resulting ontology is large. The Semantic Web users would have to understand this large ontology to construct SPARQL queries. In order to link the concepts and properties in DBpedia and Geonames with PROTON, they manually designed the rules with subsumption relations from FactForge to PROTON. This approach is not scalable when there are many data sets to be linked with the PROTON.

- **BLOOMS+**

  The Semantic Web applications require relationships at both the schema and instance level, but the LOD cloud only contains instance level links. The BLOOMS+ system that can automatically find schema-level links between two LOD ontologies is introduced in [54]. The main steps of BLOOMS+ for aligning two LOD ontologies are as follows:

  1. **Construct BLOOMS+ Forest**

     BLOOMS+ constructs a BLOOMS+ Forest F for each class C from the source and target ontologies. BLOOMS+ uses the preprocessed terms of C to retrieve relevant Wikipedia pages using Wikipedia search web service[2]. It treats each page as a possible sense s_i of C and constructs a category hierarchy tree - BLOOMS+ tree T_i. For s_i we perform the following steps:

     - The root of the tree is s_i.
     - The immediate children of s_i are all Wikipedia categories that s_i belongs to.
     - Each subsequent level includes all unique and direct super categories of the categories at the current level.

  2. **Compute Class Similarity**

     BLOOMS+ compares each class C in the source ontology with each class D in the target ontology to determine their similarity. This is done by comparing $T_i \in F_C$ with each $T_j \in F_D$ where F_C and F_D are the forests for C and D, respectively. The overlap between two BLOOMS+ trees is calculated as follows:

     $$Overlap(T_i, T_j) = \frac{log \sum_{n \in T_i \cap T_j} (1 + e^{d(n)^{-1} - 1})}{log 2|T_i|}$$

     where $n \in T_i \cap T_j$ are the common nodes between the source and target tree; and $d(n)$ is the depth of a common node n in T_i. The value of $Overlap(T_i, T_j)$ is between 0 and 1.

  3. **Compute Contextual Similarity**

     BLOOMS+ calculates the contextual similarity between two sources C and D to further determine whether these two classes should be aligned. The

[2]http://en.wikipedia.org/w/api.php

contextual similarity between C and D with respect to T_i and T_j is calculated using the harmonic mean:

$$CSim(T_i, T_j) = \frac{2R_C \cdot R_D}{R_C + R_D}$$

where R_C and R_D are the fraction of superclasses in $N(C)$ and $N(D)$ supported by T_i and T_j, respectively. Here, $N(C)$ and $N(D)$ are the neighborhoods of C and D respectively.

4. **Compute Overall Similarity**
   The overall similarity between classes C and D is calculated as follows:

$$O(T_i, T_j) = \frac{\alpha \cdot Overlap(T_i, T_j) + \beta \cdot CSim(T_i, T_j)}{2}$$

where α and β are the weights for the concept and contextual similarity respectively. The default values for both α and β are set to 1.0 to give equal importance to each measure.

The tree pairs are selected according to the overall similarity score that are higher than the alignment threshold H_A. BLOOMS+ establishes a link between C and D according to the following regulations:

– If $O(T_i, T_j) = O(T_j, T_i)$, then set C owl:equivalentClass D.

– If $O(T_i, T_j) < O(T_j, T_i)$, then set C rdfs:subClassOf D.

– Otherwise, set D rdfs:subClassOf C.

- **YAM++**

The YAM++ ontology matching system combines several ontology matching components and analyzes how the interactions between the matching components impact the overall quality of the produced ontology alignments [55]. Yam++ contains four main components: a terminological, a structure-based and a semantics-based matcher accompanied by a mapping selection module.

Terminological Matcher: It discovers mappings by comparing annotations of entities. The terminological matching component is divided into *local* and *global* methods.

  - **Local Terminology-Based Methods**: More than 50 local methods are used for the terminology-based matching methods. The representative methods are:
    - **Edit-distance-based methods**: The similarity of two strings is calculated according to the number of edit operations to transform one string to another. They applied Levenstein and ISUB [56].
    - **Token-based methods**: These methods tokenize strings and compare the tokens using string-based methods such as QGrams and TokLev (using Levenstein to compare tokens).
    - **Hybrid methods**:Compare tokens by combining string-based and linguistic-based methods. They used HybLinISUB and HybJCLev, where HybLinISUB uses a combination of ISUB and Lin; HybJCLev relies on the Levenshtein and the JCL method.
  - **Global-Terminology-Based Methods**: Combine local methods by taking into account the semantic context that two entities belong to. The implemented global methods are:
    - **Weighted Average with Local Confidence (LC)**: A local confidence value is assigned to each local method to compute the final similarity score between entities [57].
    - **Harmony-based Adaptive Similarity Aggregation (HADAPT)**: Each local method is assigned a weight for computing the final similarity score, where the weight is calculated using the harmony estimation algorithm [47, 58].
    - **Machine-Learning-Based Approach (ML)**: This method combines all local methods and constructs a classification function on the basis of given training data. Among 15 applied machine learning techniques, J48 decision tree performed best for the ontology matching task [59].
    - **Information-Retrieval-Based Approach (IR)**: This method calculates the similarity between two entities using the amount of overlaps of the information content of their labels. The IR extends Tversky's similarity measure [60] with weight of tokens to compute a similarity score between labels of entities.

Structure-based Matcher: Analyze the similarity of the structural patterns of entities to discover mappings.

Semantic Matcher: It exploits the semantic constraints between entities in the ontologies to discover conflicts between potential mappings and removes them from candidate mappings.

Mapping Selection: It filters out the best mapping candidates, at each of the different mapping level.

2.2 Analysis of the SameAs Links

The equivalent instances in the LOD cloud are interlinked by *owl:sameAs*, which form a very large directed graph by connecting interlinked instances [61]. Therefore, we can find ontology alignments by analyzing the graph formed with interlinked equivalent instances. Ontology alignments are the mapping results of the ontology matching methods, that are commonly used for solving ontology heterogeneity problems. Many statistical and empirical studies have been conducted on the SameAs links [62–65].

2.2.1 Interlinked Instances

In [61], the authors analyzed the SameAs networks that consist of interlinked instances. The SameAs networks are neither large nor complex like the foaf:knows network. They analyzed the Pay-Level-Domain network to examine how data are connected by comparing the five most frequent types. However, when only frequent types are considered, it is not possible to determine exactly how data are connected. They also analyzed the Class-Level Similarity (CLS) network that consists of "class-overlap" relation. Two classes are considered overlapping when they share common instances. The authors claimed that we can automatically detect clusters of classes and ontology mappings with the CLS network using machine learning techniques.

Another empirical study of the owl:SameAs in the Linked Data is introduced in [62]. The *owl:sameAs* property is frequently used to support Linked Data integration via declaratively interconnecting "equivalent" resources across distributed datasets. They designed the following three metrics to measure the *owl:sameAs* network:

- **Size of network** is the number of unique URIs in the *owl:sameAs* network.
- **Dereferenceable portion of network** is measured by the number of dereferencable URIs in the *owl:sameAs* network.

- **Useful portion of network** is measured by the number of URIs in the *owl:sameAs* network, which has been described by more than one triple.

Four fundamental challenges are introduced in [63, 64] to sustainably establish the Web of Data:

- Improving the performance of large-scale RDF data management.
- Increasing and easing the interlinking and fusion of information.
- Improving the structure, semantic richness and quality of linked database.
- Adaptive user interfaces and interaction paradigms.

These four challenges can be tackled by mutual fertilization between approaches include:

- The detection of mapping on the schema level will directly affect instance level matching.
- Ontology schema mismatches between knowledge bases can be compensated by learning which concepts of one are equivalent to which concepts of the other knowledge base.
- Feedback and input from users can be taken as training data for machine learning techniques.
- Semantically enriched knowledge bases improve the detection of inconsistencies and modeling problems, which in turn results in benefits for interlinking, fusion and classification.
- The querying performance of the RDF data management directly affects all other components and the nature of queries issued by the components affects the RDF data management.

The task of increasing and easing the interlinking and fusion of information can draw from previous work such as deduplication [66], entity identification [67], record linkage [68], and etc. Most of the encountered problems are generally caused by data heterogeneity. With the availability of large open data sets and links between them, the generation of benchmarks for interlinking becomes feasible, but still insufficiently covered by research. Both unsupervised and supervised machine learning methods should be investigated for integrating schema mapping and data interlinking algorithms into a mutual refinement cycle that can help improving mapping and interlinking.

Some scalable machine learning techniques that solve the following challenges and problems are introduced in [64]:

- How to find all links between two knowledge bases (high recall)?
- How to verify the correctness of found links (high precision)?
- How to maintain such a link structure with evolving knowledge bases?

2.2.2 Alignments on the SameAs Networks

The data sets in the LOD cloud are interlinked using *owl:sameAs* at the instance level, where the URIs representing entities are reused or aligned towards external URIs. Statistical techniques for ontology alignment on the links of the LOD cloud is a promising approach to resolve the semantic heterogeneity. In [65], the authors assume that *owl:sameAs* equivalence bundles [69] can be treated as singleton instances whose interpretation is provided by following *owl:sameAs* semantics. Therefore, they consider all the interlinked equivalent instances from different data sets as a unique instance.

$$type(?x, ?xt) \wedge sameas(?x, ?y) \wedge type(?y, ?yt) \rightarrow type(?b, ?xt) \wedge type(?b, ?yt)$$

where *type* is rdf:type and *sameas* is *owl:sameAs*.

The equivalence bundles are treated as instances and the degree of overlapping between concepts are computed by processing the typing statements in [65]. They used the Jaccard similarity coefficient [70] to measure the similarity between two concepts when interpreted as sets of instances. The Jaccard coefficient is defined as follows:

$$J(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} = \frac{|A \bigcap B|}{|A| + |B| - |A \bigcup B|}$$

where $|A \bigcap B|$ is calculated by counting the cardinality of the set $< x, y >$: $type(x, A) \wedge sameas(x, y) \wedge type(y, B)$.

They used DBpedia as the source and used OpenCyc, NYTimes, DrugBank, Diseasome, Factbook, and DailyMed as target sources. At first, they analyzed the *owl:sameAs* bundles, which show the typical signs of a power-based network. A small number of *owl:sameAs* bundles contained many entities and the vast majority contained no more than five instances. The relaxation of the acceptance criteria on the alignment from equivalence only to equivalence or subsumption did not influence the overall performance of the measures while giving better performances of the respective more restrictive measures.

2.2.3 Link Discovery

Link discovery frameworks discover missing links between knowledge bases that should be linked via a typed link such as *owl:sameAs* [71]. The Silk - Link Discovery Framework consists of three main components: a link discovery engine, an evaluation tool, and a protocol for maintaining RDF links between continuously changing data sources [72]. The main features of the link discovery engine of the Silk framework are [72, 73]:

- It supports the generation of *owl:sameAs* links as well as other types of RDF links.
- It provides a flexible, declarative language for specifying link conditions.
- It can be employed in distributed environments without having to replicate datasets locally.
- It can be used in situations where terms from different vocabularies are mixed and where no consistent RDFs or OWL schemata exist.
- It implements various caching, indexing and entity preselection methods to increase performance and to reduce network load.

The Silk framework provides some built-in similarity metrics such as weighted average, maximum value, minimum value, euclidian distance metric, and weighted product. Furthermore, jaroSimilarity, jaroWinklerSimilarity, qGramSimilarity, stringEquality, numSimilarity, dateSimilarity, uriEquality, taxonomicSimilarity, maxSimilarityInSet, and setSimilarity are provided as user-defined similarity metrics. Since the property values are often represented differently across datasets, they use transformation functions before passing them to a similarity metric. The transformation functions include removing whitespace from string, converting a string to lower/upper case, and stemming, etc.

FIGURE 2.4 shows the overview of the architecture of the Linked Data applications that operates on top of the Web of Linked Data [74]. The Semantic Web applications that utilize the Web of Linked Data might implement the module depicted in the Data Access, Integration and Storage Layer. We will describe the functionality of each module.

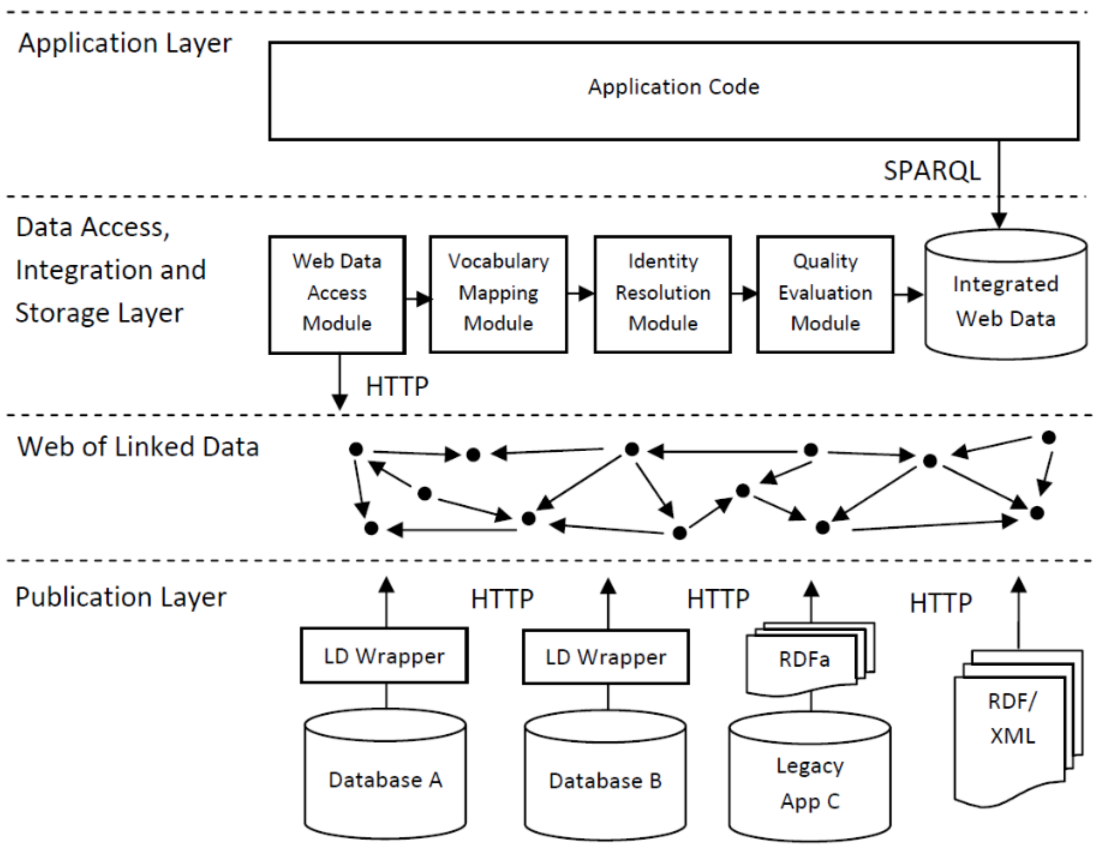

FIGURE 2.4: Schematic Architecture of Linked Data Applications.

- **Web Data Access Module**
  Access the Linked Data to dereference HTTP URIs into RDF descriptions and discover additional data by traversing RDF links.

- **Vocabulary Mapping Module**
  Translate terms from different vocabularies into the application's target schema.

- **Identity Resolution Module**
  Use the *owl:sameAs* statements of the original data and newly discovered linked entities that can be retrieved using the functionality of the Silk Server.

- **Quality Evaluation Module**
  The Linked Data applications should employ a data quality evaluation module that may filter RDF SPAM.

- **Integrated Web Data**
  The cleaned Web of Data is stored in a repository together with provenance information to be used by the application layer. The Named Graphs is commonly used for representing the Web of Data together with provenance information [75].

2.3 Concept Extraction

The Semantic Web is growing rapidly and contains more than seven thousand ontologies [76]. Therefore, it is difficult to identify core ontology schemas in the data sets, which help users easily understand the key concepts of the ontologies. The authors introduced a method to extract n key concepts from a given ontology, where the key concepts can best describe the ontology. Although the Semantic Web Search engines such as Watson and Swoogle [11] help users to find and locate semantic information on the web, they don't support users to quickly understand what an ontology is about and what kind of information it contains [77]. There are several research about retrieving key concepts from ontologies instead of depicting a whole ontology for users.

In [77], the authors aimed to develop an easy and quick way to obtain a general impression of what a particular ontology is about. The authors built vector from an ontology corpus and used a vector description to interpret a keyword. Since term frequency has been proven useful for information retrieval tasks [78], they also applied TF-IDF (Term Frequency - Inverse Term Frequency) approach for weighing terms. However, they used vectors for calculating *IDF* instead of the corpus. The *IDF* is calculated as follows:

$$IDF = log_{10}\frac{N}{n}$$

where the *IDF* factor varies inversely with the number of vectors n to which a term is assigned in a collection of N vectors. By constructing vector description of a domain, their approach achieved high accuracy in terms of ontological knowledge belonging to the domain.

The paper [79] introduced a supervised machine learning approach that uses features defined over pairs of Friend Of A Friend (FOAF) [80] individuals to produce a classifier for identifying co-reference FOAF instances. The FOAF ontology is one of the most widely used Semantic Web ontologies which defines classes and properties for describing people, their attributes, and their relationships to other people, organization, and objects. Although OWL's inverse-functional property can help provide a way to recognize co-reference profiles, it cannot provide complete solutions. The FOAF vocabulary defines, for instance, foaf:homepage and foaf:mbox to be inverse functions, providing strong evidence that two FOAF agents are co-reference if they share an identical value for either of those properties.

The approach for solving the co-reference problem is introduced in [79], which consists of three main steps:

1. **Generating Candidate Pairs**
   Potential FOAF pairs are generated based on a simple heuristic method. They calculate a simple score based on string matches using the common properties among pairs of FOAF instances.

2. **Co-reference Classification**
   Since the Support Vector Machine (SVM) has high accuracy in it's performance [81], they apply SVM for classifying the candidate pairs to decide if they are co-referent or not. The classifier may return a boolean answer or a score that can be used as certainty measure.

3. **Clustering**
   They find connected components in the graph using clustering, each of which represents a set of FOAF profiles that represent the same person.

The supervised machine learning method generates a classifier that can be used for identifying co-reference FOAF instances for linking. They used basic FOAF properties as features for the classifier and used inverse functions and string similarity heuristics for linking the same FOAF instances.

The predicates of ontologies are used in different knowledge bases for different purposes because of the heterogeneity of LOD. Hence, the authors in [82] introduced a data-driven approach to dissolve the existing synonym dependencies for predicate expansion. They proposed an association rule based approach that discovers overlaps between attribute values in an RDF corpus. The FP-Growth [83] association rule mining algorithm is applied in their method. Any part of subject-predicate-object (SPO) statement can be regarded as a context, which is used for grouping one of the two remaining parts of the statement as the target for mining. They defined six possible configurations and their preliminarily identified use-cases such as schema discovery, basket analysis, clustering, range discovery, topical clustering, and schema matching.

Their approach aims at discovering all possible predicate pairs, where each predicate could be the expansion of the other one. Three basic strategies are combined to discover these candidate pairs.

- **Schema Analysis**: Three scoring functions such as confidence aggregations, reversed correlation coefficient, and syn-function are used to express negative correlation.
- **Range Content Filtering**: Since synonym predicates have a similar meaning, they also share a similar range of object values. Therefore, they apply an efficient range content filtering (RCF) approach based on mining configurations.
- **Range Structure Filtering**: They analyze type distributions in predicate ranges, where they only consider the most specific type of an entity. Having type vectors for a predicate pair, the range type similarity can be calculated using measures, such as cosine similarity or weighted Jaccard similarity.

Their approach addressed data inconsistencies due to synonymously used predicates in RDF and introduced concept of predicate expansion for SPARQL patterns. Their algorithm performs well on data containing only subjects of one domain, but produces more false positives on RDF data where the subjects represent entities of many different types.

2.4 Unresolved Issues

We discussed many related work in previous sections that are helpful for our approach to construct an integrated ontology from the LOD cloud. Here, we discuss some limitations in the related research work that we can solve with our approach.

- **Limitation on Domains**
  Although many research work such as [48] and [50] have been successfully matched ontologies to reduce the ontology heterogeneity problem, their approaches are limited to specific domains of the data sets. Furthermore, in [48], they only focused on ontology alignment at the class level and did not consider at the property level, which is useful to find related instances from different data sets.
- **Limitation on the Lightweight Ontologies**
  Some researchers applied a debugging method with machine learning methods for mapping lightweight ontologies that performs better than other state of the art ontology matching systems [51]. However, the method is only limited to the expressive lightweight ontologies that cannot be applied to real data sets, which lack of expressive disjointness axioms and contain large ontologies.

- **Limitation on the Analysis of the SameAs Networks**
  In [84], the authors extracted features from the entities of the graph patterns to detect hidden *owl:sameAs* links or relations in geographic data sets. They applied a supervised learning method on the frequent graph patterns to find useful attributes that link instances. However, their approach did not discuss about what kind of features are important for finding the hidden links. Also some authors analyzed the basic properties of the SameAs networks, the Pay-Level-Domain network, and the Class-Level similarity network [61]. They analyzed the Pay-Level-Domain network to examine how data are connected by comparing the five most frequent types. However, they only considered frequent types, which is not possible to determine exactly how data are connected.

In contrast to the research described above, our method is domain-independent and successfully integrates heterogeneous ontologies by extracting related properties and classes that are critical for interlinking instances. Our approach can find alignments for big or rudimentary ontologies, which most of the ontology matching tools failed to deal with. The discovered alignments with our approach are both at the class and property level. Furthermore, we also discover frequently used core properties and top-level classes in each data set that can help data publishers detect misuses of ontologies in the published data sets. In addition, for the instances of specific classes, we can recommend core properties that are frequently used for the instance description.

2.5 Summary

The ontology heterogeneity problem is one of the most challenging problems that affects the interoperability of the Linked Open Data. In order to achieve semantic interoperability for various data sets from different domains, many ontology matching tools have been developed to solve the problem. We introduced commonly used ontology matching methods that are applied in our research and also other related work in the ontology matching field.

Since the links play an important role in the LOD cloud, we can collect adequate information by crawling the *owl:sameAs* links. We introduced the fundamental knowledge about the linked instances and the SameAs networks in the LOD. Then we described some approaches that find alignments on the SameAs networks and discover missing SameAs links.

We introduced some research that work on retrieving key ontology concepts using machine learning methods and some works that quickly extract main concepts from an ontology to help users understand the ontology. At last, we listed some unsolved problems in the related work, which are solved in our research work.

Chapter 3

Integrating Heterogeneous Ontologies

In this chapter, we present two approaches that can solve the ontology heterogeneity problem in the LOD cloud by integrating the heterogeneous ontologies. Ontology integration is defined as a process that generates a single ontology from different existing ontologies. The first approach is Mid-Ontology learning approach, which includes ontology manipulations such as ontology term extraction, ontology matching, and ontology integration. However, there are some limitations in the Mid-Ontology learning approach. In order to overcome the limitations of the Mid-Ontology learning approach, we propose the graph-based ontology integration method. The graph-based ontology integration consists of 5 main steps: graph pattern extraction, <Predicate, Object> collection, related classes and properties grouping, aggregation for all graph patterns, and manual revision.

At first, we discuss the motivation of integrating heterogeneous ontologies in Section 3.1. In Section 3.2, we introduce the detailed Mid-Ontology learning approach and evaluate the approach from five different perspectives: the effectiveness of data reduction, the quality of Mid-Ontology, the effectiveness of information retrieval with a SPARQL query, and analysis of the integrated ontology schemas, and discussion of possible applications with Mid-Ontology. In Section 3.3, we will introduce the detailed graph-based ontology integration framework and then discuss experimental evaluations. At last, we summarize and discuss about the advantages of the graph-based ontology integration approach.

3.1 Motivation

The ontology heterogeneity problem in the LOD cloud induces the difficulty of accessing to various data sets and remains as one of the most challenging problems in the Semantic Web research. This problem persists because multiple data sets provide different values for the same predicate of an object or provide different terms to represent the same predicate [13].

SPARQL Protocol and RDF Query Language (SPARQL) is a powerful RDF query language that enables Semantic Web users to access to the Linked Data [7]. However, the users have to understand the ontology schemas of the data sets in order to construct SPARQL queries [8]. Learning all the ontology schemas is not feasible and is time-consuming, because each data set has a specially designed ontology and thus thousands of distinct ontology predicates might exist. For example, the postal code is represented using *db-onto:postalCode*[1] in DBpedia, while it is represented using *geo-onto:postalCode*[2] in Geonames. If these two predicates are integrated together into one ontology predicate, for instance, in the *mo-onto:postalCode*, we can easily search places with the same postal code in both DBpedia and Geonames by querying with the *mo-onto:postalCode*. Querying with one simple ontology that integrates various ontologies can simplify SPARQL queries and help Semantic Web application developers easily understand the ontology schemas so that they can retrieve rich information from various linked data sets.

Another problem of dealing with the LOD cloud is that not all the data sets in the LOD are trustworthy. For instance, the data publishers sometimes make mistakes when they convert data into RDF triples. They may use different terms of the properties for the same concept. For example, the "birthday" is represented using "birthDate", "birthdate", "dateOfBirth", and etc. Furthermore, some of the instances are described with general ontology classes rather than specific classes. We should correct these mistaken data, but it is time-consuming and infeasible to manually inspect large ontologies of the linked data sets to discover these mistakes.

To solve the problems mentioned above, an automatic method that constructs a simple ontology by integrating ontology schemas from diverse domain data sets must be developed. Ontology alignment, or ontology matching is commonly used to find correspondences between ontologies to solve the ontology heterogeneity problem [23]. Ontology learning technology can automate the ontology construction process from structured,

[1]db-onto: http://dbpedia.org/ontology/.
[2]geo-onto: http://www.geonames.org/ontology#.

TABLE 3.1: Collected data based on the db:Hamburg.

Predicate	Object
$db - prop : name$	"Free and Hanseatic City of Hamburg"@en
$db - prop : population$	1769117
$db - onto : populationTotal$	1769117
$rdf : type$	db-onto:City
$rdf : type$	db-onto:Place
......	......
$geo - onto : officialName$	"Hamburg"@de
$geo - onto : officialName$	"Free and Hanseatic City of Hamburg"@en
$geo - onto : population$	1739117
......	......
$skos : prefLabel$	"Hamburg (Germany)"@en
$nyt - prop : first\_use$	2006-12-12
$nyt - prop : latest\_use$	2010-05-14

semi-structured or unstructured data [85]. An ontology learning cycle that includes ontology design, ontology learning, and validation phases is introduced in [86]. However, the majority of the research on the ontology learning technology focus on text files [87, 88]. In order to adapt to the LOD data sets, we designed an automatic Mid-Ontology learning approach, which can construct an integrated ontology from various linked data sets [89–91].

3.2 Mid-Ontology Learning Approach

3.2.1 Objective

In the LOD cloud, data sets are linked with *owl:sameAs* at an instance level, but few links exist at the class or property level. Although the RDF link types owl:equivalentClass and owl:equivalentProperty are designed to indicate that two classes or properties refer to the same concept, there are only few links at the class level or property level [7]. Hence, whenever linked data sets are queried with SPARQL, the predicates of the ontology schemas must be manually learned, an infeasible task if there are thousands of distinct predicates.

Our aim is to automatically construct a simple ontology that integrates ontology schemas from various linked data sets. By collecting linked instances, we can identify different predicates that indicate identical or related information. For example, TABLE 3.1 shows

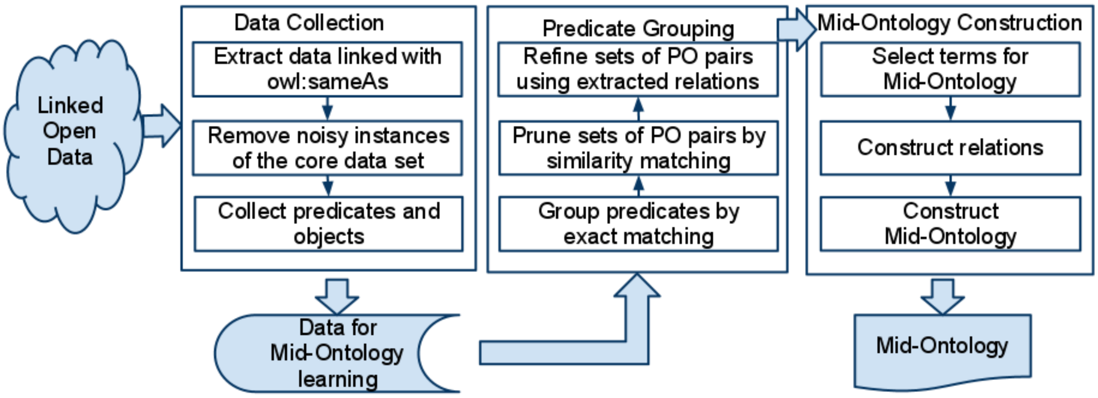

FIGURE 3.1: Architecture of the Mid-Ontology Learning Approach.

the collected <Predicate, Object> pairs of the instances that indicate the place "Hamburg" from DBpedia [92], Geonames, and NYTimes, where these instances are linked with *owl:sameAs*. In TABLE 3.1, there are three distinct predicates that indicate "Hamburg," each belongs to a different data set. If we can integrate these related predicates into one predicate, we can query the entire data sets with one single predicate that indicates the name of a place.

3.2.2 Detailed Approach

In the following, we will describe the architecture of our Mid-Ontology learning approach, as shown in FIGURE 3.1. The architecture of the Mid-Ontology learning approach comprises three phases: data collection, predicate grouping, and Mid-Ontology construction.

1. **Data Collection**

   Although the SameAs (*owl:sameAs*) link is designed to link identical things, it also links related or similar things in the published linked data sets [93]. Hence, we can find identical or related information if we investigate on the instances linked with *owl:sameAs*. The data collection phase consists of three steps: extract data linked with *owl:sameAs*, remove noisy instances of the core data set, and collect predicates and objects to construct the final data set for our Mid-Ontology learning.

   (a) **Extract Data Linked with *owl:sameAs***

   To extract data linked with *owl:sameAs*, we have to select a core data set, which serves as a hub of several linked data sets. A good core data set should have inward or outward links to other data sets from diverse domains. After the core data set selection, we also collect all the instances that have the SameAs links with instances in the core data set.

For instance, if we select DBpedia as the core data set, then we select all the DBpedia instances that have SameAs links with other data sets, such as Geonames and NYTimes. TABLE 3.1 shows an example of the collected data based on the DBpedia instance db:Hamburg[3]. We also collect the contents of the instances *http://data.nytimes.com/N78...471* and *http://sws.geonames.org/2911297/* that are connected with the DBpedia instance db:Hamburg.

(b) **Remove Noisy Instances of the Core Data Set**
We say an instance is *noisy*, if none of the triples of the instance contains information that can represent the characteristics of the instance. For instance, if all the triples of an instance are SameAs links or broken links, we cannot learn any information that can represent the characteristics of an instance. We remove these noisy instances of the core data set before collecting predicates and objects.

(c) **Collect Predicates and Objects**
Data collection is based on each retrieved instance of the core data set, from which we collect all the <Predicate, Object> pairs of the instance and of the instances connected with the instance. Hereafter, we use *PO* to represent a <Predicate, Object> pair.

In this *PO* collection step, we do not collect the triples with the SameAs link, because we have already collected triples of the linked instances. Furthermore, for the instances of the core data set, if there is any link to another instance of the core data set, we also collect triples from the linked instance. In addition, if an instance has a redirection to another instance, we also collect triples from the redirected instance, which normally contains richer information than the original instance. The collected data consists of *PO* pairs based on each instance of the core data set, which is used for the predicate grouping phase.

2. **Predicate Grouping**
Grouping related predicates from different ontology schemas is critical to the Mid-Ontology construction, because there exist many similar ontology predicates that represent the same thing. The predicate grouping phase consists of three main steps: group predicates by exact matching, prune sets of *PO* pairs by similarity

[3]db: http://dbpedia.org/resource/

matching, and refine sets of PO pairs using the extracted relations such as domain and range of the predicates.

(a) **Group Predicates by Exact Matching**

The first step of the predicate grouping is to create the initial sets of PO pairs that share identical information or the same object by the exact string matching method. The collected data set is based on each instance of the selected core data, which consists of PO pairs. We perform a pairwise comparison of PO_i and PO_j and create the initial sets $S_1, S_2, \ldots, S_k$ by checking whether they share an identical predicate or object. Here, S is a set of PO pairs.

For example, in TABLE 3.1, both geo-onto:officialName and db-prop:name[4] have the same object, "Free and Hanseatic City of Hamburg"@en, and the predicate geo-onto:officialName has another object, "Hamburg"@de. Hence, these two predicates and objects are grouped together to create an initial set. After creating initial sets for all the PO pairs, we create initial sets for each PO pair that has not yet been grouped. For instance, nyt-prop:first_use[5] is in an initial set by itself because no predicate has the same object "2006-12-12" and no identical predicate exists in the data.

(b) **Prune Sets of *PO* pairs by Similarity Matching**

The second step of the predicate grouping is pruning the initial sets by string-based similarity matching and knowledge-based similarity matching, which are commonly used to match ontologies at the concept level [19]. We observed that some of the same values that are written in different languages and some semantically identical words, such as U.K. and United Kingdom, may be ignored in the exact matching step. Therefore, similarity matching is required to group semantically similar predicates.

In our approach, we adopted four string-based similarity measures, namely, prefix, suffix, Levenshtein distance, and n-gram, as introduced in [45, 94], and nine knowledge-based similarity measures [34], namely, LCH, RES, HSO, JCN, LESK, PATH,WUP, LIN, and VECTOR, which are based on WordNet (a large lexical database of English [95]). String-based similarity measures are applied to compare the objects of the predicates, because objects may contain URIs instead of lexical labels or phrases. However, knowledge-based similarity measures are applied to compare pre-processed terms of the predicates because the terms of the predicates are more likely to have semantic meanings that can

[4]db-prop: http://dbpedia.org/property/.

[5]nyt-prop: http://data.nytimes.com/elements/.

TABLE 3.2: WordNet-based similarity on pairwise terms.

Pairwise Terms	LCH	RES	HSO	JCN	LESK	PATH	WUP	LIN	VECTOR
population, population	1	1	1	1	1	1	1	1	1
population, total	0.4	0	0	0.06	0.03	0.11	0.33	0	0.06

be recognized as a concept. In the following, the term T_S indicates the pre-processed terms of the predicates in S, the term O_S indicates the objects stored in S, and the term P_S indicates the predicates stored in S.

To extract the terms of predicates, we pre-process each predicate of the PO pairs by performing natural language processing (NLP), which includes tokenizing terms, removing stop words, and stemming terms using the porter stemming algorithm [96]. NLP is a key method for the data pre-processing phase, in which terms are extracted from ontologies; this method helps improve the performance of ontology building [97].

$Sim(S_i, S_j)$ is the similarity between S_i and S_j, which is calculated using the formula:

$$Sim(S_i, S_j) = \frac{StrSim(O_{S_i}, O_{S_j}) + WNSim(T_{S_i}, T_{S_j})}{2}$$

where $StrSim(O_{S_i}, O_{S_j})$ is the average of the four string-based similarity values and $WNSim(T_{S_i}, T_{S_j})$ is the average of the nine applied WordNet-based similarity values. For WordNet-based similarity measures, we do not consider the term pairs that have no similarity value returned from WordNet-based similarity measures. Because if we consider them, it means that we treat the similarity value as zero. However, no returned similarity value from WordNet-based similarity measures does not mean the similarity is zero.

If $Sim(S_i, S_j)$ is higher than a predefined similarity threshold, we consider that these two initial sets share similar predicates, and we merge these two sets. After comparing all the pairwise initial sets, we remove the initial set S_i if it has not been merged during this pruning process and has only one PO pair.

Here, we show how to calculate the similarity between two initial sets S_i and S_j, where these two initial sets are created based on TABLE 3.1. Suppose S_i includes db-prop:population and db-onto:populationTotal with the object "1769117", and set S_j includes geo-onto:population with the object "1739117". T_{S_i} includes "population" and "total", while T_{S_j} includes "population". Here, O_{S_i} is "1769117" and O_{S_j} is "1739117".

We performed four string-based similarity measures, including prefix, suffix, Levenshtein, and n-gram, on the two objects, "1769117" and "1739117". The similarity values for prefix, suffix, Levenshtein, and n-gram are 0.29, 0.56, 0.4, and 0.86, respectively. Hence, the string-based similarity $StrSim(O_{S_i}, O_{S_j})$ is 0.5275, which is the average of the 4 string-based similarity values. TABLE 3.2 shows the WordNet-based similarity values of the pairwise terms in S_i and S_j. $WNSim(T_{S_i}, T_{S_j})$ is 0.5825, which is the average of the 15 similarity values returned by WordNet-similarity measures, as listed in TABLE 3.2. Hence, the final similarity $Sim(S_i, S_j)$ is 0.555, which is the average of 0.5275 and 0.5825. If this value is higher than the predefined similarity threshold, we merge S_i and S_j. In this work, we set the default similarity threshold to 0.5.

The pruned sets of *PO* pairs are created by performing pairwise similarity measures on the initial sets and are passed to the refining process.

(c) **Refine Sets of *PO* pairs using Extracted Relations**

The final step of the predicate grouping is to split the predicates of each pruned S_i according to the relations of rdfs:domain and rdfs:range [5]. Even though the objects or terms of predicates are similar, the predicates may belong to different domains or ranges. For further refinement, we determine the frequency of each pruned S_i in all of the data and keep any S_i that appears with a frequency that is higher than the predefined frequency threshold. This refining process is applicable to any type of objects because we only consider the domain and range information. The final refined sets of *PO* pairs are passed to the next phase, Mid-Ontology construction.

3. **Mid-Ontology Construction**

According to the refined sets of *PO* pairs, we construct the Mid-Ontology with automatically selected terms and a specially designed predicate.

(a) **Select Terms for Mid-Ontology**

To perform automatic term selection, we pre-process all the terms of the predicates in each set by tokenization, stop words removal, and stemming. During the pre-process, non-literal terms and stop words are removed because they can not convey meanings. We also keep the original terms because sometimes a single word is ambiguous when it is used to represent a set of terms. For example, "area" and "areaCode" have different meanings but may have the same frequency because the former is extracted from the latter. Hence, when two terms have the same frequency, we choose the longer one. The predicate mo-onto:Term is designed to represent a class term, where the "Term" is automatically selected.

(b) **Construct Relations**

We designed a predicate mo-prop:hasMembers to link sets of predicates with the Mid-Ontology classes. This predicate indicates that a set of integrated predicates are members of a Mid-Ontology class. We use the relation mo-prop:hasMembers instead of the existing predicates owl:equivalentClass or owl:equivalentProperty in order to reduce the number of triples to connect the integrated predicates. For example, if the number of integrated predicates in a set S_i is n, we need $n*(n-1)$ triples to connect all the pairs of predicates using owl:equivalentClass or owl:equivalentProperty. However, by connecting with our Mid-Ontology classes, we only need n triples with mo-prop:hasMembers.

(c) **Construct Mid-Ontology**

A Mid-Ontology is automatically constructed with the refined sets of integrated predicates, automatically selected terms of predicates, and the designed predicate mo-prop:hasMembers, which links sets of predicates and Mid-Ontology classes.

Implementation

Many Semantic Web tools have been developed to help researchers query linked data, publish linked data, or manage enormous data sets. Virtuoso[6] is a high-performance server that supports the storage of a large RDF data, provides a SPARQL endpoint, and supports the creation of RDF models [98]. Therefore, we use Virtuoso to store linked data sets and query SPARQL examples for experiments. A Virtuoso Jena RDF Data Provider is also provided, enabling Java applications to directly query the Virtuoso RDF data through Jena RDF Frameworks.

For knowledge-based similarity matching, we used WordNet::Similarity[7] [34], which is implemented in Perl. Several WordNet-based similarity measuring algorithms are implemented in this tool. If two terms are identical, we return 1, which is the maximum similarity value; otherwise, we apply WordNet-based similarity measures. The similarity measures JCN, PATH, WUP, LIN, and VECTOR return normalized values between zero and one. However, the similarity values of LCH, RES, HSO, and LESK are not normalized. To normalize the similarity values of LCH, RES, HSO, and LESK measures,

[6]http://virtuoso.openlinksw.com/

[7]http://wn-similarity.sourceforge.net/

we divide the returned values by the maximum value that can be obtained from all the pairwise terms in the collected data set. The normalized similarity Sim_{alg} is calculated using the formula:

$$Sim_{alg} = \frac{WordNet_{alg}}{Max_{alg}}$$

where $WordNet_{alg}$ indicates the returned value from WordNet::Similarity tool and Max_{alg} indicates the maximum value we obtained from the $WordNet_{alg}$. The Max_{alg} of LCH, RES, HSO, and LESK are 3.7, 10, 16, and 5.6, respectively.

3.2.3 Experimental Evaluation

In this section, we first introduce the experimental data used in our experiments. Then we evaluate the Mid-Ontology learning approach from five different perspectives: the effectiveness of data reduction, the quality of constructed Mid-Ontology with different combinations of data sets, the effectiveness of information retrieval with a SPARQL example, and the analysis of the characteristics of integrated predicates in the Mid-Ontology. This experimental evaluation is an extension of the research work [89, 99] with more linked data sets to show the adaptability of our method on different combinations of data sets.

1. **Experimental Data**

   We used the following four data sets in the LOD cloud to evaluate our approach.

   DBpedia is a core cross-domain data set that describes over 3.5 million things including persons, places, music albums, movies, video games, organizations, species, and diseases. DBpedia has more than 232 million RDF triples and more than 8.9 million distinct URIs.

   Geonames is a data set that is categorized in the geographic domain and contains more than 7 million unique URIs that represent geographical information on places across the world.

   NYTimes data is a small data set that consists of 10,467 subject headings, where 4,978 are about people, 1,489 are about organizations, 1,910 are about locations, and 498 are about descriptors.

   LinkedMDB is the Linked Movie DataBase, which contains high-quality interlinks to movie-related data in the LOD cloud as well as links to movie-related web

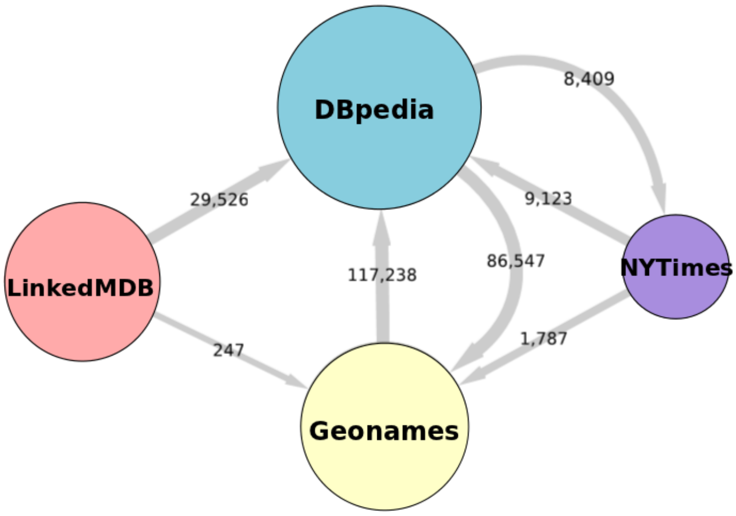

FIGURE 3.2: SameAs Links between Data Sets.

pages. LinkedMDB consists of approximately 6 million RDF triples and more than 0.5 million entities.

FIGURE 3.2 shows the SameAs links connecting the above four data sets, plotted using Cytoscape [100]. In this figure, the size of a node is determined by the total number of distinct instances in a data set on a logarithmic scale. The thickness of an arc is determined by the number of SameAs links on a logarithmic scale. The number of SameAs links are labeled on each arc, which links different data sets.

In addition to the SameAs links, there are links at class and property level, which are linked with owl:equivalentClass and owl:equivalentProperty. However, in our experimental data sets, there is no owl:equivalentClass link among them. Although we found 10 pairs of properties linked with the property owl:equivalentProperty in DBpedia, most of the properties are removed during the data collection phase because of their infrequent usage in the data sets. Hence, these few links between classes and properties do not affect the evaluation results with our approach.

2. **Evaluation of Data Reduction**

   We evaluate the effectiveness of data reduction during the data collection phase by comparing the number of distinct instances in the original data sets with the number of distinct instances we extracted after performing the *owl:sameAs* retrieval process and the noisy instances removal process.

   DBpedia has served as a hub within the Web of Data because of its breath of topical coverage and the wealth of inward and outward links connecting the instances in DBpedia to the instances in other data sets [7]. Hence, we select DBpedia as a core

Table 3.3: Number of instances in each data set.

Data	Number of Instances	*owl:sameAs* Retrieval	Noisy Instances Removal
DBpedia	8,955,728	163,916	115,364
Geonames	7,479,714	128,482	82,055
NYTimes	10,467	10,408	9,242
LinkedMDB	503,242	29,526	28,946

data set and collect all the instances that have the SameAs links as do instances in DBpedia.

The instances of DBpedia that contain only the following predicates db-prop:wordnet-type[8], *owl:sameAs*, and db-prop:hasPhotoCollection are defined as "*noisy*" and are removed from the collected linked instances. We reduce the data in such a way because although these instances have SameAs links, from them we cannot learn any information that can represent the characteristics of a DBpedia instance. For example, most of the corresponding objects of the property db-prop:hasPhotoCollection link to a broken link from which we cannot extract information about the link.

The instances of four data sets kept after the data reduction process are shown in TABLE 3.3. These retained instances are used for our Mid-Ontology learning approach to discover related classes and properties. For example, the db:Hamburg in TABLE 3.1 is kept after the data reduction process because it has SameAs links to instances of other data sets and it is not a "noisy" instance. Therefore, we crawl the SameAs links shown in FIGURE 3.2 and collect <Predicate, Object> pairs from the interlinked instances as listed in TABLE 3.1.

TABLE 3.3 illustrates the number of distinct instances that exist before and after linked data retrieval and noisy instances removal are performed during the data collection process. The data sets contain 8,955,728 DBpedia instances, 7,479,714 Geonames instances, 10,467 NYTimes instances, and 503,242 LinkedMDB instances. After the linked data retrieval process, we extracted 163,916 DBpedia instances, 128,482 Geonames instances, 10,408 NYTimes instances, and 29,526 LinkedMDB instances, which are 1.83%, 1.72%, 99.44%, and 5.87% of the number of instances in the data sets, respectively.

Then, we pre-process the extracted sub-data set by removing noisy DBpedia instances. After the noisy instances removal, we obtained 115,364 DBpedia instances,

[8]db-prop: http://dbpedia.org/property/

TABLE 3.4: Comparison results of constructed Mid-Ontologies.

Data set	Mid-Ontology	Number of Classes	Number of Predicates	Accuracy	Correctly Labeled	Correctly Grouped	Predicate Richness
DBpedia	MO_1_no_prune_refine	9	228	78.34%	6 (66.67%)	6 (66.67%)	25.33
Geonames	MO_1_no_prune	14	203	82.53%	10 (71.43%)	9 (64.29%)	14.5
NYTimes	MO_1_no_refine	21	166	93.92%	16 (76.19%)	19 (90.48%)	7.90
	MO_1	23	148	94.18%	19 (82.61%)	19 (82.61%)	6.43
DBpedia	MO_2_no_prune_refine	10	240	77.94%	7 (70%)	6 (60%)	24
Geonames	MO_2_no_prune	15	238	78.23%	11 (73.33%)	8 (53.33%)	15.87
LinkedMDB	MO_2_no_refine	19	157	93.16%	15 (78.95%)	17 (89.47%)	8.26
	MO_2	24	150	96.92%	21 (87.5%)	22 (91.67%)	6.25
DBpedia	MO_3_no_prune_refine	9	99	84.76%	7 (77.78%)	6 (66.67%)	11
LinkedMDB	MO_3_no_prune	15	82	85.76%	13 (86.67%)	10 (66.67%)	5.47
NYTimes	MO_3_no_refine	10	120	95.39%	9 (90%)	9 (90%)	12
	MO_3	14	106	93.77%	13 (92.86%)	12 (85.71%)	7.57
DBpedia	MO_4_no_prune_refine	9	169	71.58%	5 (55.56%)	5 (55.56%)	18.78
Geonames	MO_4_no_prune	14	155	79.99%	9 (64.29%)	9 (64.29%)	11.07
NYTimes	MO_4_no_refine	18	111	94.63%	15 (83.33%)	16 (88.89%)	6.17
LinkedMDB	MO_4	22	105	94.84%	19 (86.36%)	20 (90.91%)	4.77

82,055 Geonames instances, 9,242 NYTimes instances, and 28,946 LinkedMDB instances, which are 70.78%, 63.86%, 88.80%, and 98.04% of the number of instances in the extracted linked data, respectively.

We dramatically scaled down the data sets by collecting information of linked instances in the data collection phase so as to keep instances that share related information. Furthermore, we successfully removed noisy instances, which may affect the quality of the constructed ontology.

3. **Evaluation of the Mid-Ontology**

To evaluate the quality of the constructed Mid-Ontology (MO), we calculate the accuracy of the Mid-Ontology using the following formula:

$$ACC(MO) = \frac{\sum_{i=1}^{n} \frac{|Correct\ Predicates\ in\ C_i|}{|C_i|}}{n}$$

where n is the number of classes in the Mid-Ontology, and $|C_i|$ indicates the number of predicates in class C_i. The *ACC(MO)* is the average of the accuracy of each class in the Mid-Ontology. If all related or identical predicates are correctly integrated in each class, *ACC(MO)* reaches 1.

TABLE 3.4 shows the improvements achieved by our Mid-Ontology approach through comparisons of the Mid-Ontologies constructed with and without our approach using different combinations of the data sets, as illustrated in the first column. The main features of our approach are the *PO* set pruning with similarity measures and the *PO* set refining by checking the ranges and domains of predicates. The second column lists the Mid-Ontologies constructed by different approaches, that is, MO_i_no_prune_refine is constructed without the pruning and refining processes, MO_i_no_prune is constructed without the pruning process, MO_i_no_refine is constructed without the refining process, and MO_i is constructed with both the pruning and refining processes. The third column lists the number of classes, and the fourth column lists the number of predicates in the constructed Mid-Ontology. The following columns list the accuracy, the percentage of correctly labeled terms, the percentage of correctly grouped classes, and the Predicate Richness (PR) of the constructed Mid-Ontology.

(a) **Evaluation of Pruning Process**

To evaluate the accuracies of the constructed Mid-Ontologies, we manually check predicates in each class to determine whether they share identical or related information, and we examine whether each term can represent the predicates in that class without disambiguation. The performance of the pruning process can be evaluated by comparing the results of MO_i_no_prune_refine and MO_i_no_refine, and the results of MO_i_no_prune and MO_i.

For example, in the first case, which is performed on DBpedia, Geonames, NYTimes, the pruning process significantly improved the accuracy of the Mid-Ontology, that is, from 78.34% to 93.92% and from 82.53% to 94.18%, with a p-value 0.01. Here, the p-value is calculated using a t-test to measure the statistical significance of improvement. The p-value calculated on eight pairwise accuracies from four different cases is 2.0E-5, indicating that our pruning process significantly improves the accuracy of the Mid-Ontology.

The pruning process is applicable to any type of objects, but it is not an optimized method. Since we consider the numerical values of objects as literals in the similarity matching, it may cause incorrect similarity values. This problem can be solved by applying different similarity measurements for different types of objects as introduced in [101].

(b) **Evaluation of Refining Process**

We can compare the results of MO_i_no_prune_refine and MO_i_no_prune, and the results of MO_i_no_refine and MO_i to evaluate the performance of the refining process. For example, in the first case, the refining process improved the accuracy of the Mid-Ontology, that is, from 78.34% to 82.53% and from 93.92% to 94.18%. Although the improvements of the refining process from four different cases are not significant, with a p-value 0.056, this value is close to the significant level 0.05.

The reason of insignificant improvements on the refining process is that most of the predicates in the ontologies have no definition of range and domain. For instance, there is no domain and range information in the LinkedMDB and NYTimes data sets. In Geonames, there are 12 range information and 21 domain information. However, all the ontology predicates are defined as geo:Feature. DBpedia has over 6000 predicates for describing instances, but less than 20% of them have domain and range information. We will seek for a solution to improve the performance of refining process on the data sets, which are lack of domain and range information.

The decrease of the accuracy in this experiment is caused in the third case, where the accuracy of MO_3 is slightly lower than MO_3_no_refine. One reason for this decrease is that in the MO_3, db-onto:producer and db-prop:playername are divided from the largest class mid-onto:name and grouped into a new class mo-onto:playername after the refinement step because both predicates indicate the name of a person. However, the db-onto:producer is related to movies and db-prop:palyername is related to sports. Hence, we define this set as incorrect, and the accuracy of mo-onto:playername is 50%, decreasing the accuracy of MO_3 by approximately 2.27%. Another reason for the accuracy decrease is that there are not as many links as there are in other cases; in other words, fewer DBpedia instances are retrieved during the data collection. The number of DBpedia instances in the third case is only 30% of the number of instances in the last case. Hence, because of the limited number of extracted instances, the refining process failed to improve the accuracy.

(c) **Evaluation of Correctly Labeled and Grouped Sets**

As we can see from TABLE 3.4, the percentages of correctly labeled sets and correctly grouped classes of MO_i are improved compared to the results obtained when the group pruning or group refining processes were not performed.

Although the percentage of correctly labeled terms for MO_i is higher than in the case of methods that do not involve group pruning or group refining, such as MO_i_no_prune_refine, MO_i_no_prune, and MO_i_no_refine, the label accuracy is not affected by the pruning and refining processes. Some of the automatically labeled terms are ambiguous to represent the precise meaning of the terms in a set. Hence, we define these unclear terms as incorrectly labeled terms. An example of an incorrectly labeled term in the fourth case in TABLE 3.4 is "date", which should represent the "release date" of a movie. The term "date" is selected because it appeared most frequently in the preprocessed terms of the set. However, this term is too ambiguous that we can not figure out whether it is a release date of a movie.

The percentage of correct groups is significantly improved after performing pruning process, with a p-value 6.27E-6. However, the improvement of the refining process is not significant, with a p-value 0.3. Although the accuracies of correct groups are high with the experimental data sets, there are some incorrect groups. An example of an incorrect group is the one including predicates about longitude and latitude. This wrong group is caused by the coincidence when the value of longitude and latitude are similar. Furthermore, due to the lack of domain and range information of the predicates that indicating longitude and latitude, they are not separated during the refining process.

Table 3.5: Predicates grouped in mo-onto:date.

```
<rdf:Description rdf:about="mo-onto:date">
    <mo-prop:hasMembers rdf:resource="db-prop:released"/>
    <mo-prop:hasMembers rdf:resource="db-onto:releaseDate"/>
    <mo-prop:hasMembers rdf:resource="mdb-movie:initial_release_date"/>
    <mo-prop:hasMembers rdf:resource="dc:date"/>
</rdf:Description>
```

(d) **Evaluation of Predicate Richness**

As TABLE 3.4 shows, when both pruning and refining are conducted, the total number of predicates from the data sets are decreased and the total number of classes are increased. To determine whether we successfully retrieved related predicates and removed redundant predicates, we introduce the term Predicate Richness (PR), which is calculated using the following formula:

$$PR = \frac{|Number\ of\ Predicates|}{|Number\ of\ Classes|}$$

where PR indicates the average number of predicates in a class of the ontology. As we can see from TABLE 3.4, the cases with a low PR have a high accuracy. Hence, according to the high accuracy and the low PR, we can conclude that through the pruning and refining process we successfully removed redundant predicates, which may reduce the accuracy of the Mid-Ontology.

4. **Evaluation with a SPARQL Example**

We evaluate the effectiveness of information retrieval with the Mid-Ontology constructed via our approach using DBpedia, Geonames, NYTimes, and LinkedMDB data sets by presenting a SPARQL query example.

TABLE 3.5 shows one of the classes in the Mid-Ontology, that integrates predicates indicating the release date of a movie from DBpedia and LinkedMDB. The predicates db-prop:released and db-onto:releaseDate are used in DBpedia, while mdb-movie:initial_release_date and dc:date[9] are used in LinkedMDB. This set does not contain any NYTimes and Geonames predicates because there is no predicate that indicates a movie's release date in both data sets. TABLE 3.6 shows a SPARQL example in which this mo-onto:date is used to find movies that are released on 2000-10-03. This SPARQL query automatically queries with all the predicates listed under mo-onto:date, as shown in TABLE 3.5.

[9]dc: http://purl.org/dc/terms/

Table 3.6: A SPARQL example: Find movies released on 2000-10-03.

```
SELECT DISTINCT ?movies
        WHERE{  <mo-onto:date>  mo-prop:hasMembers  ?prop.
        ?movies  ?prop  ?date.
         FILTER  REGEX(?date, "2000-10-03").}
```

Table 3.7: SPARQL example results with each single predicate.

Single property for the release date of a movie	Number of Results
db-prop:released	37
db-onto:releaseDate	43
mdb-movie:initial_release_date	2
dc:date	2

Table 3.8: Sample classes in the Mid-Ontology.

Data sets	Mid-Ontology class
DBpedia	mo-onto:producer mo-onto:areacode
DBpedia & Geonames	mo-onto:population mo-onto:postal
DBpedia & LinkedMDB	mo-onto:date
DBpedia & Geonames & NYTimes & LinkedMDB	mo-onto:name

We identify 46 movies with mo-onto:date, where 44 are from DBpedia and 2 are from LinkedMDB. However, with the single predicate listed in TABLE 3.7, we can find 37, 43, 2, and 2 movies. Because the predicates grouped in this class all correctly represent the release date of a movie, the returned results are all correct. The result queried with mo-onto:date is a combination of the results retrieved with each predicate in that set. Furthermore, it is difficult to manually find all four predicates that indicate the release date in different data sets.

As this example shows, our approach simplifies SPARQL queries and returns all the possible results without user interaction; conversely, it is time-consuming to find each single predicate manually through user interaction.

5. **Analysis of the Integrated Ontology Schemas**
We evaluate whether our approach successfully integrates related predicates by illustrating examples of classes in the Mid-Ontology. TABLE 3.8 shows some of the classes in the Mid-Ontology, which integrated predicates from the DBpedia, Geonames, NYTimes, and LinkedMDB instances.

The classes listed in the first row include only predicates from DBpedia ontology, which indicate the producer of a movie and the area code of a place. The second row lists classes that integrate predicates from both DBpedia and Geonames, which indicate the population and postal code of a place. The third row includes the release date of a movie, which integrates predicates from DBpedia and LinkedMDB. The last row includes a predicate that integrates predicates from DBpedia, Geonames, NYTimes, and LinkedMDB, which indicates the name of persons, places, news, or movies.

From the characteristics of the integrated classes, we can observe that the linked instances between DBpedia and Geonames are about places. The instances that link DBpedia and LinkedMDB are based on the release date of a movie, and all the predicates that refer to the label of a thing in four data sets are integrated in mo-onto:name. The predicate mo-onto:name can be the name of a place, a movie, an actor or actress, a news title, etc.

Although, the Mid-Ontology successfully integrated some related predicates, there are still some missing predicates that should be integrated together. For example, the predicate mdb-movie:runtime[10] and db-prop:runtime both indicate the runtime of a movie, but these two predicates do not appear in the final Mid-Ontology. This failure of integration occurs because we filtered out infrequent sets of PO pairs that have a frequency lower than the predefined threshold: 0.1% of the total number of retrieved core data instances. Hence, when the threshold is high, some sets such as "runtime" and "prominence" are filtered out.

Another interesting observation is that some predicate terms that are written in different languages are integrated together. For instance, db-prop:einwohner is integrated in the mo-onto:population, which means "the population" in German. The other integrated predicates such as geo-onto:population and db-onto:populationTotal have clear terms in English. Another example is db-prop:vorwahl which is also in German meaning the area code of a place. Two predicates db-prop:areaCode and db-onto:areaCode are integrated together with db-prop:vorwahl in the mo-onto:areaCode. The terms written in English are easy to understand, but it is impossible to identify the term "einwohner" or "vorwahl" if we do not know German.

6. **Discussion**

Experimental results demonstrate that our Mid-Ontology learning approach successfully integrates predicates from different data sets. The automatically constructed Mid-Ontology has a high quality and can be applied in the information

[10]mdb-movie: http://data.linkedmdb.org/resource/movie/.

retrieval field. Because the Mid-Ontology integrates the most related predicates, we can search related triples or instances from the LOD cloud with a simple SPARQL query.

Furthermore, our Mid-Ontology learning approach is implemented with the collected data set that is extracted with *owl:sameAs*. Hence, with our Mid-Ontology in a SPARQL query, it is possible to find missing links that should be linked with *owl:sameAs*. For example, the predicate mo-onto:date has predicates in DBpedia and LinkedMDB that indicate the release date of a movie. Therefore, we can find the same movie in DBpedia and LinkedMDB by searching movies that are released on the same date with the same title. From the results of the SPAQL query in TABLE 3.6, we found one missing link that should connect two instances db:Scooby-Doo_and_the_Alien_Invaders and mdb-film:45394[11]. The DBpedia instance db:Scooby-Doo_and_the_Alien_Invaders has predicates db-prop:released and db-onto:releaseDate with the value "2000-10-03", and mdb-film:45394 has predicates mdb-movie:initial_release_date and dc:date with the same value, "2000-10-03". This LinkedMDB instance also indicates the movie "Scooby-Doo and the Alien Invaders", but there is no *owl:sameAs* link between these two instances.

Therefore, we can find missing links with our Mid-Ontology if there exist predicates from different domains grouped under the same Mid-Ontology class. In our constructed Mid-Ontology, we can find missing links according to the predicates mo-onto:birthdate, mo-onto:population , mo-onto:postalcode, etc. In some SPARQL queries, we can find many missing links that should be connected with "owl:sameAs". For example, we can use mo-onto:birthdate to find out persons who were born on the same day. With mo-onto:birthdate, we found 18 instances of persons who were born on "1961-08-04", where 4 of them indicate "Barack Obama", but no SameAs links among these instances. Hence, we can add SameAs links among the instances.

Although, it is difficult to find out all the missing links in the linked data, we can discover missing links with a specific SPARQL query template.

[11]mdb-film: http://data.linkedmdb.org/resource/film/

3.3 Graph-Based Ontology Integration

3.3.1 Objective

The data sets of the LOD use different ontologies to describe instances, which causes the ontology heterogeneity problem. Dealing with the heterogeneous ontologies is a challenging problem and it is time-consuming to manually learn big ontologies in the LOD. In previous Section 3.2, we introduced the Mid-Ontology learning approach that can reduce ontology heterogeneity by integrating related predicates of the ontologies. However, the Mid-Ontology learning approach has some drawbacks. In the following, we list the drawbacks in each process of the Mid-Ontology approach.

- **Data Collection:** A hub data is necessary to collect data, which is difficult to decide among various data sets.
- **Predicate Grouping:** All the types of property values are treated as String when calculating the similarities of predicates. Furthermore, we only considered predicates and did not consider grouping related classes.
- **Mid-Ontology Construction:** The domain and range information is not included in the Mid-Ontology, that makes it difficult to be understood the relations between predicates and classes, and the ranges of predicates.

In order to overcome the above drawbacks, we propose the graph-based ontology integration approach, which is an extension of the Mid-Ontology learning approach. We retrieve graph patterns from several linked data sets without a hub data set, and collect data for further processing. During the data collection, we classify <Predicate, Object> pairs into five different types and perform different methods to identify related ontology classes and properties in each graph pattern. We also add domain information for the groups of related predicates by analyzing existing definitions of the integrated predicates. Graph-based ontology integration approach can reduce the ontology heterogeneity and help Semantic Web application developers easily query on various data sets with the integrated ontology.

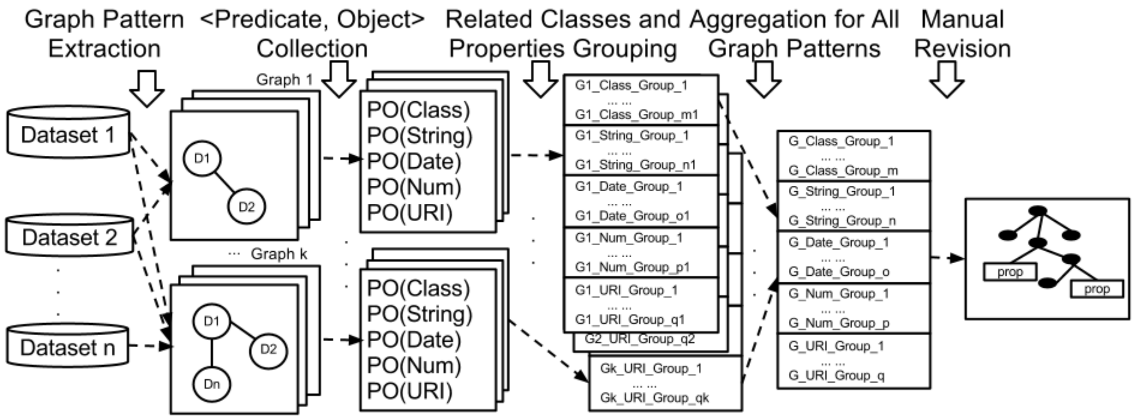

FIGURE 3.3: Architecture of the Graph-Based Ontology Integration.

3.3.2 Detailed Approach

The instances that are interlinked by *owl:sameAs* can be used for constructing graphs and we can apply ontology matching methods on the graphs to find alignments for the related classes and properties [101]. FIGURE 3.3 shows the architecture of the graph-based ontology integration component, which contains 5 main steps. In the following, we describe each step of the graph-based ontology integration framework introduced in [90, 101]:

1. **Graph Pattern Extraction**

   We collect all the instances that have the *owl:sameAs* (SameAs) links to construct graph patterns that can be analyzed for mapping different ontology classes and properties. In the following we list the definitions of the terms: SameAs Triple, SameAs Instance, SameAs Graph and graph pattern:

   Definition 3.1. ***SameAs Triple***. A SameAs Triple is an RDF triple that contains the *owl:sameAs* predicate.

   Definition 3.2. ***SameAs Instance***. A SameAs Instance is a tuple SI = (U, T, L), where U is the URI of the instance that appears in a SameAs Triple <U, *owl:sameAs*, X> or <X, *owl:sameAs*, U>, T is the number of the distinct SameAs Triples that contain U, L is the label of the data set that includes the instance.

   Definition 3.3. ***SameAs Graph***. An undirected SameAs Graph SG = (V, E, I), where V is a set of vertices, which are the labels of data sets that include linked SameAs Instances, $E \subseteq V \times V$ is a set of sameAs edges, and I is a set of URIs of the interlinked SameAs Instances.

Algorithm 1: SameAs graphs extraction.

```
Input   : IndexSI: A set of SameAs Instances
Output: SetSG: A set of SameAs Graphs
Variable: LinkedInst: Linked instances of SI
begin
    SetSG ⟵ ∅
    for SI ∈ IndexSI do
        if SI.visited = false then
            SG ⟵ ∅
            SI.visited ⟵ true
            SG ⟵ SearchGraph(SG, SI)
            SetSG.put(SG)
    return SetSG

SearchGraph(SG, SI)
begin
    SG.V.put(SI.L)
    SG.I.put(SI.U)
    LinkedInst ⟵ ∅
    for X ∈ ( < SI, owl:sameAs, X > ∪ < X, owl:sameAs, SI > ) do
        if X.visited = false then
            LinkedInst.put(X)
    for X ∈ LinkedInst do
        if X.visited = false then
            SG.E ⟵ (SI, X)
            X.visited ⟵ true
            SG ⟵ SearchGraph(SG, X)
    return SG
```

Here, we give an example of the SameAs Graph constructed with the interlinked instances of "France" as shown in Fig 1.2. The SameAs Graph SG_{France} = (V, E, I), where V = {M, D, G, N}, E = {(D, G), (D, N), (G, M), (G, N)}, I = {mdb-country:FR[12], db:France[13], geo:3017382[14], nyt:67...21[15]}. The M, D, G, and N represent the labels of data sets LinkedMDB, DBpedia, Geonames, and NYTimes, respectively.

In order to collect all the SameAs Graphs in the linked data sets, we extract all the SameAs Instances and rank them based on the value of T, which is the number of distinct SameAs Triples. The ranked SameAs Instances are indexed in the *IndexSI*,

[12]mdb-country: http://data.linkedmdb.org/resource/country/
[13]db: http://dbpedia.org/resource/
[14]geo: http://sws.geonames.org/
[15]nyt: http://data.nytimes.com/

TABLE 3.9: Type classification.

Type	Built-in Data Types
String	http://www.w3.org/2001/XMLSchema#string
Date	http://www.w3.org/2001/XMLSchema#date http://www.w3.org/2001/XMLSchema#gYear http://www.w3.org/2001/XMLSchema#gMonthDay
Number	http://www.w3.org/2001/XMLSchema#integer http://www.w3.org/2001/XMLSchema#float http://www.w3.org/2001/XMLSchema#double http://www.w3.org/2001/XMLSchema#int
URI	http://www.w3.org/2001/XMLSchema#anyURI

from which we extract a set of SameAs Graphs $SetSG$ from the linked data sets with Algorithm 1.

In Algorithm 1, for each unvisited SI in the $IndexSI$, we create an empty SameAs Graph SG and construct a SameAs Graph using the search function SearchGraph(SG, SI). We put the L and U of SI into the SG, and then search for the instances linked with SI and put them in the *LinkedInst*. For each unvisited instance X in the *LinkedInst*, we put the edge (SI, X) into the SG, and mark X as visited. Then we recursively search with SG and X, and assign the returned value to SG, until all the instances in the *LinkedInst* are visited. The function SearchGraph(SG, SI) returns a SameAs Graph SG and all the SameAs Graphs constructed with the instances in the $IndexSI$ are stored in the $SetSG$.

Definition 3.4. We say two SameAs Graphs SG_i and SG_j have the same ***graph pattern (GP)***, if $SG_i.V = SG_j.V$ and $SG_i.E = SG_j.E$.

All the same SameAs Graphs consist a graph pattern, from which we can detect related classes and properties. After retrieving graph patterns, we collect useful information for each graph pattern for further analysis.

2. **<Predicate, Object> Collection**
   An instance consists of a collection of RDF triples in the form of <subject, predicate, object>, where the subject is the URI of an instance. Since a SameAs Graph contains linked instances, we collect all the <Predicate, Object> pairs of the interlinked instances as the content of the SameAs Graph. Hereafter, PO is used to represent the <Predicate, Object>.

   In order to avoid comparison between different types of objects, we classify the PO pairs into five different types: Class, String, Date, Number, and URI. The type

TABLE 3.10: *PO* pairs and types for SG_{France}

Predicate	Object	Type
rdf:type	db-onto:Country	Class
rdfs:label	"France"@en	String
foaf:name	"France"@en	String
foaf:name	"Rpublique franaise"@en	String
db-onto:wikiPageExternalLink	http://us.franceguide.com/	URI
db-prop:populationEstimate	65447374	Number
......	......	......
geo-onto:name	France	String
geo-onto:alternateName	"France"@en	String
geo-onto:featureCode	geo-onto:A.PCLI	Class
geo-onto:population	64768389	Number
......	......	......
rdf:type	mdb:country	Class
mdb:country_name	France	String
mdb:country_population	64094000	Number
rdfs:label	France (Country)	String
......	......	......
rdf:type	skos:Concept	Class
skos:inScheme	nyt:nytd_geo	Class
skos:prefLabel	"France"@en	String
nyt-prop:first_use	2004-09-01	Date

of Class can be identified from the predicates rdf:type[16] and skos:inScheme[17]. The other four types of *PO* pairs can be identified from the object values with the built-in data types as listed in TABLE 3.9. Usually the data types of objects are followed by the symbol "^^". If the data types are not given expressively in the RDF triples, we analyze the object values in the following way:

- **Number:** The value consists of all numbers.
- **URI:** Starts with "http://".
- **String:** All the other values that can not be classified.

TABLE 3.10 shows an example of the collected *PO* pairs of the interlinked instances as shown in FIGURE 1.2 and the types of *PO* pairs in the SG_{France}. The first two columns list *PO* pairs and the last column lists the types of the *PO* pairs. The content of the SG_{France} in TABLE 3.10 is used in the next step to discover related classes and properties.

[16]rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
[17]skos: http://www.w3.org/2004/02/skos/core#

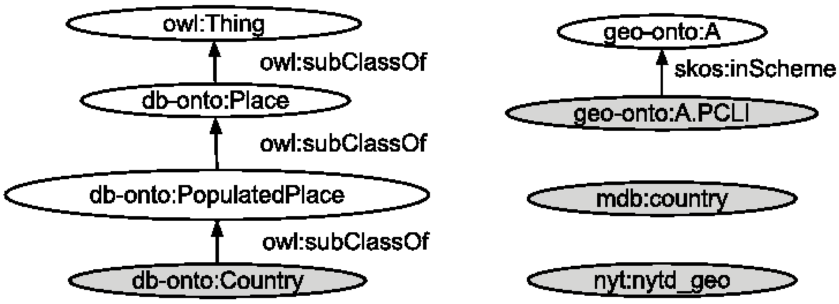

FIGURE 3.4: Collected Classes from SG_{France}.

3. **Related Classes and Properties Grouping**

   In order to find related classes and properties for each graph pattern, we perform different methods on different types of *PO* pairs. In the following, we describe how to discover related classes by checking the subsumption relations and how to find related properties using ontology alignment methods.

   (a) **Related Classes Grouping**

   The ontology classes have subsumption relations such as owl:subClassOf and skos:inScheme. The two triples $< C_1$, owl:subClassOf, $C_2 >$ and $< C_1$, skos:inScheme, $C_2 >$ mean that the concept of C_1 is more specific than the concept of C_2. In order to identify the types of linked instances, we mainly focus on the most specific classes from the linked instances by tracking the subsumption relations. The classes and subsumption relations consist a tree, and the most specific classes are called leaf nodes in the tree. A class which has no subsumption relation is considered as a leaf node.

   From each SameAs Graph, we construct trees with the classes extracted from the *PO* pairs classified in the type Class. Then we group the most specific classes, which are the leaf nodes in the trees. For example, FIGURE 3.4 is a collection of classes extracted from SG_{France}, which are connected with the subsumption relations owl:subClassOf and skos:inScheme. The grey nodes are mdb:country[18], db-onto:Country[19], geo-onto:A.PCLI[20], and nyt:nytd_geo, which are the leaf nodes in FIGURE 3.4. Therefore, we can group these four classes, that are used for describing countries in different data sets.

   (b) **Related Properties Grouping**

   We perform exact and similarity matching methods on the collected *PO* pairs

[18]mdb: http://data.linkedmdb.org/resource/movie/
[19]db-onto: http://dbpedia.org/ontology/
[20]geo-onto: http://www.geonames.org/ontology#

to find out related properties, which are also used as predicates in the PO pairs. This is an extension of the similarity matching method introduced in [91]. In this work, we use different combination of the similarity measures for different types of PO and

i. **Exact Matching for Creating the Initial Sets of PO Pairs**
   The first step in the predicate grouping is to create the initial sets of PO pairs by exact string matching method. For each classified type of PO pairs, we perform a pairwise comparison of PO_i and PO_j , and create the initial sets $S_1, S_2, \ldots, S_k$ by checking whether they have identical predicates or objects. Here, S is a set of PO pairs.
   For example, in TABLE 3.10, the predicates rdfs:label[21], foaf:name, geo-onto:alternateName, mdb:country_name, skos:prefLabel, and geo-onto:name have the same value "France", and the predicate foaf:name has another object, "République franaise"@en. Hence, these six PO pairs are grouped together to create an initial set. After creating initial sets by exact matching, we create an initial set for each PO pair that has not yet been grouped. For instance, nyt-prop:first_use[22] is in an initial set by itself because no predicate has the same object "2004-09-01" and no other PO pairs with the same predict in the data.

ii. **Similarity Matching on the Initial Sets of PO Pairs**
   The identical predicates of PO pairs that are classified into Date and URI can be discovered by exact matching. However, for the types of Number and String, the objects may be slightly different. For instance, the population of a country may have different values in diverse data sets and the values in String may have different representations for the same meaning. In order to find out related initial sets, we apply similarity matching methods on the PO pairs of two initial sets and merge them if the similarity of any two PO pairs is higher than the predefined similarity threshold.
   The string-based and knowledge-based similarity matching methods are commonly used for matching ontologies at the concept level [19]. In this approach, we adopted three string-based similarity measures, namely, JaroWinkler distance [102], Levenshtein distance, and n-gram, as introduced in [94]. String-based similarity measures are applied to compare

[21]rdfs:http://www.w3.org/2000/01/rdf-schema#
[22]nyt-prop: http://data.nytimes.com/elements/.

the objects of PO pairs that are classified in String. $ObjSim(PO_i, PO_j)$ is the similarity of objects between two PO pairs calculated as follows:

$$ObjSim(PO_i, PO_j) = \begin{cases} 1 - \frac{|O_{PO_i} - O_{PO_j}|}{O_{PO_i} + O_{PO_j}} & \text{if } O_{PO} \text{ is Number} \\ StrSim(O_{PO_i}, O_{PO_j}) & \text{if } O_{PO} \text{ is String} \end{cases}$$

where $StrSim(O_{PO_i}, O_{PO_j})$ is the average of the three string-based similarity values and the term O_{PO} indicates the object of PO.

The knowledge-based similarity matching is required to group semantically similar predicates as discussed in [91]. We adopted the same approach to calculate the similarity between two predicates $PreSim(PO_i, PO_j)$ using the formula:

$$PreSim(PO_i, PO_j) = WNSim(T_{PO_i}, T_{PO_j})$$

where T_{PO} indicates the pre-processed terms of the predicates in PO and $WNSim(T_{PO_i}, T_{PO_j})$ is the average of the nine applied WordNet-based similarity values.

$Sim(PO_i, PO_j)$ is the similarity between PO_i and PO_j calculated as follows:

$$Sim(PO_i, PO_j) = \frac{ObjSim(PO_i, PO_j) + PreSim(PO_i, PO_j)}{2}$$

If $Sim(PO_i, PO_j)$ is higher than the predefined similarity threshold, we consider that these two PO pairs are similar and merge two sets S_m and S_n that contain PO_i and PO_j, respectively. In this work, we set the default similarity threshold to 0.5. After comparing all the pairwise initial sets, we remove the initial set S_i if it has not been merged during this process and has only one PO pair.

iii. **Refine Sets of PO Pairs**

The final step of the related properties grouping is to split the predicates of each S_i according to the relation rdfs:domain [5]. Even though the objects or terms of predicates are similar, the predicates may belong to different domains. For further refinement, we determine the frequency of each pruned S_i in all of the data and keep any S_i that appears with a frequency that is higher than the predefined frequency threshold.

From the sets of PO pairs retrieved from each graph pattern, we collect the classes and properties. Then we construct integrated groups of classes and properties for the types Date, String, Number, and URI.

4. **Aggregation of All the Graph Patterns**
   In this step, we aggregate the integrated classes and properties from all the graph patterns to construct a preliminary integrated ontology. During the aggregation process, we keep the rdfs:domain information of properties. The property rdfs:domain infers which class the groups of properties belong to. Then we aggregate the integrated groups of classes and properties according to the following rules:

   (a) **Select A Term for Each Set**
   To perform automatic term selection, we pre-process all the terms of the classes and properties in each set by tokenization, stop words removal, and stemming. We keep the original terms because sometimes a single word is ambiguous for representing a set of terms and choose the longest and most frequent term as introduced in the previous chapter. The predicate ex-onto:ClassTerm is designed to represent a class, where the "ClassTerm" is automatically selected and starts with a capitalized character. The predicate ex-onto:propTerm is designed to represent a property, where the "propTerm" is automatically selected and starts with a lowercase character.

   (b) **Construct Relations**
   We use the predicate ex-prop:hasMemberClasses to link the integrated classes with the class ex-onto:ClassTerm, and use ex-prop:hasMemberDataTypes to link the integrated properties with the property ex-onto:propTerm. Here we use the ex-prop:hasMemberClasses and ex-prop:hasMemberDataTypes instead of the existing OWL properties owl:equivalentClass or owl:equivalentProperty in order to reduce the number of triples to connect the related classes and properties.

   For example, if the number of integrated classes or properties in a set S_i is n, we need $n*(n-1)$ triples to connect all the pairs of classes or properties using the owl:equivalentClass or owl:equivalentProperty. However, in our approach we only need n triples with the properties ex-prop:hasMemberClasses and ex-prop:hasMemberDataTypes.

(c) **Construct A Preliminary Integrated Ontology**

A preliminary integrated ontology is automatically constructed with the integrated sets of related classes and properties, the selected terms ClassTerm and propTerm, and two relation predicates ex-prop:hasMemberClasses and ex-prop:hasMemberDataTypes.

5. **Manual Revision**

The automatically constructed preliminary integrated ontology includes related classes and properties from different data sets. However, not all the terms of classes and properties are properly selected, and there are some missing statements of rdfs:domain. Hence, we need experts to work on revising the integrated ontology by choosing a proper term for each group of classes or properties, and by amending wrong groups of classes or properties. Since the integrated ontology is much smaller than the original ontology schema, it is a lightweight work.

The graph-based ontology integration component can discover related classes and properties from various data sets. By analyzing the extracted graph patterns, we detect related classes and properties, which are classified into different data types: Date, URI, Number, and String. Similar classes are integrated by tracking the subsumption relations and different similarity matching methods are applied on different types of PO pairs to retrieve similar properties. We automatically integrate related classes and properties for each graph pattern, and then aggregate all of them to construct a preliminary integrated ontology.

3.3.3 Experimental Evaluation

In this section, we will discuss some experimental evaluations performed with graph-based ontology integration approach. We use the same experimental data sets as in Section 3.2. Then we analyze the graph patterns that are extracted from the linked instances, and analyze the characteristics of interlinked instances with the integrated ontologies at the class-level. The analysis at the property-level is also discussed, from which we can detect standard properties. At last, we compare the Mid-Ontology learning approach and the graph-based ontology integration approach to show the advantages of the latter method.

1. **Graph Patterns of Linked Instances**

In this experiment, we analyze the graph patterns extracted with the experimental

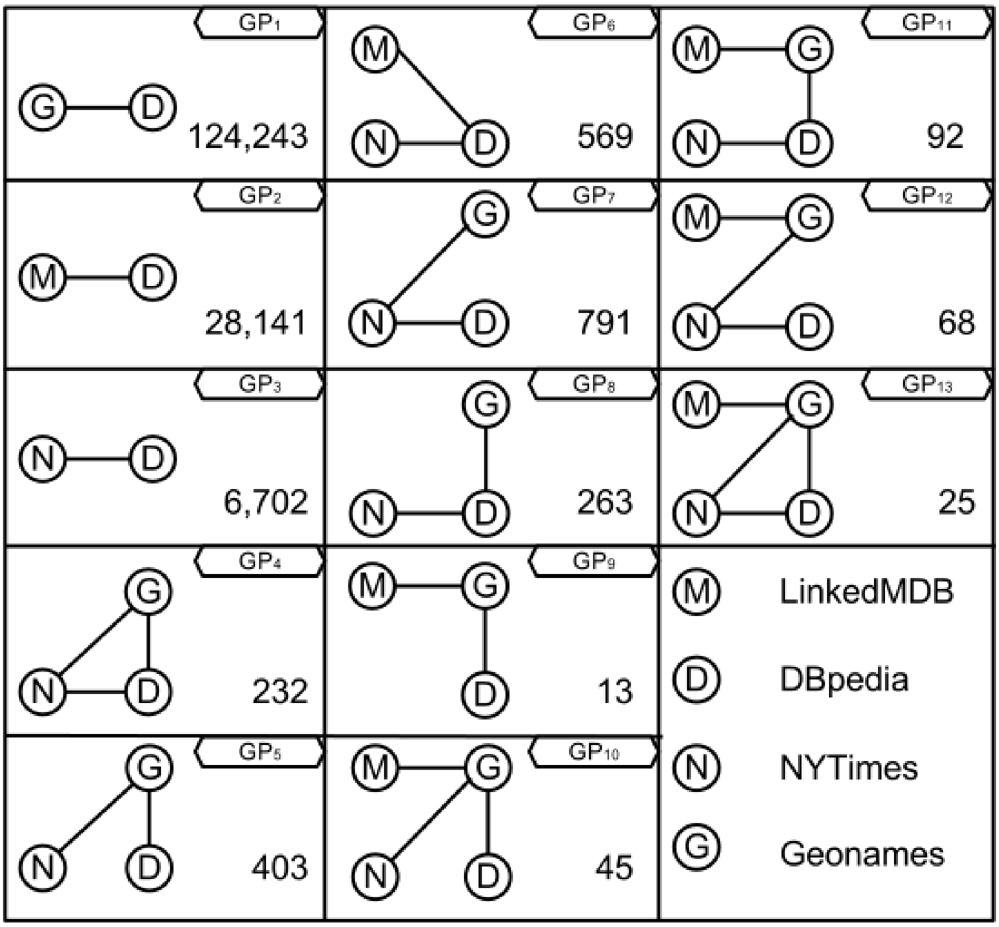

FIGURE 3.5: SameAs Graph Patterns.

data sets to observe how the data sets are interlinked. We retrieved 13 different graph patterns from the SameAs Graphs as listed in Figure 3.5. The labels of nodes M, D, N, and G represent the data sets LinkedMDB, DBpedia, NYTimes, and Geonames, respectively. The number on the right side of each graph pattern is the number of SameAs Graphs, which is used to decide the frequency threshold for refining sets of *PO* pairs.

As shown in Figure 3.5, the top 3 most frequent graph patterns are GP_1, GP_2, and GP_3, which contain only two nodes, (Geonames and DBpedia), (LinkedMDB and DBpedia), and (NYTimes and DBpedia), respectively. Four graph patterns GP_4, GP_5, GP_7, and GP_8 contain nodes from Geonames, DBpedia, and NYTimes with different set of edges. The other graph patterns with three nodes are GP_6 and GP_9, where GP_6 contains LinkedMDB, NYTimes, and DBpedia, while GP_9 contains LinkedMDB, Geonames, and DBpedia. The graph patterns GP_{10}, GP_{11}, GP_{12}, and GP_{13} contain all the nodes from four different data sets with different set of edges.

Table 3.11: Characteristics of graph patterns.

Class Type	Graph Pattern
Actor	GP_2, GP_6
Person(Athlete, Politician, etc)	GP_3
Organization/Agent	GP_1, GP_3, GP_8
Film	GP_2
City/Settlement	GP_1, GP_4, GP_5, GP_7, GP_8
Country	GP_9, GP_{10}, GP_{11}, GP_{12}, GP_{13}
Place(Mountain, River, etc)	GP_1, GP_3, GP_7

2. **Class-level Analysis**

   In order to observe what kind of instances are interlinked in each graph pattern in Figure 3.5, we analyze the integrated classes in the third step of our approach. We classified the characteristics of graph patterns into seven different types: Actor, Person (except Actor), Organization/Agent, Film, City/Settlement, Country, and Place as shown in Table 3.11. The first column in Table 3.11 lists the class types and the second column lists the graph patterns, which have the corresponding class type. In total, we created 48 integrated classes from the experimental data sets.

   The type of instances shared by the four data sets is Country. The integrated class for Country is ex-onto:Country, which contains geo-ontoA.PCLI, geo-onto:A.PCLD, mdb:country, db-onto:Country, and nyt:nytd_geo. As we can see in Table 3.11, the graph patterns GP_9, GP_{10}, GP_{11}, and GP_{12} are sub-graphs of GP_{13}. However, GP_{13} is not a complete graph and there are missing links between (M, N) and (M, D). Hence, with the ex-onto:Country, we can link missing links of countries among these four data sets.

   Furthermore, with the integrated classes, we can discover missing class information of linked instances. For instance, the db:Shingo_Katori is only described as a musical artist, but in fact he is also an actor and the DBpedia instance has a link to the mdb-actor:27092[23]. Hence, we should add the class db-onto:Actor to the instance db:Shingo_Katori, because all the instances linked with mdb-actor should be an actor unless it is a wrong linkage.

   The main classes of each data set can be recognized from the integrated classes. The NYTimes is mainly categorized into person, organization, and place. The LinkedMDB is mainly categorized into movie, actor, and country. The main classes appeared in the graph patterns for the Geonames are A(country, administrative region), P (city, settlement), T (mountain), S (building, school), and H (Lake, river),

[23]mdb-actor: http://data.linkedmdb.org/resource/actor/

where A, P, T, S, and H are the feature classes used in the Geonames ontology. The main classes of DBpedia linked with other data sets are person (artist, politician, athlete), organization (company, educational institute, sports team), work (film), and place (populated place, natural place, architectural structure).

3. **Property-level Analysis**

   Only considering related classes from different data sets is not enough to detect identical instances. Hence, we analyze the properties of integrated ontologies to find out what kind of properties are critical to identify related instances from different data sets. We integrated 38 groups of properties from the graph patterns in Figure 3.5. However, 15 groups of the properties have no information about rdfs:domain; it is difficult to find which instances should be described with the properties without rdfs:domain. Hence, missing domain information is added to the groups of properties during the manual revision. The manual revision on the integrated ontology is a light work, which only takes few hours.

   The experts can only match the given ontologies, but some of the ontology schemas might be mistakenly used in the real data. If we want to link a person in different data sets, we can combine the classes that indicate a person with another property such as the birth date, the place of birth, or the name, etc. However, there are various properties exist to describe the same kind of property.

   For example, among the 7 different properties that indicate the birthday of a person as listed in TABLE 3.12, only the property "db-onto:birthDate" has the domain definition with the class db-onto:Person and has the highest frequency of usage that appeared in 287,327 DBpedia instances. The second column in TABLE 3.12 represents the number of distinct instances use the property listed in the first column. From the definitions of the properties and the number of instances which contain the corresponding properties, we can assume that the properties except "db-onto:birthDate" are mistakenly used when the data providers publish the DBpedia data.

   Furthermore, the property "db-onto:birthDate" is well defined with rdfs:domain and has the highest frequency of usage in the DBpedia instances. Therefore, we can suggest "db-onto:birthDate" as the standard property to represent the birthday of a person, and correct the other properties with this standard property. We can integrate heterogeneous ontologies and also can recommend standard ontology schemas so that we can validate if the ontologies are correctly used in the data.

TABLE 3.12: Predicates grouped in ex-prop:birthDate.

Property	Number of Instances	rdfs:domain
db-onto:birthDate	287,327	db-onto:Person
db-prop:datebirth	1,675	N/A
db-prop:dateofbirth	87,364	N/A
db-prop:dateOfBirth	163,876	N/A
db-prop:born	34,832	N/A
db-prop:birthdate	70,630	N/A
db-prop:birthDate	101,121	N/A

TABLE 3.13: Links not included in the Mid-Ontology learning approach.

Links between Data Sets	Graph Pattern
Geonames and NYTimes	GP_5, GP_7, GP_{10}, GP_{12}
Geonames and LinkedMDB	GP_9, GP_{10}, GP_{11}, GP_{12}, GP_{13}

Other than recommending standard properties, we also successfully integrated different property descriptions from diverse data sets. For instance, the properties such as geo-onto:population, db-onto:populationTotal, mdb:country_population, and other nine DBpedia properties are integrated into the property ex-prop:population. By combining the ex-onto:Country and ex-prop:population, we can detect the same country or countries with similar population.

4. **Comparison with Mid-Ontology Learning Approach**

In this section, we compare the graph-based ontology integration approach with the Mid-Ontology learning approach introduced in Section 3.2. In the Mid-Ontology learning approach, we used DBpedia as a hub data set, and collected all the SameAs links that contain the instances of DBpedia. Hence, we were not able to collect the links between other data sets if they are not directly connected with DBpedia. The links that can not be found in the Mid-Ontology learning approach, but can be collected with graph-based ontology integration are listed in Table 3.13. The links between Geonames and NYTimes from the graph patterns GP_5, GP_7, GP_{10}, GP_{12}, and the links between Geonames and LinkedMDB from GP_9, GP_{10}, GP_{11}, GP_{12}, GP_{13} are retrieved with graph-based ontology integration, but can not be retrieved in the Mid-Ontology learning approach.

Furthermore, in the Mid-Ontology learning approach we did not identify the types of objects and considered all the object values as String during the predicates grouping step. Hence, even the numbers are compared with string-based similarity matching and it results in a wrong similarity value. For example, the similarity between 5000 and 4999 is 0 in the Mid-Ontology learning approach, but with the graph-based ontology integration approach the similarity is approximately 1. In the graph-based ontology integration, we classified the types of PO pairs into Class, String, Date, Number, and URI. Then we performed different methods to integrate related classes and properties from the PO pairs. For instance, we discover the most specific classes by tracking subsumption relations to find related classes. In order to detect related properties, we applied only exact matching on the types of Date and URI, and applied similarity matching on the types of Number and String in a different way. Furthermore, we analyzed the linked instances at both class and property level, while the Mid-Ontology learning approach only analyzed at the property level.

The integrated Mid-Ontology was constructed with properties only, without any relations and classes. However, in the graph-based ontology integration, we also integrated classes of different ontologies and linked the integrated properties with the classes. Hence, it is easier to observe the characteristics of the interlinked instances with the integrated classes. Furthermore, the relations between properties and classes indicate what kind of core properties can well describe the instances and help detecting related instances.

Moreover, many important links were not found in the Mid-Ontology learning approach, because they used only one common threshold to find frequent groups of related predicates. In this graph-based ontology integration, the frequency threshold is calculated based on the number of SameAs Graphs in each graph pattern. Therefore, we can retrieve more properties that were not discovered in the Mid-Ontology learning approach. For example, the runtime of a movie extracted from the links between LinkedMDB and DBpedia are mdb:runtime, db-prop:runtime[24], db-onto:runtime, and db-onto:Work/runtime. This is retrieved with graph-based ontology integration, but can not be retrieved in the Mid-Ontology learning approach. With the approach introduced in Mid-Ontology learning approach, only 105 predicates are retrieved and classified into 22 groups, while with the graph-based ontology integration, 367 properties are integrated into 38 groups.

[24]db-prop: http://dbpedia.org/property/

3.4 Summary

In this chapter, we first introduced the Mid-Ontology learning approach involves the use of Linked Open Data and integrates diverse ontology schemas without learning them manually. The main procedures of the Mid-Ontology learning approach are linked instances collection, the ontology predicate grouping process, and Mid-Ontology construction. The predicate grouping algorithm applied lexical similarity matching to collect similar predicates and implemented the relation extraction method to refine predicate sets. The Mid-Ontology learning approach can automatically extract related predicates between linked data sets and integrate them into the Mid-Ontology.

However, there are some drawbacks in the Mid-Ontology learning approach such as a hub data set is necessary to extract linked instances and only related predicates are discovered in this approach. Furthermore, the ontology similarity matching methods treat all the types of property values as String, which in fact can be numbers, dates, or URIs. In order to overcome these limitations, we introduced the graph-based ontology integration in the next section.

The graph-based integration approach contains 5 main steps, which is an extension of the Mid-Ontology learning approach. At first, we extract graph patterns by crawling the interlinked instances. Then from the extracted graph patterns, we detect related classes and properties that are classified into different data types: String, Date, Number, and URI. Similar classes are integrated by tracking subsumption relations and different similarity matching methods are applied on different types of PO pairs to retrieve similar properties. We automatically integrate related classes and properties for each graph pattern, and then aggregate all of them to construct an integrated ontology. Minor manual revision by an expert is required to add missing domain information of properties and to correct groups of integrated classes and properties.

Experiments show that the graph-based ontology integration can retrieve more information than the Mid-Ontology learning approach, and we can understand the characteristics of the interlinked instances at both class and property levels. By combining related classes and properties from various data sets, we can find missing SameAs links and reduce the ontology heterogeneity problem that help Semantic Web application developers easily understand the relations between different ontologies without manual inspection.

Chapter 4

Enriching the Integrated Ontology

The integrated ontology constructed using graph-based ontology integration approach successfully integrated related ontology schemas from various data sets. However, the integrated ontology may miss some core ontology schemas that are important for describing instances of the data sets. Furthermore, because of the missing domain or range information in the definitions of the ontology schemas, it is difficult to observe the relations between properties and classes or the ranges of property values. In order to solve these two problems: the difficulty in identifying core ontology schemas and missing domain or range information, we propose two approaches in this chapter.

We start with the motivation of our research in Section 4.1. Then in Section 4.2, we introduce the machine-learning-based approach which contains Decision Table and Apriori. In Section 4.3, we briefly introduce how to add domain and range information automatically by analyzing samples of the instances. Experimental results show that machine-learning-based approach can effectively retrieve core ontology schemas, which are important for describing instances. Furthermore, the integrated ontology constructor successfully adds missing domain and range information with additional annotations.

4.1 Motivation

Besides integrating heterogeneous ontologies, we also need to know core ontology schemas of each data set to retrieve rich information by accessing to various data sets. As mentioned in Chapter 1, we still have two remaining problems to solve: the difficulty in identifying core ontology schemas and missing domain or range information problem. In the following, we propose our solution to solve these two problems.

1. **Difficulty in Identifying Core Ontology Schemas**
   Other than retrieving related classes and properties, we also need frequently used core classes and properties of the ontologies to construct a high-quality integrated ontology. This can also help the Semantic Web developers easily identify the core ontology schemas for describing instances in each data. The core ontology schemas can assist us to construct SPARQL queries and to add missing core information in the data sets.

   Machine learning methods such as association rule learning and rule-based classification can be applied to discover core properties for describing instances in a specific class. Apriori is a well-known algorithm for learning association rules in a big database [24], while the rule-based learning method - Decision Table can retrieve a subset of properties that leads to high prediction accuracy with cross-validation [25]. By applying machine learning approaches to the linked data sets, we can retrieve core ontology schemas that are important for describing instances in the data sets. The detailed machine-learning-based approach is described in Section 4.2.

2. **Missing Domain or Range Information**
   The relations between the ontology classes and properties are described with the property rdfs:domain, which indicates that the properties are designed to be used for describing the instances of a specific class. Furthermore, the range information of the values can help users better understand the data sets. Besides the domain and range information, we should also retrieve the description of each ontology schema to construct an easily understandable integrated ontology.

   We can randomly select some samples of the instances that contain the retrieved classes and properties and analyze the usage of the domain information and values. We can also retrieve the default range information of the properties or analyze the values of the properties from the sample instances, which are mainly categorized into String and Resource. By analyzing samples of the instances, we can reduce the analysis time and also be able to retrieve the domain and range information. The default annotations make the integrated ontology easily understandable. The detailed approach of the integrated ontology constructor is described in Section 4.3.

4.2 Machine-Learning-Based Approach

4.2.1 Objective

In this section, we propose machine-learning-based approach to identify core ontology schemas from various data sets [103]. Here, the core ontology schemas indicates top-level ontology classes and frequently used core ontology properties.

Top-level ontology class: Each instance is described using one top-level class and subclasses of the top-level class that have more specific class information. Hence, we consider the top-level classes as part of the core ontology schemas.

Frequent core ontology property: Some properties are frequently used in every instance or instances of some specific classes. Hence, we consider these properties as the frequent core ontology properties, which are frequently used for describing instances.

The top-level classes can be retrieved by tracking subsumption relations in the ontology or by retrieving categories in the data sets which do not contain ontologies. By applying machine learning methods, we can find core properties that are frequently used to describe instances of a specific class. The Decision Table is a rule-based algorithm that can retrieve a subset of core properties and the Apriori algorithm can find a set of associated properties that are frequently used for describing instances. Hence, we apply the Decision Table and the Apriori algorithm to retrieve frequent core properties from the linked data sets.

4.2.2 Detailed Approach

Although, the graph-based ontology integration method can retrieve related classes and properties from different ontologies, it may miss some core ontology schemas that are important to describe instances in the data sets. Therefore, we need another method to extract top-level classes and frequent core properties, which are essential for describing instances. We apply two machine learning methods - Decision Table and Apriori to extract the core ontology schemas.

In order to perform the machine learning methods, we randomly select a fixed number of instances for each top-level class from the data sets. For the data sets built based on ontology schemas, we track the subsumption relations to retrieve the top-level classes. For instance, we track owl:subClassOf subsumption relation to retrieve the top-level classes in DBpedia and track skos:inScheme in Geonames. However, some data sets use categories without any structured ontology schema. For this kind of data sets, we use the categories as the top-level classes. For example, NYTimes instances are only categorized into people, locations, organizations, and descriptors. We use this strategy to collect the top-level classes in each data set, and then extract properties that appear more than the frequency threshold θ. The selected instances, properties, and top-level classes are used for performing machine learning methods.

1. **Decision Table**

   The Decision Table is a simple rule-based supervised learning algorithm, which leads to high performance with simple hypothesis [25]. The Decision Table algorithm can retrieve a subset of core properties that can predict unlabeled instances with a high accuracy. Therefore, the properties retrieved by the Decision Table play an important role in the data description.

   We convert the instances of the linked data sets into data that is adaptable to the Decision Table algorithm. The data consists of a list of weights of the properties and the class labels, where the weight represents the importance of a property in an instance and the labels are top-level classes. The weight of a property in an instance is calculated in a similar way as the Term Frequency - Inverse Document Frequency (TF-IDF), which is often used as a weighing factor in the information retrieval and text mining [104]. The TF-IDF value reflects how important a word is to a document of a collection or a corpus. The weight of each property in an instance is defined as the product of the property frequency (PF) and the inverse instance frequency (IIF) in a similar way as the TF-IDF. The property frequency $pf(prop, inst)$ is the frequency of the property *prop* in the instance *inst*.

   The inverse instance frequency of the property *prop* in the data set D is $iif(prop, D)$, calculated as follows:

$$iif(prop, D) = \log \frac{|D|}{|inst_{prop}|}$$

where $inst_{prop}$ indicates an instance that contains the property *prop*. The value of $iif(prop,\ D)$ is the logarithm of the ratio between the number of instances in D and the number of instances that contain the *prop*. If *prop* appears in *inst*, the weight of *prop* is calculated according to the following equation:

$$weight(prop, inst) = pf(prop, inst) \times iif(prop, D)$$

The properties retrieved with the Decision Table in each data set are critical for describing instances in the data set. Hence, we use these retrieved properties and top-level classes as parts of the final integrated ontology.

2. **Apriori**

The association rule learning method can extract a set of properties that occur frequently in the instances of a specific class. Apriori is a classic association rule mining algorithm, which is designed to operate on the databases of transactions. A frequent itemset is an itemset whose support is greater than the user-specified minimum support. Each instance in a specific class represents a transaction, and the properties that describe the instance are treated as items. Hence, the frequent itemsets represent the frequently used properties for describing the instances of a specific class. The frequent core properties can be recommended to the data publishers or help them find missing important descriptions of the instances.

For each instance, a top-level class and all the properties that appear in the instance are collected as a transaction. The Apriori algorithm can extract associated sets of properties that occur frequently in the instances of a top-level class. Hence, the retrieved sets of properties are essential for describing the instances of a specific class. Furthermore, we can either identify commonly used properties in each data set or unique properties used in the instances of each class. Therefore, the properties extracted with the Apriori algorithm are necessary for the integrated ontology.

4.2.3 Experimental Evaluation

In this section, we introduce the experimental data sets and then discuss experimental results with the Decision Table and the Apriori algorithm that retrieve top-level classes and frequent core properties. Then, we compare the extracted ontology classes and properties between the graph-based ontology integration and machine learning approach.

TABLE 4.1: Data sets for experiments.

Data Set	Instances	Selected Instances	Class	Top-level Class	Property	Selected Property
DBpedia	3,708,696	64,460	241	28	1385	840
Geonames	7,480,462	45,000	428	9	31	21
NYTimes	10,441	10,441	5	4	8	7
LinkedMDB	694,400	50,000	53	10	107	60

1. **Experimental Data**

   We use the same data sets as in previous Chapter 3: DBpedia (v3.6), Geonames (v2.2.1), NYTimes, and LinkedMDB from the LOD cloud to evaluate our framework. DBpedia is a cross-domain data set with about 8.9 million URIs and more than 232 million RDF triples. Geonames is a geographic domain data set with more than 7 million distinct URIs. NYTimes and LinkedMDB are both from media-domain with 10,467 and 0.5 million URIs, respectively.

   The number of instances in our database are listed in the second column of TABLE 4.1. The graph-based ontology integration component uses all the instances in the data sets. However, for the machine learning methods, we randomly choose samples of the data sets to speed up the modeling process as well as to concern unbiased data size for each top-level class. We randomly selected 5000 instances per top-level class in Geonames and LinkedMDB, 3000 instances per top-level class in DBpedia, and used all the instances in NYTimes. The number of the selected instances from DBpedia is less than 84,000, because some classes include less than 3000 instances.

   The original number of classes and properties, the number of the top-level classes and the selected properties for the machine learning methods are listed in the TABLE 4.1. We track the subsumption relations such as owl:subClassOf and skos:inScheme to collect the top-level classes. Since there are a big number of properties in the data sets, we filter out infrequent properties that appear less than the frequency threshold θ. For each data set, we manually set a different frequency threshold θ as $\sqrt{n}$, where n is the total number of instances in the data set.

2. **Evaluation of Decision Table**

   The Decision Table algorithm is used to discover a subset of features that can achieve high prediction accuracy with cross-validation. Hence, we apply the Decision Table to retrieve core properties that are essential in describing instances of the data sets. For each data set, we perform the Decision Table algorithm to retrieve core properties by analyzing randomly selected instances of the top-level classes. In this

TABLE 4.2: Results for the Decision Table algorithm.

Data Set	Average Precision	Average Recall	Average F-Measure	Selected Properties
DBpedia	0.892	0.821	0.837	53
Geonames	0.472	0.4	0.324	10
NYTimes	0.795	0.792	0.785	5
LinkedMDB	1	1	1	11

experiment, we evaluate whether the retrieved sets of properties are important for describing instances by testing the performance on instance classification.

In Table 4.2, we listed the percentage of the weighted average of precision, recall, and F-measure. Precision is the ratio of the correct results to all the results retrieved, and recall is the percentage of the retrieved relevant results to all the relevant results. The F-measure is a measure of a test's accuracy, that considers both the precision and the recall. The F-measure is the weighted harmonic mean of the precision and recall, calculated as:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The F-measure reaches its best value at 1 and worst value at 0. A higher F-measure value means the retrieved subset of properties can well classify instances, which implies that these properties are important for describing the instances of a specific class. A lower F-measure fails to classify some instances, because the retrieved properties are commonly used in every instance. In the following, we will discuss the experimental results in each data set using the Decision Table algorithm.

- **DBpedia.** The Decision Table algorithm extracted 53 DBpedia properties from 840 selected DBpedia properties. For example, the DBpedia properties db-onto:formationYear, db-prop:city, db-prop:debut, and db-prop:stateName are extracted from DBpedia instances. The precision, recall, and F-measure on DBpedia are 0.892, 0.821, and 0.837, respectively.
- **Geonames.** We retrieved 10 properties from 21 selected Geonames properties, such as geo-onto:alternateName, geo-onto:countryCode, and wgs84_post:alt, etc. Since all the instances of Geonames are from geographic domain, the Decision Table algorithm can not well distinguish different classes with these commonly used properties. Hence, the evaluation results on Geonames are very low with 0.472 precision, 0.4 recall, and 0.324 F-measure.

TABLE 4.3: Examples of the retrieved properties with the Apriori algorithm.

Data Set	Class	Properties
DBpedia	db-onto:Event	db-onto:place, db-prop:date, db-onto:related/geo.
	db-onto:Species	db-onto:kingdom, db-onto:class, db-onto:family.
	db-onto:Person	foaf:givenName, foaf:surname, db-onto:birthDate.
Geonames	geo-onto:P	geo-onto:alternateName, geo-onto:countryCode.
	geo-onto:R	wgs84_pos:alt, geo-onto:name, geo-onto:countryCode.
NYTimes	nyt:nytd_geo	wgs84_pos:long.
	nyt:nytd_des	skos:scopeNote.
LinkedMDB	mdb:actor	mdb:performance, mdb:actor_name, mdb:actor_netflix_id.
	mdb:film	mdb:director, mdb:performane, mdb:actor, dc:date.

- **NYTimes.** Among 7 properties used in the data set, 5 of them are retrieved using the Decision Table algorithm. We retrieved skos:scopeNote, nyt:latest_use, nyt:topicPage, skos:definition, and wgs84_pos:long. In NYTimes, there are only few properties for describing news articles and most of them are commonly used in every instance. The results of the cross-validation test with NYTimes are 0.795 precision, 0.792 recall and 0.785 F-measure.
- **LinkedMDB.** The algorithm can classify all the instances in the LinkedMDB correctly with the 11 properties selected from 60 properties. Other than the commonly used properties such as foaf:page, rfs:label, we also extracted some unique properties such as director_directorid, mdb:writer_writerid, and mdb:performance_performanceid, etc.

The experimental results show that the properties extracted from LinkedMDB can well distinguish the types of instances, because the properties are unique IDs of different types of instances. The performance of classification in DBpedia and NYTimes is lower because the retrieved properties contain commonly used properties that reduce the classification performance. Since most of the instances in Geonames are described using common properties, the performance in predicting the types of instances is very low. We observed that the Decision Table either retrieves unique properties or commonly used properties, where both are important for describing instances. Decision Table retrieves core properties in the data sets, therefore it is necessary for creating an integrated ontology.

3. **Evaluation of Apriori**

The Apriori algorithm is a classic algorithm for retrieving frequent itemsets based on the transaction data. We list representative examples of the top-level class and the corresponding set of properties that are retrieved using the Apriori algorithm.

Furthermore, we analyze the performance of the algorithm in each data set with the examples. For the experiment, we use the parameters upper and lower bound of minimum support as 1 and 0.2, respectively. We set the minimum confidence as 0.9. With a lower minimum support, we can retrieve more properties that frequently appear in the data.

We retrieved frequently appeared core properties using the Apriori algorithm. Some examples are listed in Table 4.3. The first column lists the experimental data sets, and the second column lists samples of the top-level classes in each data set. The third column lists some of the retrieved properties from each top-level class. From Table 4.3, we observed that the place, date and geographic properties are important for describing event instances. The best-known taxonomies such as kingdom, class, and family are also extracted by analyzing the data of species. From the LinkedMDB, we extracted mdb:actor_name, mdb:performance, and mdb:actor_netflix_id that are critical for distinguishing different instances. Furthermore, the properties of director, performance, actor and date of a film are extracted from the instances in the class mdb:film.

In DBpedia and LinkedMDB, we retrieved some unique properties for each class. However, for Geonames and NYTimes, we only retrieved some commonly used properties in the data sets. From the instances of Geonames, we found the properties geo-onto:alternateName, wgs84_pos:alt, and geo-onto:countryCode, that are commonly used in the Greonames instances. NYTimes contains few properties and most of them are commonly used in every instance, except the property wgs84_pos:long in the nyt:nytd_geo class and the property skos:scopeNote in the nyt:nytd_des class. Hence, the weighted average F-measure of the classes nyt:nytd_geo and nyt:nytd_des are much higher than other classes.

We retrieved frequent sets of properties in most of the cases except in the db-onto:Planet class. Because db-onto:Planet contains 201 different properties for describing instances, which are sparsely used. In addition, we only retrieved db-onto:title and rdfs:type from db-onto:PersonFunction and only rdfs:type property from db-onto:Sales. This is caused by the lack of descriptions in the instances: most of the instances in db-onto:PersonFunction and db-onto:Sales only defined the class information without other detailed descriptions.

The set of properties retrieved from each class implies that the properties are frequently used for instance description of the class. Hence, for each property *prop* retrieved from the instances of class *c*, we automatically add <*prop*, rdfs:domain,

TABLE 4.4: Extracted Classes and Properties

	Graph-Based Integration	Machine Learning Approach	
		Decision Table	Apriori
Class	97	50 (38 new)	50 (38 new)
Property	357	79 (49 new)	119(80 new)

c> to assert that *prop* can be used for describing instances in the class *c*. Therefore, we can automatically recommend missing core properties for an instance based on its top-level class.

The Apriori algorithm can retrieve associated core properties from the data sets and we can automatically add domain information with the results of Apriori. However, in some classes it failed to discover properties because of the lack of the instance descriptions.

4. **Comparison of Extracted Classes and Properties**

   The graph-based ontology integration framework introduced in [101] only focuses on the related classes and properties acquisition, that may miss some core properties and classes. Hence, we applied machine learning methods to find out core properties for describing instances. The second column of Table 4.4 lists the number of classes and properties retrieved with the graph-based integration method. The next two columns list the number of classes and properties retrieved with the machine learning methods - Decision Table and Apriori.

   With the graph-based ontology integration framework, we retrieved 97 classes and 357 properties, which are grouped into 49 and 38 groups, respectively. Both of the Decision Table and the Apriori algorithms are performed on 50 selected top-level classes, among them 38 are not retrieved in the graph-based ontology integration. With the Decision Table, we extracted 79 properties, where 49 are not found in the graph-based ontology integration. The Apriori algorithm discovered 119 properties in total, where 80 properties are newly added. With the same data sets, Apriori can retrieve more properties than the Decision Table algorithm. Among the newly retrieved properties, 33 properties are retrieved from both Decision Table and Apriori.

   By adding machine-learning-based approach component to the graph-based ontology integration, the integrated ontology become more concrete with groups of related classes and properties, top-level classes, and core properties that are frequently used in instances. For each retrieved property, we automatically added property type definition. Furthermore, for the properties retrieved with the Apriori results, we automatically added domain information to indicate the relations between properties and classes.

Since most of the ontology matching tools fail to find alignments for the datasets that do not have a well designed ontology schema [18], we cannot use them to find alignments among DBpedia, Geonames, NYTimes, and LinkedMDB. The failure in the ontology alignment is caused by some ontologies that have ambiguous meaning of the concepts or the absence of corresponding concepts in the target dataset. However, our approach can find alignments for the poorly structured datasets by analyzing the contents of the interlinked instances.

4.3 Integrated Ontology Constructor

4.3.1 Objective

The graph-based ontology integration component outputs groups of related classes and properties and the machine-learning-based approach outputs a set of core properties retrieved by the Decision Table and a set of properties along with a top-level class retrieved using the Apriori. These extracted ontology schemas can be used to construct an integrated ontology for various data sets. However, there are many missing domain information and many missing range information of properties. An ontology without domain and range information is difficult to understand the ontology schemas and the relations between properties and classes. Hence, we need to add these missing information for constructing a robust integrated ontology.

In order to construct an easily understandable ontology, we enrich the definitions of the retrieved ontology classes and properties by adding annotations, domains, and ranges information. The domain information of the properties is automatically added using the results of the Apriori algorithm in the previous Section 4.2.

4.3.2 Detailed Approach

In order to construct an easily understandable integrated ontology, we automatically add missing domain and range information along with default annotation in the integrated ontology constructor. This integrated ontology constructor mainly consists of the ontology enrichment, ontology merger, and naming validator. In the following, we will describe each part in details.

1. **Ontology Enrichment**
   Most of the retrieved ontology classes and properties lack of clear definitions. Hence, only collecting the default definitions of them is not adequate for users to understand the meanings of the ontologies, and the relations between the properties and classes. We enrich the retrieved classes and properties in the following way:

   - **Annotation**: We collect all the default annotation definitions of the classes and properties from the data sets. In this process, for each group of the classes and properties, we simply remove the duplicated annotations and simple annotations that are included in more comprehensive ones.
   - **Domain**: The domain information of a property should be included in the integrated ontology because it indicates the relation between a property and a class. It can help users easily understand what kinds of properties can be used for a specific class. In order to retrieve the domain information of a property, we randomly select m number of samples of the instances, which have the property. Then we collect all the class information of the sample instances and recursively do the sampling process for n times. The class information can be collected by tracking with the property rdf:type and skos:inScheme, etc.

     Then we analyze the collected class information to retrieve proper domain information for a property. We choose the most frequently appeared classes as the domains of a property, which are mostly appeared in every sample instance. However, we observed that some classes are also frequently used, but are missing in few instances. Hence, we set a frequency threshold for the domain retrieval as 0.95 * $Freq_{top}$, where $Freq_{top}$ is the highest frequency of a class. If we could not retrieve the frequent class information or default definition of the domain information, we set owl:Thing as the domain information.
   - **Range**: The range information of a property is also important for the users when they create SPARQL queries or publish data sets. However, most of the ranges are missing and sometimes the values are published in various ranges. In order to retrieve the range information, we also use the same sample instances as described above. Then, we analyze the values of the properties in the sample instances.

We can retrieve the built-in data types by tracking the symbol "^^". For other values that do not expressively show the data types, we classify them into two types: Resource and String. If the value contains resource information, we classify it as Resource, otherwise we consider it as String.

2. **Ontology Merger**
We adopt OWL 2 for constructing the integrated ontology. During the merging process, we also add relations between classes and properties so that we can easily identify what kinds of properties are used to describe the instances of a specific class. We obey the following rules to construct the integrated ontology, where "ex-onto" and "ex-prop" are the prefixes of the integrated ontology.

- **Class**
Related classes are collected from the graph-based ontology integration component and the top-level classes in each data set are collected from the machine-learning-based approach.
  (a) **Classes from the Graph-Based Ontology Integration**
  Related classes from different data sets are extracted by analyzing the *SameAs* graph patterns and grouped into $cgroup_1$, $cgroup_2$, ..., $cgroup_z$. For each group, we define *ex-onto:ClassTerm*, where the $ClassTerm$ is the most frequent term in the group. For all $c_i \in cgroup_k$, $<$ *ex-onto:ClassTerm*$_k$, *ex-prop:hasMemberClasses*, $c_i >$ is added automatically.
  (b) **Classes from the Machine-Learning-Based Approach**
  Top-level classes in each data set are added to the integrated ontology. If a top-level class $c_i \notin cgroup_k (1 \leq k \leq z)$, we create a new group $cgroup_{z+1}$ for each class c_i and create a new term *ex-onto:ClassTerm*$_{z+1}$ for the new group. Then we add $<$ *ex-onto:ClassTerm*$_{z+1}$, *ex-prop:hasMemberClasses*, $c_i >$ to the integrated ontology.
- **Property**
The extracted properties from two components are merged according to the following rules. At first, we extract the existing property type and domain information of each property from the data sets. The property type is mainly defined using the rdf:Property, owl:DataTypeProperty, and the object property owl:ObjectProperty. If the type is not clearly defined, we set the type as rdf:Property.
  (a) **Properties from the Graph-Based Ontology Integration**
  Related properties from various data sets are extracted by analyzing the

SameAs graph patterns and grouped into $pgroup_1$, $pgroup_2$, ..., $pgroup_p$. For each group, we choose the most frequent term *ex-onto:propTerm*. Then, for each property $prop_i \in pgroup_t$ $(1 \leq t \leq p)$, we add the triple < *ex-onto:propTerm*$_t$, *ex-prop:hasMemberProperties*, $prop_i$ > and the triple < *ex-onto:propTerm*$_t$, *rdfs:domain*, dInfo>, where dInfo is the retrieved domain information of the $prop_i$ in the ontology enrichment process.

(b) **Properties from the Machine-Learning-Based Approach**
We automatically add domain information for the properties retrieved using the Apriori method. For each property *prop* extracted from the instances of class *c*, < *prop*, *rdfs:domain*, *c* > is automatically added, if it's not defined in the data set.

3. **Naming Validator**
The naming validator corrects the terms, which have different naming format with others. In the pitfall catalogue introduced in the OOPS! (OntOlogy Pitfall Scanner!) system [105], they suggested using consistent naming criteria for validating the ontology quality. In our regulation, we don't allow any special character in the terms, such as "–", "_", and "/", etc. We remove all the special characters and modify as a combination of words starts with a capitalized letter.

4.3.3 Experimental Evaluation

In this experiment, we evaluate the integrated ontology with an ontology validator to check if we successfully added missing domain, range information and annotations. We validated with the OWL validator [106] developed by the University of Manchester. However, this OWL validator only validates if the ontology is in the OWL 2 profile or not, and it cannot detect if there is missing information in the ontology. Hence, we use the OOPS! (OntOlogy Pitfall Scanner!) to validate the integrated ontology.

1. **Evaluation with OOPS! Validator**
The OOPS! (OntOlogy Pitfall Scanner!) analyze whether an ontology contains anomalies or pitfalls [105]. Currently they use 29 pitfalls from four dimensions such as human understanding, logical consistency, modeling issues, and real world representation. We validated the integrated ontology constructed without and with the integrated ontology constructor using the OOPS! validator.

The OOPS! detected 55 missing ranges, 9 missing domains, and different naming criteria pitfalls from the integrated ontology constructed without this integrated ontology constructor. However, we didn't detect any of these pitfalls from the integrated ontology constructed with this integrated ontology constructor, because we automatically added ranges and domains, and corrected naming mistakes in the integrated ontology constructor component. In fact, we found 26 missing domains and automatically added the domain info by analyzing the object values, which means the OOPS! did not detect all the missing domains of the properties.

With the integrated ontology constructor, we successfully removed the pitfalls caused by missing domain, missing range, and inconsistent naming criteria. The domain information can help users understand the relations between properties and classes, and the range information can help us easily validate or add values for properties.

4.4 Summary

In this chapter, we mainly focus on solving two problems: the difficulty in identifying core ontology schemas and missing domain or range information problem. We applied machine learning methods such as Decision Table and Apriori to retrieve core ontology schemas to complement the integrated ontology constructed from graph-based ontology integration. Furthermore, we added missing domain and range information automatically by analyzing samples of the instances in the data sets.

Experimental results show that by applying machine learning methods, we extracted core ontology schemas that are important for describing instances. Furthermore, we successfully added missing domain, range, and annotations to make the integrated ontology easily understandable.

Chapter 5

Framework for InTegrating ONtologies - FITON

The Linked Open Data (LOD) cloud contains tremendous amounts of interlinked instances, from where we can retrieve abundant knowledge. However, because of the heterogeneous and big ontologies, it is time-consuming to learn all the ontologies manually and it is difficult to observe which properties are important for describing instances of a specific class. In order to construct an ontology that can help users easily access to various data sets, we propose a semi-automatic **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) that can solve the proposed three main problems: ontology heterogeneity problem, difficulty in identifying core ontology schemas, and missing domain or range information problem [107].

We start with the motivation of our research in Section 5.1. Then in Section 5.2, we introduce **FITON** that consists of three main components: graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. Experimental results in Section 5.3 show that **FITON** successfully integrates heterogeneous ontologies and creates high quality integrated ontology. In Section 5.4, we discuss some possible applications with the integrated ontology and summarize this chapter in Section 5.5.

5.1 Motivation

The data sets in the LOD cloud are published according to the Linked Data principles and also provide links to other data resources [2]. However, there is no standard ontology for all the data sets, but all kinds of ontologies that cause the ontology heterogeneity problem. In [19], they categorized the ontology heterogeneity problem into four different types: syntactic heterogeneity, terminological heterogeneity, conceptual heterogeneity, and semiotic heterogeneity. We mainly focus on the terminological and conceptual heterogeneity problems. The terminological heterogeneity problem occurs when the same entities in different ontologies are represented differently. The conceptual heterogeneity is also called semantic heterogeneity in [20] and logical mismatch in [21], which occurs due to the use of different axioms for defining concepts or due to the use of totally different concepts. For instance, FIGURE 1.2 shows the interlinked instances of "France", where all the properties (labeled on the dotted line) connected with the grey boxes (objects) represent the name of "France", and the properties connected to the black boxes represent the population.

In previous Chapter 3 and 4, we described different approaches to solve the proposed three main problems in accessing to the Linked Data. In order to solve the ontology heterogeneity problem, we proposed the graph-based ontology integration, which is an extension of the Mid-Ontology learning approach. The difficulty in identifying core ontology is resolved by machine learning approaches such as Decision Table and Apriori that retrieve top-level classes and frequently used core properties. The third problem is that there is a lot of missing domain or range information in the real data sets, which is solved using the integrated ontology constructor. By combining above three approaches, we propose the **F**ramework for **InT**egrating **ON**tologies (**FITON**), which decreases the ontology heterogeneity in the linked data sets, retrieves core ontology schemas, and automatically enriches the integrated ontology by adding missing domain, range, and default annotations.

5.2 FITON

In this section, we introduce the **F**ramework for **InT**egrating **ON**tologies (**FITON**) that can automatically find alignments between different ontologies and extract core ontology schemas that are used for describing instances. In addition, we enrich the integrated ontology with domains, ranges, and annotations information that can help users easily

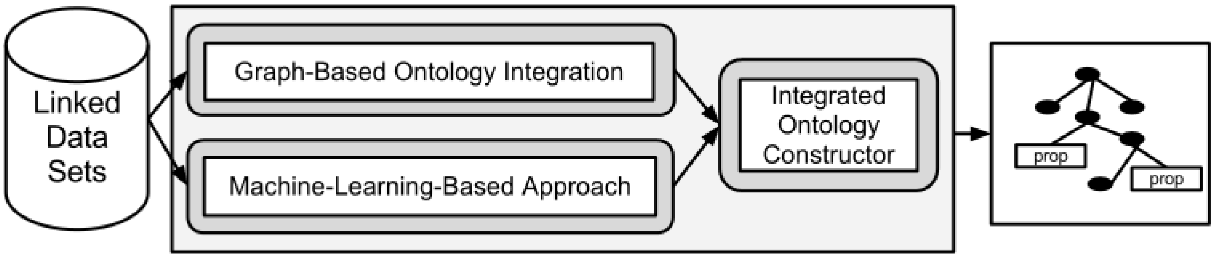

FIGURE 5.1: Semi-automatic **F**ramework for **InT**egrating **ON**tologies (FITON).

understand the ontology. The framework of semi-automatic **FITON** is shown in FIGURE 5.1, which consists of three main components: graph-based ontology integration (Chapter 3), machine-learning-based approach (Chapter 4), and integrated ontology constructor (Chapter 4) that can automatically enrich the integrated ontology with rich information.

FITON applied different techniques to solve the following three problems during integrating ontologies of various data sets in the LOD cloud:

1. **Ontology Heterogeneity Problem**
   In order to access to various data sets simultaneously, we have to understand their heterogeneous ontologies in advance to achieve semantic interoperability. Ontology integration is defined as the process that generates a single ontology from different existing ontologies [22]. However, there are only few links at a class or property level that makes it difficult to directly retrieve equivalent classes and properties for the ontology integration. Ontology alignment, or ontology matching is commonly used to find correspondences between ontologies to solve the ontology heterogeneity problem [23]. We combined the string-based and knowledge-based ontology matching methods on the predicates and objects to discover similar concepts.

   Since the same instances are linked by *owl:sameAs*, we can create undirected graphs with the linked instances and analyze the graphs to retrieve related classes and properties. By analyzing the graph patterns, we can observe how the same classes and properties are represented differently in various data sets. The detailed graph-based ontology integration approach is introduced in Chapter 3, Section 3.3.

2. **Difficulty in Identifying Core Ontology Schemas**
   Other than retrieving related classes and properties, we also need frequently used core classes and properties of the ontologies to construct a high-quality integrated ontology. This can also help the Semantic Web developers easily identify the core ontology schemas for describing instances in each data. The core ontology schemas

can assist us to construct SPARQL queries and to add missing core information in the data sets.

Machine learning methods such as association rule learning and rule-based classification can be applied to discover core properties for describing instances in a specific class. Apriori is a well-known algorithm for learning association rules in a big database [24], while the rule-based learning method - Decision Table can retrieve a subset of properties that leads to high prediction accuracy with cross-validation [25]. By applying machine learning approaches to the linked data sets, we can retrieve core ontology schemas that are important for describing instances in the data sets. The detailed machine-learning-based approach is described in Chapter 4, Section 4.2.

3. **Missing Domain or Range Information**
The relations between the ontology classes and properties are described with the property rdfs:domain, which indicates that the properties are designed to be used for describing the instances of a specific class. Furthermore, the range information of the values can help users better understand the data sets. Besides the domain and range information, we should also retrieve the description of each ontology schema to construct an easily understandable integrated ontology.

   We can randomly select some samples of the instances that contain the retrieved classes and properties and analyze the usage of the domain information and values. From the contents of the sample instances, we automatically retrieve the domain information to link the properties with the classes. We can also retrieve the default range information of the properties or analyze the values of properties from the sample instances, which are mainly categorized into String and Resource. By analyzing samples of the instances, we can reduce the analysis time and also be able to retrieve the domain and range information. The default annotations make the integrated ontology easily understandable. The detailed integrated ontology constructor is described in Chapter 4, Section 4.3.

5.3 Experimental Evaluation

We selected the same data sets as used in previous Chapters to evaluate **FITON** framework. At first, we discuss the comparison results with other ontology matching tools using ontology reference alignments. Then, we evaluate the quality of the integrated ontology with the ontology reference alignments. The class hierarchy of the integrated ontology is shown in Appendix A and the the integrated ontology constructed with FITON is shown in Appendix B.

5.3.1 Comparison with Other Ontology Matching Tools

Many state-of-the-art ontology matching tools have been developed, but most of them accept only two ontologies as inputs. Furthermore, since they need ontologies for analysis, they could not find alignments for the data sets which do not contain structured ontologies such as NYTimes and LinkedMDB. Hence, we only compare the alignments between DBpedia and Geonames with other ontology matching tools. The following two comparison experiments are conducted using DB-Geo alignments and BLOOMS alignments[1].

1. **Comparison with DB-Geo Alignments**
   Since there is no standard benchmark for the LOD schema alignments, an expert manually created the DB-Geo alignments between DBpedia and Geonames as shown in Table 5.1. The DB-Geo alignments contain 49 reference alignments such as (db-onto:totalPopulation and geo-onto:population), (db-onto:location and geo-onto:location), (db-onto:Airport and geo-onto:S.AIRP), and (db-onto:Mountain and geo-onto:T.MT), etc. We compare **FITON** with AROMA [14] ontology matching system that can find alignments between DBpedia and Geonames.

   The AROMA system uses the association rule mining on the data for matching ontologies [14]. AROMA can find 11 alignments between DBpedia and Geonames, where only 2 are correct alignments. As shown in Table 5.2, the precision, recall and F-measure of the AROMA system is 0.18, 0.04, and 0.07, respectively. **FITON** finds alignments with 0.64 precision, 0.37 recall, and 0.47 F-measure. We found 28 alignments in total, where 18 of them are correctly matched. The experimental results show that **FITON** performs much better than the AROMA system.

   In some cases, we found correct matchings, but some of them are categorized as incorrect mappings according to the manual alignments because of the misuses of the schemas in the real data. For instance, in the reference alignments,

[1] http://wiki.knoesis.org/index.php/BLOOMS

Table 5.1: DB-Geo Ontology Alignments.

Geonames	DBpedia
SpatialThing	Place
T.MTS	MountainRange
S.BLDG	Building
S.LTHSE	Lighthouse
S.BDG,S.BDGQ	Bridge
S.HSTS	HistoricBuilding
S.MALL	ShoppingMall
S.REST	Restaurant
S.HTL	Hotel
S.MUS	Museum
S.STDM	Stadium
S.HSP	Hospital
S.AIRP	Airport
S.HSTS	HistoricPlace
S.MNMT	Monument
T.PASS	MountainPass
L.RES	ProtectedArea
T.VAL	Valley
L.PRK	Park
P.PPL	PopulatedPlace
A.ADMD, A.ADMD1-4	AdministrativeRegion
L.CONT	Continent
T.ATOL	Atoll
T.ISL	Island
A	Country
P	Settlement
P.PPL	Village
P.PPL	City
P.PPL	Town
S.BUSTN	Station
S.MTRO	Station
S.RSTN	Station
S.CAVE	Cave
T.MT	Mountain
H	BodyOfWater
H.LK	Lake
H.STM	Stream
S.SCHC	Organisation, EducationalInstitution, College
S.SCH	Organisation, EducationalInstitution, School
S.UNIV	Organisation, EducationalInstitution, University
S.STNR	Organisation, RadioStation
R, R.RD	Infrastructure, Road
S.PS	Infrastructure, PowerStation
skos:prefLabel	title
shortName	abbreviation
population	totalPopulation
postalCode	postalCode
locatedIn	location
wikipediaArticle	wikiPageRedirects

TABLE 5.2: Comparison with DB-Geo alignment references

DBpedia-Geonames	AROMA	**FITON**
Precision	0.18	0.64
Recall	0.04	0.37
F-measure	0.07	0.47

db-onto:totalPopulation is matched to geo-onto:population, but the property db-onto:totalPopulation never appears in the DBpedia data set. In fact, the population is described with other properties such as db-onto:populationTotal, db-prop:populationEstimate, and db-prop:population, etc. **FITON** successfully grouped these properties with geo-onto:population.

2. **Comparison with BLOOMS Alignments**

The BLOOMS ontology matching approach utilizes the Wikipedia category hierarchy and constructs BLOOMS forests to find alignments between two ontologies [18]. They provided 48 reference alignments between DBpedia and Geonames for evaluation. They matched the wgs84_pos:SpatialThing to most of the DBpedia ontology classes, which means that they matched all the geographic information in the DBpedia ontology classes to the Geonames class wgs84_pos:SpatialThing. However, in DB-Geo alignments, we matched them with more specific geographic feature codes. Hence, in order to compare the alignment results with BLOOMS and other systems with the BLOOMS alignments, we assume all the feature codes of Geonames are subclasses of the wgs84_pos:SpatialThing.

According to [18], only S-Match [15] can find ontology alignments between DBpedia and Geonames, while RiMOM [16] causes errors and all the other systems including BLOOMS failed to find matching ontology schemas. The failure in the ontology alignment is caused by some ontologies that have ambiguous meaning of the concepts or the absence of corresponding concepts in the target dataset. However, **FITON** can find alignments for the poorly structured datasets by analyzing the contents of the interlinked instances. In Table 5.3, we listed the results of BLOOMS, RiMOM, and S-Match from the evaluation results in the paper [18]. The precision of **FITON** is 0.65, which is about 3 times of the precision of the S-Match: means 65% of the found alignments are correct with **FITON**. However, the recall is lower than S-Match, where S-Match can find all the alignments that are used in the evaluation. The F-measure of **FITON** and S-Match is the same with 0.37.

By adjusting DB-Geo alignments with the alignments provided by BLOOMS, we compared **FITON** with BLOOMS, RiMOM, and S-Match ontology matching tools.

TABLE 5.3: Comparison with BLOOMS alignments.

Geonames-DBpedia	Alignments from BLOOMS			
	FITON	BLOOMS	RiMOM	S-Match
Precision	0.65	0	err	0.23
Recall	0.26	0	err	1
F-measure	0.37	N/A	N/A	0.37

Since, both BLOOMS and RiMOM fail to find alignments between DBpedia and Geonames, we compared **FITON** with S-Match only. As a result, **FITON** can find alignments with a high precision, while S-Match can find more alignments with a low precision. Although we cannot find all the alignments, we can find accurate alignments with **FITON**.

5.3.2 Evaluation of the Integrated Ontology

The graph-based ontology integration framework introduced in Chapter 3, Section 3.3 only focuses on the related classes and properties acquisition, that may miss some core classes and properties. Hence, we applied machine learning methods such as Decision Table and Apriori to find out the core properties for describing instances of a specific class. The final integrated ontology contains 135 classes and 450 properties that are grouped into 87 and 97 groups, respectively. In this section, we evaluate the quality of the integrated ontology using the ontology alignments created by ontology experts. Furthermore, we discuss the comparison of the ontologies among the original data, the integrated ontology (IntOnto) from **FITON**, and the final revised integrated ontology.

1. **Comparison with Ontology Alignments**

   The quality of the integrated ontology is evaluated with the ontology alignments created by the experts who are familiar with the LOD data sets. They created alignments among DBpedia, Geonames, LinkedMDB, and NYTimes.

   As shown in TABLE 5.4, the precision reaches 1 for the alignments of DBpedia-LinkedMDB, LinkedMDB-NYTimes, and Geonames-NYTimes. For the DBpedia-Geonames and DBpedia-NYTimes we found some incorrect alignments, but we could not get any alignment for the LinkedMDB-Geonames. The system can perform best to find the alignments between DBpedia and Geonames. The main reason is that most of the links are between DBpedia and Geonames as shown in FIGURE

TABLE 5.4: Evaluation of the integrated ontology.

Data Pair	Precision	Recall	F-measure
DBpedia-Geonames	0.64	0.37	0.47
DBpedia-LinkedMDB	1	0.1	0.2
DBpedia-NYTimes	0.93	0.02	0.04
LinkedMDB-NYTimes	1	0.07	0.13
LinkedMDB-Geonames	0	0	n/a
Geonames-NYTimes	1	0.04	0.08

3.2, while there are only 247 links between LinkedMDB and Geonames which are too less to find correct alignments.

Furthermore, since we only analyze the interlinked instances for finding alignments, we can not find alignments if there are only few links between the instances. Therefore, the recall is low between most of the dataset pairs. Although, it cannot find some alignments that experts created, they can find some alignments that the experts can not discover. For instance, nyt:nytd_geo, mdb-movie:country, geo-onto:A.PCLI, geo-onto:A.PCLD, and db-onto:Country are integrated with our approach. However, the experts can not discover that geo-onto:A.PCLI and geo-onto:A.PCLD are used to represent countries in Geonames, because there is no annotation for these two feature codes.

The time cost for manual revision on the created integrated ontology is only few hours by the experts. Finding alignments for the data sets is time-consuming, especially for the DBpedia that has hundreds of classes and thousands of properties. It is even impossible for the experts to find alignments for the heterogeneous ontologies manually. Our approach can dramatically decrease the time cost for finding the ontology alignments.

2. **Comparison between the Original Data and the Integrated Ontology**

The integrated ontology consists of parts of the ontology schemas in the original data. Here, we compare the ontology classes and properties among the original data, the integrated ontology (IntOnto) created using **FITON**, and the ontology created after manual revision.

Table 5.5 shows the number of classes and properties in the ontology of each data set, that are included in the original data, integrated ontology from **FITON** , and the final integrated ontology after manual revision. The last two rows show the total number of classes and properties from the above four data sets. Here, we

TABLE 5.5: Comparison between the Original Data and the Integrated Ontology.

Data Set	Schema	Original Data	**FITON** IntOnto	Revised IntOnto
DBpedia	Class	237	84	88
	Property	1366	383	383
Geonames	Class	428	36	43
	Property	20	16	16
NYTimes	Class	4	4	4
	Property	3	3	3
LinkedMDB	Class	51	11	13
	Property	98	27	27
Total	Class	720	135	148
	Property	1487	429	429

didn't count some properties such as SKOS and FOAF, that are used in the data sets for describing instances.

In the following, we will discuss pairwise comparison on the ontology classes and properties according to Table 5.5.

(a) Original Data and **FITON** IntOnto
The integrated ontology from **FITON** contains partial of the total ontology schemas in the original data. The total number of classes and properties in **FITON** are 135 and 429, respectively, which are 19% and 29% of the original data. Among the integrated ontology of **FITON** , 62% of the classes and 89% of the properties are dominated by DBpedia ontology classes and properties. Since NYTimes contains only 4 classes and 3 properties that are commonly used in each NYTimes instance, we retrieved all these NYTimes classes and properties. For Geonames, we retrieved 8% (36) of the total classes and 80% of the properties in the original data. From LinkedMDB, we retrieved 11 classes and 27 properties, that are 21.6% and 27.5% of the ontology classes and properties of LinkedMDB in the original data.

(b) **FITON** IntOnto and Revised IntOnto
The last column lists the number of classes and properties in the final revised integrated ontology, which is a revised version of the **FITON** IntOnto. The number of classes in DBpedia, Geonames, and LinkedMDB are increased because some incorrectly grouped classes are divided into different groups or some ontology classes are added by manual alignments. The total number of classes in **FITON** and the revised IntOnto are 135 and 148, respectively. The classes in the **FITON** IntOnto are grouped into 87 groups and the classes in

the revised IntOnto are grouped into 86 groups. The number of properties are not changed because it's too difficult to check the whole property alignments manually. However, the 429 properties are grouped into 97 and 102 groups in **FITON** IntOnto and revised IntOnto, respectively. This indicates that the wrong groups of properties from **FITON** are divided into different groups after manual revision.

(c) Original Data and Revised IntOnto
Comparing the final revised IntOnto with the ontology schemas in the original data, we found that only 21% of the classes and 29% of the properties are integrated.

The integrated ontology contains only partial of the ontology classes and properties in the original data sets, which is easier for the users to understand. Furthermore, because of the heterogeneity of the ontology schemas in DBpedia, approximately 60% of the classes and 90% of the properties in the revised integrated ontology are from DBpedia. The results in Table 5.5 imply that **FITON** can reduce heterogeneity of the ontologies and can create an integrated ontology with only partial of the whole ontology schemas, which is easier to be understood than the original ontology schemas.

5.4 Discussion

In this section, we first discuss the performance of each component of FITON on different types of data sets. Then we discuss possible applications with the graph patterns extracted using **FITON** and with the integrated ontology. The integrated ontology consists of different ontology schemas from various linked data sets. Hence, we can access to various data sets with the integrated ontology classes and properties to integrate information from different data sets. We can find missing SameAs links either with the graph patterns or with the integrated ontology. Furthermore, we can retrieve more query results with the integrated ontology than using one single ontology schema. In the following, we will discuss in details with examples.

1. **Performance of FITON**
FITON can integrate ontologies of the data sets from various domains. It consists of

three main components such as graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. In the following, we will discuss the performance of each component on different types of data sets.

- **Graph-Based Ontology Integration**
  As the experimental results shown in Section 3.3.3, this component performs well when there are many SameAs links between different data sets. Furthermore, for the data sets with heterogeneous ontology itself such as DBpedia, this component can effectively reduce the ontology heterogeneity by applying ontology matching methods.

  This approach mainly focuses on solving the terminological heterogeneity problem. However, it also helps solving partial of the conceptual heterogeneity problem using discovered alignments between concepts. This solution performs well on the data sets with many links among the instances. Moreover, we only consider 1:1 mappings and do not consider blank nodes.

- **Machine-Learning-Based Approach**
  We discussed the experimental results of Decision Table and Apriori in Section 4.2.3. From the experimental results, we can conclude that this component can perform well on identifying core ontology schemas when the data sets contain various categories of instances and different types of instances, or use distinct properties for describing the instances. For the Geonames and NYTimes data sets, the performance is not good because they both use commonly used properties to describe instances.

- **Integrated Ontology Constructor**
  This component automatically adds missing domain or range information by analyzing samples of the instances. Hence, it performs well when there are enough samples of the instances that utilize the ontology schemas.

FIGURE 5.2 and FIGURE 5.3 show the examples of Film and Species in the integrated ontology. As these two figures show, the integrated ontology integrated related ontology schemas and also successfully identified core ontology schemas using machine learning approach. Furthermore, the missing domain and range information is also automatically added, that makes the integrated ontology easily understandable.

2. **Discover Missing Links with Integrated Ontology**
   Here, we show two examples as shown in TABLE 5.6 and TABLE 5.7, which can

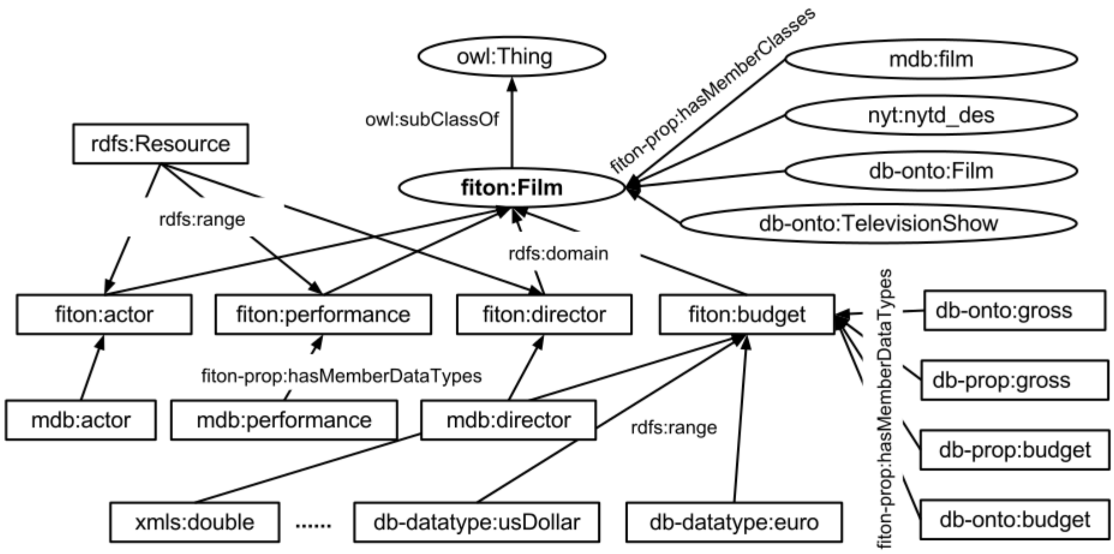

Figure 5.2: Example of Film in the Integrated Ontology.

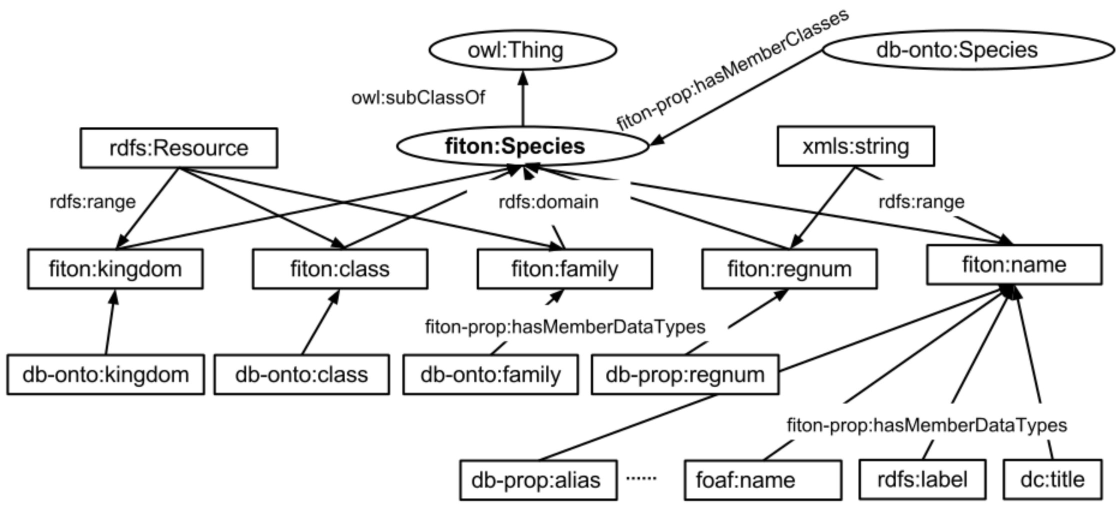

Figure 5.3: Example of Species in the Integrated Ontology.

Table 5.6: Find missing links of islands.

Example 1: Link Islands
SELECT DISTINCT ?geo ?db ?string
where { ?geo geo-onto:featureCode geo-onto:T.ISL.
?geo ?gname ?string.
ex-onto:name ex-prop:hasMemberDataTypes ?gname.
?db rdf:type db-onto:Island.
ex-onto:name ex-prop:hasMemberDataTypes ?dname.
?db ?dname ?string. }

TABLE 5.7: Find missing links of countries.

Example 2: Link Countries
SELECT DISTINCT ?geo ?db
where { ex-onto:name ex-prop:hasMemberDataTypes ?gname.
{ ?geo geo-onto:featureCode go-onto:A.PCLI. } UNION
{ ?geo geo-onto:featureCode geo-onto:A.PCLD. }
?geo ?gname ?string.
?db rdf:type db-onto:Country.
ex-onto:name ex-prop:hasMemberDataTypes ?dname.
?db ?dname ?string. }

find missing SameAs links with the integrated ontology. The first example is to find the missing links for the island instances between DBpedia and Geonames. The db-onto:Island and geo-onto:T.ISL are used for island instances in DBpedia and Geonames, respectively. The Example 1 in TABLE 5.6 shows a SPARQL query to find the same islands that has the same name in DBpedia and Geonames. In total, we retrieved 509 links, including 218 existing SameAs links. Among the existing links, 97 links are from DBpedia to Geonames, 211 links are from Geonames to DBpedia, and 90 bidirectional links between DBpedia and Geonames. Hence, we discovered 291 missing links which links the same islands.

The second example is to find missing links for country instances. The class db-onto:Country is integrated with geo-onto:A.PCLI and geo-onto:A.PCLD. With the SPARQL query Example 2 in TABLE 5.7, we retrieved 663 links, including 221 existing SameAs links. Among the existing links, 30 are from DBpedia to Geonames, 220 are from Geonames to DBpedia, and 29 links are bidirectional links. As a result, we discovered 442 new links for countries with exact matching on the names.

The above SPARQL examples show that we can find missing links with the integrated ontology using exact matching on the labels of the instances. If we can analyze the labels with similarity string matching, we could discover more links with the integrated ontology. However, currently the SPARQL endpoint does not support similarity matching query.

3. **More Answers with the Integrated Ontology**
   The heterogeneous ontologies make it difficult to find proper properties to construct a SPARQL query. However, the integrated ontology grouped all kinds of properties that are used in the real data sets, including properties that are not defined in the ontology. Hence, we can discover more query results by simply using the classes and properties in the integrated ontology. Here, we show two SPARQL examples as

shown in TABLE 5.8. We selected two questions from the QALD-1 Open Challenge [2].

The first query is to find the cities that have a population more than 10 million. The SPARQL query on the left side is given by the QALD-1 Open Challenge, which utilizes the db-onto:City class and the standard property db-prop:populationTotal. The query on the right side uses all the population properties used in the real DBpedia. As a result, we retrieved 20 distinct cities with our SPARQL query, while we only retrieved 9 cities with the standard query.

The second query is to answer the height of Claudia Schiffer. With the standard property db-onto:height, we can retrieve the answer of 1.8(m). However, with our integrated ontology, which consists of db-onto:height, db-prop:height, db-prop:heightIn, db-prop:heightFt, and db-onto:Person/height, we found one more answer - 180.0 (centimetre) in a different measurement.

Since we integrated properties that are not defined in the ontologies, but commonly used in the published data sets, we can retrieve more answers with the integrated ontology. Hence, it is helpful for the QA systems for discovering more related answers with simple queries.

[2] http://www.sc.cit-ec.uni-bielefeld.de/qald-1

TABLE 5.8: SPARQL examples from QALD-1 Open Challenge.

Give me all the cities with more than 10,000,000 inhabitants.	
Standard Query	Query with the Integrated Ontology
SELECT DISTINCT ?uri ?string WHERE { ?uri rdf:type db-onto:City. ?uri db-prop:populationTotal ?inhabitants . FILTER (?inhabitants > 10000000). OPTIONAL { ?uri rdfs:label ?string. FILTER (lang(?string) = 'en') }}	SELECT DISTINCT ?uri ?string WHERE { ?uri rdf:type db-onto:City. ex-onto:population ex-prop:hasMemberDataTypes ?prop. ?uri ?prop ?inhabitants. FILTER (?inhabitants > 10000000). OPTIONAL { ?uri rdfs:label ?string. FILTER (lang(?string) = 'en') }}
How tall is Claudia Schiffer?	
Standard Query	Query with the Integrated Ontology

SELECT DISTINCT ?height
WHERE {
res:Claudia_Schiffer db-onto:height ?height.
}
	SELECT DISTINCT ?height
	WHERE {
	ex-onto:height ex-prop:hasMemberDataTypes ?hprop.
	res:Claudia_Schiffer ?hprop ?height. }

5.5 Summary

The Semantic Web developers may want to integrate data sets from various domains, but it is time-consuming to manually learn all the ontologies from different data sets. Moreover, because of the big and heterogeneous ontologies, it is infeasible to manually map ontologies. Constructing a global ontology by integrating heterogeneous ontologies of the linked data can help effectively integrate various data resources.

We proposed the **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) that consists of three main components: graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. The graph-based ontology integration component retrieves related ontology classes and properties by analyzing the graph patterns of the interlinked instances, which reduces the heterogeneity of the ontologies in the LOD cloud. The integrated ontology contains top-level classes and frequent core properties retrieved from the machine-learning-based approach, which can help Semantic Web application developers easily find core properties used for a specific instance. We also enriched the integrated ontology and validated it during the ontology merging process, which creates a high-quality and easily understandable ontology. With the integrated ontology, we can also detect misuses of ontologies in the data sets and can recommend core properties for describing instances. Furthermore, we can also detect missing links using the integrated ontology or use it for QA systems to retrieve more query results.

Chapter 6

Conclusion

In this chapter, we discuss the advantages of our approach by comparing with related work and also discuss remaining problems in Section 6.1. Then we conclude our research work in Section 6.2 that integrates heterogeneous ontologies by analyzing graph patterns of the interlinked instances, identifies core ontology schemas by applying machine learning methods, and adds missing domain or range information using the integrated ontology constructor. In Section 6.3, we propose future work to achieve our vision and propose possible further research by using the integrated ontology constructed with **FITON**.

6.1 Discussion

The **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) is designed for integrating heterogeneous LOD ontologies. **FITON** consists of three main components: graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor. FITON solves three main problems and each component is designed to resolve each problem. The graph-based ontology integration solves the ontology heterogeneity problem that is one of the most challenging problems in dealing with the Linked Open Data. The machine-learning-based approach is designed to solve the problem of difficulty in identifying core ontology schemas. The last component is the integrated ontology constructor that enriches the integrated ontology by adding missing domain and range information.

Unlike the research work of [48] and [50] that are limited to specific domains, **FITON** can integrate ontologies from various domains. Furthermore, mapping ontology properties is useful to find related instances from different data sets by querying with SPARQL. However, some researchers mainly focus on mapping ontology classes only [48]. Our approach retrieves both related ontology classes and properties that give us a clear view of the interlinked instances in different data sets, and makes it easier to discover missing links among instances.

Some researchers also applied machine learning methods for mapping ontologies [51], but only worked with expressive lightweight ontologies. Therefore, the method cannot be applied to real data sets which lack of expressive axioms or constructed with large ontologies. In contrast, **FITON** can find alignments for large ontologies such as DBpedia. Since we analyze the contents of the interlinked instances, we can find alignments for rudimentary ontologies that lack of expressive axioms such as NYTimes and Geonames.

There are also some papers which introduced extracting key features from the entities of the graph patterns to detect hidden *owl:sameAs* links or relations [84]. However, there is no discussion about what kind of features are important for finding the hidden links. **FITON** retrieved core ontology classes and properties that are commonly used for describing instances in each data set. Furthermore, by combining them with the related ontology schemas, we can find missing links using simple SPARQL queries. The integrated ontology constructed with **FITON** can be used for link discovery, or for correcting mistakes in the real data sets.

In case of adding a new data set for ontology integration, we have to analyze additional graph patterns that contain the instances of the new data set to retrieve new groups of related ontology schemas using the graph-based ontology integration and then merge with the existing integrated ontology. The machine-learning-based approach is performed independently on each data set, therefore we can easily extend the existing integrated ontology with additional core ontology schemas retrieved from the new data set. However, if we only use one data set for **FITON**, we can only retrieve core ontology schemas, but cannot extract related ontology schemas using graph-based ontology integration because there is no graph patterns to analyze.

Although, we have solved some fatal problems of ontology integration for the linked data sets, there are still some remaining problems. For example, some related ontology classes and properties cannot be identified if the instances that contain them are not connected in the LOD cloud. Because our graph-based ontology integration component highly depends on the linked instances, we can not collect data from the instances that are not connected. One possible solution is to manually assign links for each type of instances as samples for further analysis. Furthermore, if we add more big data sets, the processing time will increase dramatically and also the machine learning tool - WEKA performs slowly on big data sets, especially with large number of dimensions (properties). A scalable technology should be applied to solve this problem.

6.2 Conclusion

In this thesis, we proposed the **F**ramework for **I**n**T**egrating **ON**tologies (**FITON**) that can integrate ontologies from various LOD data sets. The **FITON** solves three main problems such as the ontology heterogeneity problem, difficulty in identifying core ontology schemas, and missing domain or range information problem. Each problem is resolved by the three main components of FITON: graph-based ontology integration, machine-learning-based approach, and integrated ontology constructor, respectively. The graph-based ontology integration component retrieves related ontology classes and properties by applying ontology similarity matching methods (string-based and WordNet-based ontology matching) on the graph patterns of the interlinked instances, which reduces the heterogeneity of ontologies in the LOD cloud. We improved the Mid-Ontology Learning Approach and applied it to the graph patterns for discovering related ontology classes and properties. We also applied machine-learning-based approaches such as Decision Table and Apriori to extract top-level classes and frequently used core properties, which can help Semantic Web application developers easily find core properties used for a specific instance. Furthermore, we automatically enriched the integrated ontology and validated it during the ontology merging process, which creates a high-quality and easily understandable integrated ontology.

Experimental results show that we successfully discovered core classes and frequently used core properties in each data set that can help data publishers detect misuses of ontologies in the published data sets. Our method is domain-independent and can perform on data sets from various domains. Furthermore, we can also integrate not well designed ontologies such as the ontologies of Geonames and NYTimes. In addition, for the instances of a specific class, we can recommend core properties that are frequently used for the instance description. Although we need minor manual revision on the automatically created integrated ontology, the ontology integration method successfully retrieves related ontology classes and properties that are critical for interlinking related instances.

The vision of our research is to integrate ontologies from various LOD data sets that can improve the interoperability among different data sets and reduce ontology heterogeneity problem. However, because of lack of the links among the available data sets in the LOD cloud and limited number of instances in some specific classes, we cannot achieve full ontology integration. The goal of our research is to solve the three proposed problems in order to construct a robust integrated ontology for some parts of the linked data sets from various domains. In this thesis, we introduced **FITON** that can reduce the ontology heterogeneity by applying graph-based ontology integration method and utilize machine learning methods to identify core ontology schemas in the data sets. Moreover, it automatically adds missing domain and range information to the integrated ontology to make it easier for the users to understand the ontology schemas. Besides the limitations of the data sets, there are still some issues remain to be solved such as the scalability problem when we deal with many big data sets.

6.3 Future Work

FITON can integrate ontologies for the linked data sets from various domains. However, we only used four data sets from three different domains for evaluating FITON. In order to achieve our vision, which is to integrate ontologies from various LOD data sets, we have to challenge the scalability problem. To perform on more big data sets, we should improve the scalability of FITON to efficiently integrate ontologies for the data sets from various domains.

Furthermore, I would like to combine **FITON** with link discovery system to improve the quality of the integrated ontology. By recursively feeding discovered missing links to the data sets, we can also update the integrated ontology with newly linked instances. Moreover, by utilizing the integrated ontology created from **FITON**, we can link instances from various data sets and improve the data interoperability. From the aspect of link discovery system, we can also retrieve more missing links by using the updated integrated ontology and by applying similarity matching methods on the object values in the link discovery process.

Bibliography

[1] Tim Berners-Lee. *Linked Data - Design Issues*, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[2] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[3] Graham Klyne and Jeremy J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation, 2004. http://www.w3.org/TR/rdf-concepts/.

[4] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. *OWL Web Ontology Language Reference*. W3C Recommendation, 2004. http://www.w3.org/TR/owl-ref/.

[5] Dan Brickley and R.V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation, 2004. http://www.w3.org/TR/rdf-schema/.

[6] W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview*. W3C Recommendation, 2012. http://www.w3.org/TR/owl2-overview/.

[7] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.

[8] E. Prud'hommeaux and A. Seaborne. *SPARQL query language for RDF*. W3C Recommendation, 2008. http://www.w3.org/TR/rdf-sparql-query/.

[9] Steve Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. W3C Recommendation, 2013. http://www.w3.org/TR/sparql11-query/.

[10] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: Live views on the web of data. *Journal of Web Semantics*, 8(4):355–364, 2010.

[11] Tim Finin, Yun Peng, R. Scott, Cost Joel, Sachs Anupam Joshi, Pavan Reddivari, Rong Pan, Vishal Doshi, and Li Ding. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pages 652–659. ACM Press, 2004.

[12] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media meets semantic web — how the bbc uses dbpedia and linked data to make connections. In *Proceedings of the 6th European Semantic Web Conference*, pages 723–737. Springer Berlin Heidelberg, 2009.

[13] Sören Auer and Jens Lehmann. Creating knowledge out of interlinked data. *Semantic Web Journal*, 1(1-2):97–104, 2010.

[14] Jérôme David, Fabrice Guillet, and Henri Briand. Matching directories and owl ontologies with aroma. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 830–831. ACM, 2006.

[15] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *Proceedings of the 1st European Semantic Web Symposium*, volume 3053 of *LNCS*, pages 61–75. Springer Berlin Heidelberg, 2004.

[16] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1218–1232, 2009.

[17] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Agreementmaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589, August 2009.

[18] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology alignment for linked open data. In *Proceedings of the 9th International Semantic Web Conference*, volume 6496 of *LNCS*, pages 402–417. Springer Berlin Heidelberg, 2010.

[19] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Berlin Heidelberg, Heidelberg, 2007.

[20] Jérôme Euzenat. Towards a principled approach to semantic interoperability. In *Proceedings of IJCAI Workshop on Ontologies and Information Sharing*, pages 19–25, 2001.

[21] Michel Klein. Combining and relating ontologies: An analysis of problems and solutions. In *Proceedings of IJCAI Workshop on Ontologies and Information Sharing*, pages 53–62, 2001.

[22] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *ACM SIGMOD Record*, 35:34–41, 2006.

[23] Shvaiko Pavel and Jérôme Euzenat. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013.

[24] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[25] Ron Kohavi. The power of decision tables. In *Proceedings of the 8th European Conference on Machine Learning*, pages 174 – 189. Springer Berlin Heidelberg, 1995.

[26] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[27] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. Springer Berlin Heidelberg, 2nd edition, 2009.

[28] Won Kim and Jungyun Seo. Classifying schematic and data heterogeneity in multidatabase systems. *Computer*, 24(12):12–18, 1991.

[29] Holger Wache, Thomas Vögele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner. Ontology-based integration of information - a survey of existing approaches. In *Proceedings of IJCAI Workshop on Ontologies and Information Sharing*, pages 108–117, 2001.

[30] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[31] Matthew A. Jaro. Unimatch: A record linkage system: User's manual. Technical report, U.S. Bureau of the Census, Washington, D.C., 1976.

[32] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa. *Journal of the American Statistical Association*, 84(406): 414–420, 1989.

[33] William E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[34] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet::similarity: Measuring the relatedness of concepts. In *Proceedings of the 19th National Conference on Artificial Intelligence*, pages 1024–1025. Association for Computational Linguistics, 2004.

[35] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[36] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc., 1998.

[37] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19–33, 1997.

[38] Claudia Leacock, George A. Miller, and Martin Chodorow. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1): 147–165, 1998.

[39] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[40] Graeme Hirst and David St-Onge. *Lexical Chains as Representation of Context for the Detection and Correction Malapropisms*, pages 305–332. The MIT Press, 1998.

[41] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 805–810. Morgan Kaufmann Publishers Inc., 2003.

[42] Siddharth Patwardhan. Incorporating dictionary and corpus information into a context vector measure of semantic relatedness. Master's thesis, University of Minnesota, 2003.

[43] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[44] Qiu Ji, Peter Haase, and Guilin Qi. Combination of similarity measures in ontology matching using the owa operator. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Sytems*, pages 243–250, 2008.

[45] Ryutaro Ichise. Machine learning approach for ontology mapping using multiple concept similarity measures. In *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science*, pages 340–346, 2008.

[46] Lihua Zhao and Ryutaro Ichise. Aggregation of similarity measures in ontology matching. In *Proceedings of the 5th International Workshop on Ontology Matching*, pages 232–233, 2010.

[47] Ming Mao, Yefei Peng, and Michael Spring. An adaptive ontology mapping approach with neural network based constraint satisfaction. In *Web Semantics: Science, Services and Agents on the World Wide Web*, number 8, pages 14–25, 2010.

[48] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Linking and building ontologies of linked data. In *Proceedings of the 9th International Semantic Web Conference*, pages 598–614, 2010.

[49] Rahul Parundekar, Jose Luis Ambite, and Craig A. Knoblock. Aligning unions of concepts in ontologies of geospatial linked data. In *Proceedings of the Terra Cognita 2011 Workshop in Conjunction with the 10th International Semantic Web Conference*, Bonn, Germany, 2011.

[50] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Discovering concept coverings in ontologies of linked data sources. In *Proceedings of the 11th International Semantic Web Conference*, volume 7649 of *LNCS*, pages 427–443. Springer Berlin Heidelberg, 2012.

[51] Christian Meilicke, Johanna Völker, and Heiner Stuckenschmidt. Learning disjointness for debugging mappings between lightweight ontologies. In *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management*, volume 5268 of *LNCS*, pages 93–108. Springer Berlin Heidelberg, 2008.

[52] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11 (2):37–50, 1912.

[53] Mariana Damova, Atanas Kiryakov, Kiril Simov, and Svetoslav Petrov. Mapping the central lod ontologies to proton upper-level ontology. In *Proceedings of the 5th International Workshop on Ontology Matching*, pages 61–72, 2010.

[54] Prateek Jain, Peter Z. Yeh, Kunal Verma, Reymonrod G. Vasquez, Mariana Damova, Pascal Hitzler, and Amit P. Sheth. Contextual ontology alignment of lod with an upper ontology: A case study with proton. In *Proceedings of the 8th extended semantic web conference on The semantic web*, pages 80–92. Springer Berlin Heidelberg, 2011.

[55] DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov. Opening the black box of ontology matching. In *Proceedings of the 10th Extended Semantic Web Conference*, page 1630. Springer Berlin Heidelberg, 2013.

[56] Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A string metric for ontology alignment. In *Proceedings of the 4th International Semantic Web Conference*, pages 624–637. Springer Berlin Heidelberg, 2005.

[57] Isabel F. Cruz, Cosmin Stroe, Michele Caci, Federico Caimi, Matteo Palmonari, Flavio Palandri Antonelli, and Ulas C. Keles. Using agreementmaker to align ontologies for oaei 2010. In *OM*. CEUR-WS.org, 2010.

[58] Ming Mao, Yefei Peng, and Michael Spring. A harmony based adaptive ontology mapping approach. In *Proceedings of the 2008 International Conference on Semantic Web & Web Services*, pages 336–342. CSREA Press, 2008.

[59] DuyHoa Ngo, Zohra Bellahsene, and Remi Coletta. A flexible system for ontology matching. In *IS Olympics: Information Systems in a Diverse World*, volume 107 of *Lecture Notes in Business Information Processing*, pages 79–94. Springer Berlin Heidelberg, 2012.

[60] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[61] Li Ding, Joshua Shinavier, Zhenning Shangguan, and Deborah L. McGuinness. SameAs networks and beyond: Analyzing deployment status and implications of owl:sameAs in linked data. In *Proceedings of the 9th International Semantic Web Conference*, pages 145–160, 2010.

[62] Li Ding, Joshua Shinavier, Tim Finin, and Deborah L. McGuinness. owl:sameAs and linked data: An empirical study. In *Proceedings of the 2nd Web Science Conference*, 2010.

[63] Sören Auer and Jens Lehmann. Creating knowledge out of interlinked data. *Semantic Web Journal*, 1(1,2):97–104, 2010.

[64] Sören Auer. Creating knowledge out of interlinked data: making the web a data washing machine. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, pages 4:1–4:8, New York, NY, USA, 2011. ACM.

[65] Gianluca Correndo, Antonio Penta, Nicholas Gibbins, and Nigel Shadbolt. Statistical analysis of the owl:sameAs network for aligning concepts in the linking open data cloud. In *Database and Expert Systems Applications*, volume 7447 of *LNCS*, pages 215–230. Springer Berlin Heidelberg, 2012.

[66] David Geer. Reducing the storage burden via data deduplication. *IEEE Computer*, 41(12):15–17, 2008.

[67] Ee-Peng Lim, Jaideep Srivastava, Satya Prabhakar, and James Richardson. Entity identification in database integration. In *Proceedings of the 9th International Conference on Data Engineering*, pages 294–301. IEEE Computer Society, 1993.

[68] Abhirup Chatterjee and Arie Segev. Data manipulation in heterogeneous databases. *SIGMOD Record*, 20(4):64–68, 1991.

[69] Hugh Glaser, Afraz Jaffri, and Ian Millard. Managing co-reference on the semantic web. In *Proceedings of the Linked Data on the Web Workshop*, volume 538 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[70] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37: 547–579, 1901.

[71] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes-a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2312–2317, 2011.

[72] Christian Bizer, Tom Heath, Tim Berners-Lee, and Kingsley Idehen, editors. *Silk - A Link Discovery Framework for the Web of Data*, volume 538 of *CEUR Workshop Proceedings*, 2009. CEUR-WS.org.

[73] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of the 8th International Semantic Web Conference*, pages 650–665. Springer Berlin Heidelberg, 2009.

[74] Robert Isele, Anja Jentzsch, and Christian Bizer. Silk server - adding missing links while consuming linked data. In Olaf Hartig, Andreas Harth, and Juan Sequeda, editors, *Proceedings of the 1st International Workshop on Consuming Linked Data*, volume 665 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[75] Jeremy Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs. *Journnal of Web Semantics*, 3(4):247–267, 2005.

[76] Silvio Peroni, Enrico Motta, and Mathieu dAquin. Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In John Domingue and Chutiporn Anutariya, editors, *The Semantic Web*, volume 5367 of *LNCS*, pages 242–256. 2008.

[77] Jozef Vrana and Marián Mach. Ontology key concepts interpretation. In *Proceedings of the 8th IEEE International Symposium on Applied machine Intelligence and Informatics*, pages 215–219, 2010.

[78] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.

[79] Jennifer Sleeman and Tim Finin. A machine learning approach to linking foaf instances. In *Proceedings of the AAAI Spring Symposium on Linked Data Meets Artificial Intelligence*. AAAI, 2010.

[80] Dan Brickley and Libby Miller. *FOAF Vocabulary Specification 0.98.* W3C Recommendation, 2010. `http://xmlns.com/foaf/spec/`.

[81] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, 2002.

[82] Ziawasch Abedjan and Felix Naumann. Synonym analysis for predicate expansion. In *Proceedings of the 10th Extended Semantic Web Conference*, pages 140–154. Springer Berlin Heidelberg, 2013.

[83] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1–12. ACM, 2000.

[84] Ngoc-Thanh Le, Ryutaro Ichise, and Hoai-Bac Le. Detecting hidden relations in geographic data. In *Proceedings of the 4th International Conference on Advances in Semantic Processing*, pages 61–68, 2010.

[85] Lucas Drumond and Rosario Girardi. A survey of ontology learning procedures. In *Proceedings of the 3rd Workshop on Ontologies and their Applications*, volume 427 of *CEUR Workshop Proceedings*, 2008.

[86] Lina Zhou. Ontology learning: State of the art and open issues. *Information Technology and Management*, 8:241–252, 2007.

[87] Paul Buitelaar and Bernardo Magnini. Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press, 2005.

[88] Jon Atle Gulla, Hans Olaf Borch, and Jon Espen Ingvaldsen. Ontology learning for search applications. In *Proceedings of the 6th International Conference on Ontologies, DataBases, and Applications of Semantics*, volume 4803, pages 1050–1062, 2007.

[89] Lihua Zhao and Ryutaro Ichise. Mid-ontology learning from linked data. In *Proceedings of the 1st Joint International Semantic Technology Conference*, pages 112–127, 2011.

[90] Lihua Zhao and Ryutaro Ichise. Integrating heterogeneous ontology schema from lod. In *Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence, 3C1-OS-13a-4*, 2012.

[91] Lihua Zhao and Ryutaro Ichise. Integrating ontologies using ontology learning approach. *IEICE Transactions on Information and Systems*, Vol.E96-D(1):40–50, 2013.

[92] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference*, pages 11–15. Springer Berlin Heidelberg, 2007.

[93] Harry Halpin, Patrick J. Hayes, James P. McCusker, Deborah L. McGuinness, and Henry S. Thompson. When owl:sameAs isn't the same: An analysis of identity in linked data. In *Proceedings of the 9th International Semantic Web Conference*, pages 305–320, 2010.

[94] Rutaro Ichise. An analysis of multiple similarity measures for ontology mapping problem. *International Journal of Semantic Computing*, 4(1):103–122, 2010.

[95] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[96] M. F. Porter. An algorithm for suffix stripping. In *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[97] Philipp Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., 2006.

[98] Orri Erling and Ivan Mikhailov. Rdf support in the virtuoso dbms. In *Networked Knowledge-Networked Media*, volume 221, pages 7–24. Springer Berlin Heidelberg, 2009.

[99] Lihua Zhao and Ryutaro Ichise. One simple ontology for linked data sets. In *Proceedings of the 10th International Semantic Web Conference: Posters and Demos*, 2011.

[100] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res*, 13(11):2498–2504, 2003.

[101] Lihua Zhao and Ryutaro Ichise. Graph-based ontology analysis in the linked open data. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 56–63. ACM, 2012.

[102] William E Winkler. Overview of record linkage and current research directions. Technical report, Statistical Research Division U.S. Bureau of the Census, 2006.

[103] Lihua Zhao and Ryutaro Ichise. Instance-based ontological knowledge acquisition. In *Proceedings of the 10th Extended Semantic Web Conference*, pages 155–169. Springer Berlin Heidelberg, 2013.

[104] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[105] María Poveda-Villalón, María del Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Validating ontologies with oops! In *Knowledge Engineering and Knowledge Management*, volume 7603 of *LNCS*, pages 267–281. Springer Berlin Heidelberg, 2012.

[106] *OWL Validator*. `http://owl.cs.manchester.ac.uk/validator/`.

[107] Lihua Zhao and Ryutaro Ichise. Accessing linked data with a simple integrated ontology. In *Proceedings of the 27th Annual Conference of the Japanese Society for Artificial Intelligence, 4C1-IOS-4b-1*, 2013.

owl:Thing
- Bridge
  - db-onto:Bridge
  - geo-onto:S.BDG
  - geo-onto:S.BDGQ
- FilmCrewGig
  - mdb-movie:film_crew_gig
  - nyt:nytd_per
  - db-onto:Person
- Work
  - db-onto:Work
  - mdb-movie:film_job
- Sales
  - db-onto:Sales
- Disease
  - db-onto:Disease
  - nyt:nytd_des
- L
  - geo-onto:L
  - nyt:nytd_geo
- Island
  - geo-onto:T.ISL
  - db-onto:Island
  - nyt:nytd_geo
- Drug
  - db-onto:Drug
  - nyt:nytd_des
- Nascardriver
  - nyt:nytd_per
  - db-onto:NascarDriver
- Airport
  - geo-onto:S.AIRP
  - db-onto:Airport
  - nyt:nytd_geo
- Hockeyteam
  - db-onto:HockeyTeam
  - nyt:nytd_org
- MeanOfTransportation
  - db-onto:MeanOfTransportation
- Settlement
  - geo-onto:P
  - db-onto:Town
  - db-onto:City
  - db-onto:Station
  - geo-onto:P.PPL
  - nyt:nytd_geo
  - db-onto:Village
  - db-onto:Settlement
- Language
  - db-onto:Language
- Broadcast
  - geo-onto:S.TOWR
  - db-onto:Broadcast
  - nyt:nytd_org
- Scientist
  - db-onto:Scientist
  - nyt:nytd_per
- Basketballteam
  - db-onto:BasketballTeam
  - nyt:nytd_org
- Actor
  - db-onto:AdultActor
  - db-onto:Person
  - mdb-movie:actor
  - nyt:nytd_per
  - db-onto:Actor
- Species
  - db-onto:Species
- LegalCase
  - db-onto:LegalCase
- Activity
  - db-onto:Activity
- Organisation
  - nyt:nytd_org
  - db-onto:Non-ProfitOrganisation
  - db-onto:Organisation
- Director
  - mdb-movie:director
  - nyt:nytd_per
  - db-onto:Person
- Beverage
  - db-onto:Beverage
- Icehockeyplayer
  - nyt:nytd_per
  - db-onto:IceHockeyPlayer
- Award
  - db-onto:Award
- Philosopher
  - nyt:nytd_per
  - db-onto:Philosopher
- Lake
  - db-onto:Place
  - geo-onto:H.LK
  - geo-onto:H.RSV
  - db-onto:BodyOfWater
  - geo-onto:H
  - db-onto:Lake
- Musicalartist
  - nyt:nytd_per
  - db-onto:MusicalArtist

- OlympicResult
  - db-onto:OlympicResult
- Museum
  - geo-onto:S.MUS
  - nyt:nytd_org
  - geo-onto:S.BLDG
  - db-onto:Museum
- Figureskater
  - nyt:nytd_per
  - db-onto:FigureSkater
- Baseballplayer
  - nyt:nytd_per
  - db-onto:BaseballPlayer
- Primeminister
  - nyt:nytd_per
  - db-onto:PrimeMinister
- Map
  - db-onto:Place
  - nyt:nytd_geo
- Congressman
  - nyt:nytd_per
  - db-onto:Congressman
- FilmCharacter
  - mdb-movie:film_character
  - nyt:nytd_per
  - db-onto:FictionalCharacter
- Architect
  - nyt:nytd_per
  - db-onto:Architect
- Infrastructure
  - db-onto:Infrastructure
- Radiostation
  - geo-onto:S.TOWR
  - geo-onto:S.STNR
  - db-onto:RadioStation
- FilmFilmDistributorRelationship
  - mdb-movie:film_film_distributor_relationship
- Americanfootballteam
  - nyt:nytd_org
  - db-onto:AmericanFootballTeam
- Writer
  - mdb-movie:writer
  - db-onto:Writer
- Film
  - db-onto:Film
  - db-onto:TelevisionShow
  - mdb-movie:film
  - nyt:nytd_des
- EthnicGroup
  - db-onto:EthnicGroup
  - nyt:nytd_org
- PersonFunction
  - db-onto:PersonFunction
- Tennisplayer
  - nyt:nytd_per
  - db-onto:TennisPlayer
- Producer
  - mdb-movie:producer
  - nyt:nytd_per
  - db-onto:Person
- Tradeunion
  - nyt:nytd_org
  - db-onto:TradeUnion
- Protein
  - db-onto:Protein
- Britishroyalty
  - db-onto:BritishRoyalty
  - nyt:nytd_per
- U
  - geo-onto:U
  - nyt:nytd_geo
- FilmCut
  - mdb-movie:film_cut
- Journalist
  - nyt:nytd_per
  - db-onto:Journalist
- MusicGenre
  - db-onto:MusicGenre
- Person
  - db-onto:Person
  - nyt:nytd_per
- V
  - geo-onto:V
  - nyt:nytd_geo
- Mountain
  - geo-onto:T.VLC
  - geo-onto:T.PK
  - db-onto:Mountain
  - nyt:nytd_geo
  - db-onto:Place
  - geo-onto:T.MT
  - geo-onto:T
- Country
  - nyt:nytd_geo
  - mdb-movie:country
  - geo-onto:A.PCLI
  - geo-onto:A.PCLD
  - db-onto:Country

- Event
    - db-onto:Event
    - db-onto:FilmFestival
    - mdb-movie:film_festival_event
- President
    - db-onto:President
    - nyt:nytd_per
- Collegecoach
    - db-onto:CollegeCoach
    - nyt:nytd_per
- Protectedarea
    - nyt:nytd_geo
    - geo-onto:L.PRK
    - geo-onto:L.RES
    - db-onto:ProtectedArea
- Militaryunit
    - nyt:nytd_org
    - db-onto:MilitaryUnit
- Militaryperson
    - nyt:nytd_per
    - db-onto:MilitaryPerson
- ChemicalCompound
    - db-onto:ChemicalCompound
- Colour
    - db-onto:Colour
- EducationalInstitution
    - geo-onto:S.SCH
    - geo-onto:S.SCHC
    - geo-onto:S.UNIV
    - db-onto:University
    - db-onto:College
    - nyt:nytd_org
    - db-onto:School
- S
    - geo-onto:S
    - nyt:nytd_geo
- Newspaper
    - nyt:nytd_org
    - db-onto:Newspaper
- Performance
    - mdb-movie:performance
- Website
    - db-onto:Website
- R
    - geo-onto:R
    - nyt:nytd_geo
- Golfplayer
    - nyt:nytd_per
    - db-onto:GolfPlayer
- Stadium
    - geo-onto:S.STDM
    - db-onto:Stadium
    - nyt:nytd_geo
- Politician
    - nyt:nytd_per
    - db-onto:Politician
- Americanfootballplayer
    - nyt:nytd_per
    - db-onto:AmericanFootballPlayer
- Device
    - db-onto:Device
- NytdDes
    - nyt:nytd_des
- Historicplace
    - nyt:nytd_org
    - db-onto:HistoricPlace
    - geo-onto:S.HSTS
- River
    - db-onto:Stream
    - db-onto:River
    - nyt:nytd_geo
    - geo-onto:H.STM
- Officeholder
    - nyt:nytd_per
    - db-onto:OfficeHolder
- Currency
    - db-onto:Currency
- AnatomicalStructure
    - db-onto:AnatomicalStructure
- Mountainrange
    - nyt:nytd_geo
    - geo-onto:T.MTS
    - db-onto:MountainRange
- Basketballplayer
    - nyt:nytd_per
    - db-onto:BasketballPlayer
- AdministrativeRegion
    - geo-onto:A.ADM1
    - db-onto:PopulatedPlace
    - geo-onto:P.PPLA3
    - geo-onto:P
    - geo-onto:A.ADM3
    - db-onto:Place
    - geo-onto:P.PPLX
    - geo-onto:P.PPLA
    - geo-onto:A
    - geo-onto:A.ADM2
    - geo-onto:P.PPL
    - geo-onto:A.ADM4
    - db-onto:AdministrativeRegion
    - geo-onto:P.PPLA2
    - nyt:nytd_geo
    - geo-onto:P.PPLC

```xml
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:fiton-prop="http://ri-www.nii.ac.jp/fiton/property/"
   xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Bridge">
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Bridge"/>
    <rdfs:comment xml:lang="en">bridge</rdfs:comment>
    <rdfs:comment xml:lang="en">a structure erected across an obstacle such as a stream, road, etc., in order to carry roads, railroads, and pedestrians across</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.BDG"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.BDGQ"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/city">
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">city</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/city"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/FilmCrewGig">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/film_crew_gig"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Person"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Work">
    <rdfs:comment xml:lang="en">work</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Work"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/film_job"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/terminus">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">terminus a</rdfs:comment>
    <rdfs:comment xml:lang="en">terminus b</rdfs:comment>
```

```
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/terminusA"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/terminusB"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Sales">
  <rdfs:comment xml:lang="en">sales</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Sales"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Disease">
  <rdfs:comment xml:lang="en">disease</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Disease"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_des"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/runtime">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/minute"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/runtime"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
  <rdfs:comment xml:lang="en">runtime (m)</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/runtime"/>
  <rdfs:comment xml:lang="en">runtime</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/Work/runtime"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/runtime"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/second"/>
  <rdfs:comment xml:lang="en">runtime (s)</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/o">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">o</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/o"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/place">
<rdfs:range rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
<rdfs:comment xml:lang="en">place</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/place"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/L">
<rdfs:comment xml:lang="en">parks,area, ...</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#L"/>
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Island">
<rdfs:comment xml:lang="en">a tract of land, smaller than a continent, surrounded by water at high water</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#T.ISL"/>
<rdfs:comment xml:lang="en">island</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Island"/>
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/artist">
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
<rdfs:domain rdf:resource="http://dbpedia.org/ontology/Work"/>
<rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/Person"/>
<rdfs:comment xml:lang="en">artist</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/artist"/>
<rdfs:domain rdf:resource="http://dbpedia.org/ontology/Album"/>
<rdfs:domain rdf:resource="http://dbpedia.org/ontology/MusicalWork"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmCutNote">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/film_cut_note"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>

```
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/latin">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">latin</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latin"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/parentCountry">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#parentFeature"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:comment xml:lang="en">level 2 administrative parent</rdfs:comment>
  <rdfs:comment xml:lang="en">level 1 administrative parent</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">level 4 administrative parent</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#parentADM1"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#parentCountry"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#parentADM2"/>
  <rdfs:comment xml:lang="en">parent feature</rdfs:comment>
  <rdfs:range rdf:resource="http://www.geonames.org/ontology#Feature"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#parentADM3"/>
  <rdfs:comment xml:lang="en">a feature parent of the current one, in either administrative or physical subdivision.</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#parentADM4"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:comment xml:lang="en">level 3 administrative parent</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  <rdfs:comment xml:lang="en">parent country</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/instrument">
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Instrument"/>
  <rdfs:comment xml:lang="en">instrument</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/instrument"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
```

```
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/inclination">
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Planet"/>
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">inclination</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/inclination"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/class">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">class</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/class"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Drug">
    <rdfs:comment xml:lang="en">drug</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Drug"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_des"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Nascardriver">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdfs:comment xml:lang="en">nascar driver</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/NascarDriver"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Airport">
    <rdfs:comment xml:lang="en">a place where aircraft regularly land and take off, with runways, navigational aids, and major facilities for the commercial handling of passengers and cargo</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.AIRP"/>
    <rdfs:comment xml:lang="en">airport</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Airport"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
```

```
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/writerWriterid">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/writer_writerid"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Writer"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Hockeyteam">
  <rdfs:comment xml:lang="en">hockey team</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/HockeyTeam"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/year">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">year</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/year"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/regnum">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">regnum</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/regnum"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/motto">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/motto"/>
  <rdfs:comment xml:lang="en">mottoeng</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/mottoeng"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">motto</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/motto"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/placeOfDeath">
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Place"/>
  <rdfs:comment xml:lang="en">deathplace</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:comment xml:lang="en">the place where they died</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/placeOfDeath"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/deathPlace"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/deathplace"/>
  <rdfs:comment xml:lang="en">death place</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/deathPlace"/>
  <rdfs:comment xml:lang="en">place of death</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation">
  <rdfs:comment xml:lang="en">mean of transportation</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/MeanOfTransportation"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Settlement">
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P"/>
  <rdfs:comment xml:lang="en">town</rdfs:comment>
  <rdfs:comment xml:lang="en">populated place</rdfs:comment>
  <rdfs:comment xml:lang="en">a relatively large and permanent settlement, particularly a large urban settlement</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Town"/>
  <rdfs:comment xml:lang="en">a city, town, village, or other agglomeration of buildings where people live and work</rdfs:comment>
  <rdfs:comment xml:lang="en">city</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/City"/>
  <rdfs:comment xml:lang="en">city, village,...</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Station"/>
  <rdfs:comment xml:lang="en">village</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPL"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Village"/>
  <rdfs:comment xml:lang="en">municipality</rdfs:comment>
```

```
  <rdfs:comment xml:lang="en">station</rdfs:comment>
  <rdfs:comment xml:lang="en">a clustered human settlement or community, usually smaller a town</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Settlement"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Language">
  <rdfs:comment xml:lang="en">language</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Language"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Broadcast">
  <rdfs:comment xml:lang="en">a high conspicuous structure, typically much higher than its diameter</rdfs:comment>
  <rdfs:comment xml:lang="en">tower</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.TOWR"/>
  <rdfs:comment xml:lang="en">broadcast</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Broadcast"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Scientist">
  <rdfs:comment xml:lang="en">scientist</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Scientist"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/latestUse">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Golfplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Tennisplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountainrange"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Militaryperson"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Hockeyteam"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Britishroyalty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Basketballteam"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Newspaper"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Journalist"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Politician"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Country"/>
```

```
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Philosopher"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Historicplace"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Congressman"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Figureskater"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Icehockeyplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Scientist"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EducationalInstitution"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Collegecoach"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Officeholder"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Basketballplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Tradeunion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protectedarea"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/President"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Museum"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Militaryunit"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Musicalartist"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballteam"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.nytimes.com/elements/latest_use"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Nascardriver"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Architect"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/NytdDes"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Baseballplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Primeminister"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/nearbyFeatures">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:range rdf:resource="http://www.geonames.org/ontology#RDFData"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
```

```
    <rdfs:comment xml:lang="en">nearby features</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#nearbyFeatures"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/capital">
    <rdfs:comment xml:lang="en">capital</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/capital"/>
    <rdfs:comment xml:lang="en">largest city</rdfs:comment>
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/largestCity"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/country_capital"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/comment">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
    <rdfs:comment xml:lang="en">short description</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/abstract"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Colour"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/NytdDes"/>
```

```
  <rdfs:comment xml:lang="en">description</rdfs:comment>
  <rdfs:comment xml:lang="en">has abstract</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/canton"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2000/01/rdf-schema#comment"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shortDescription"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://purl.org/dc/elements/1.1/description"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
  <rdfs:comment xml:lang="en">an account of the resource.</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2004/02/skos/core#scopeNote"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  <rdfs:comment xml:lang="en">canton</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Basketballteam">
  <rdfs:comment xml:lang="en">basketball team</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/BasketballTeam"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/leadertitle">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/leaderTitle"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/leaderTitle"/>
```

```
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/property/b_percent_C3_percent_BCrgermeistertitel"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdfs:comment xml:lang="en">bürgermeistertitel</rdfs:comment>
    <rdfs:comment xml:lang="en">leader title</rdfs:comment>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Actor">
    <rdfs:comment xml:lang="en">adult actor</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/AdultActor"/>
    <rdfs:comment xml:lang="en">person</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Person"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/actor"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdfs:comment xml:lang="en">actor</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Actor"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/debut">
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">debut</rdfs:comment>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/debut"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/actorNetflixId">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/actor_netflix_id"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/atcPrefix">
```

```
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/nicaraguanCórdoba"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">atc prefix</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/atcPrefix"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shipName">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">ship name</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shipName"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Ship"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/MeanOfTransportation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/actor">
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/actor"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/imagesize">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">imagesize</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/imagesize"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Species">
  <rdfs:comment xml:lang="en">species</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Species"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/height">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/heightIn"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/height"/>
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:comment xml:lang="en">height (m)</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/Person/height"/>
  <rdfs:comment xml:lang="en">height ft</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/foot"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/height"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/heightFt"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <rdfs:comment xml:lang="en">height (cm)</rdfs:comment>
  <rdfs:comment xml:lang="en">height in</rdfs:comment>
  <rdfs:comment xml:lang="en">height</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/inch"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/centimetre"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/deathdate">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/deathDate"/>
  <rdfs:comment xml:lang="en">died</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">date of death</rdfs:comment>
  <rdfs:comment xml:lang="en">death date</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/deathDate"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/died"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/dateOfDeath"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <rdfs:comment xml:lang="en">deathdate</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/deathdate"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/publisher">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
```

```
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/publisher"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Company"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/publisher"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:comment xml:lang="en">publisher</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase">
  <rdfs:comment xml:lang="en">legal case</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/LegalCase"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/n">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">n</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/n"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Activity">
  <rdfs:comment xml:lang="en">activity</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Activity"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/seeAlso">
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2000/01/rdf-schema#SeeAlso"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Writer"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/route">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">route</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/route"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
```

```xml
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Organisation">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
    <rdfs:comment xml:lang="en">non-profit organisation</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Non-ProfitOrganisation"/>
    <rdfs:comment xml:lang="en">organisation</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Organisation"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Director">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/director"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Person"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Beverage">
    <rdfs:comment xml:lang="en">beverage</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Beverage"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/featureClass">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:comment xml:lang="en">feature class</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#featureClass"/>
    <rdfs:range rdf:resource="http://www.geonames.org/ontology#Class"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
    <rdfs:comment xml:lang="en">the main category of the feature, as defined in geonames
taxonomy.</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Icehockeyplayer">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
```

```
    <rdfs:comment xml:lang="en">ice hockey player</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/IceHockeyPlayer"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/origin">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">origin</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/origin"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmReleaseRegion">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_release_region"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Award">
    <rdfs:comment xml:lang="en">award</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Award"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Philosopher">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdfs:comment xml:lang="en">philosopher</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Philosopher"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Lake">
    <rdfs:comment xml:lang="en">a location</rdfs:comment>
    <rdfs:comment xml:lang="en">reservoir(s)</rdfs:comment>
    <rdfs:comment xml:lang="en">place</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Place"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#H.LK"/>
    <rdfs:comment xml:lang="en">a large inland body of standing water</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#H.RSV"/>
    <rdfs:comment xml:lang="en">an artificial pond or lake</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/BodyOfWater"/>
```

```
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#H"/>
  <rdfs:comment xml:lang="en">lake</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Lake"/>
  <rdfs:comment xml:lang="en">stream, lake, ...</rdfs:comment>
  <rdfs:comment xml:lang="en">body of water</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/isDefinedBy">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2000/01/rdf-schema#isDefinedBy"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Musicalartist">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">musical artist</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/MusicalArtist"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/stumpings">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">catches/stumpings</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/catches/stumpings"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Athlete"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Cricketer"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmCutFilmCutid">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
```

```
  <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_cut_film_cutid"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/kingdom">
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">kingdom</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/kingdom"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/subject">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
  <rdfs:comment xml:lang="en">the topic of the resource.</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdfs:comment xml:lang="en">subject</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://purl.org/dc/terms/subject"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
```

```
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/startDate">
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/careerStart"/>
    <rdfs:comment xml:lang="en">term start</rdfs:comment>
    <rdfs:comment xml:lang="en">date</rdfs:comment>
    <rdfs:comment xml:lang="en">a point or period of time associated with an event in the lifecycle of the
resource.</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/draftyear"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
    <rdfs:comment xml:lang="en">draft year</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/activeYearsStartDate"/>
    <rdfs:comment xml:lang="en">draftyear</rdfs:comment>
    <rdfs:comment xml:lang="en">turnedpro</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
    <rdfs:comment xml:lang="en">active years start date</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:comment xml:lang="en">career start</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:comment xml:lang="en">debutyear</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://purl.org/dc/terms/date"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/termStart"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/years"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/draftYear"/>
    <rdfs:comment xml:lang="en">start</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/turnedpro"/>
    <rdfs:comment xml:lang="en">years</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/debutyear"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/start"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult">
  <rdfs:comment xml:lang="en">olympic result</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/OlympicResult"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Museum">
  <rdfs:comment xml:lang="en">a building where objects of permanent interest in one or more of the arts and sciences are preserved and exhibited</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.MUS"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdfs:comment xml:lang="en">a structure built for permanent use, as a house, factory, etc.</rdfs:comment>
  <rdfs:comment xml:lang="en">building(s)</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.BLDG"/>
  <rdfs:comment xml:lang="en">museum</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Museum"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/placeOfBirth">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/born"/>
  <rdfs:comment xml:lang="en">birthplace</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:comment xml:lang="en">birth place</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Place"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/birthplace"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:comment xml:lang="en">placeofbirth</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/birthPlace"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/placeofbirth"/>
  <rdfs:comment xml:lang="en">placebirth</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/birthPlace"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/placeOfBirth"/>
  <rdfs:comment xml:lang="en">where the person was born</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">place of birth</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/placebirth"/>
  <rdfs:comment xml:lang="en">born</rdfs:comment>
```

```
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmCrewGigFilm">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_crew_gig_film"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCrewGig"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/website">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Congressman"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Tradeunion"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Writer"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Militaryperson"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Performance"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Golfplayer"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballplayer"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Basketballplayer"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Nascardriver"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCrewGig"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/page"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Collegecoach"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Journalist"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Icehockeyplayer"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Producer"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://data.nytimes.com/elements/topicPage"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Politician"/>
```

```
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EducationalInstitution"/>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Hockeyteam"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
<rdfs:comment xml:lang="en">website</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballteam"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/website"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Newspaper"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Figureskater"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Historicplace"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/NytdDes"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/homepage"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
<rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Britishroyalty"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Primeminister"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
<rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Tennisplayer"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Militaryunit"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Baseballplayer"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/wikiPageExternalLink"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCharacter"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Philosopher"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Museum"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Officeholder"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
```

```
    <rdfs:comment xml:lang="en">link from a wikipage to an external page</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Colour"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Musicalartist"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Director"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Scientist"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/President"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Basketballteam"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Architect"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/basinCountries">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
    <rdfs:comment xml:lang="en">basin countries</rdfs:comment>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/BodyOfWater"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/basinCountries"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmOfDistributor">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_of_distributor"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmFilmDistributorRelationship"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/hasPhotoCollection">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
```

```
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/hasPhotoCollection"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmDistributor">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_distributor"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmFilmDistributorRelationship"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/students">
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/undergrad"/>
    <rdfs:comment xml:lang="en">number of undergraduate students</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/students"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment xml:lang="en">postgrad</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/enrollment"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/ontology/numberOfUndergraduateStudents"/>
    <rdfs:comment xml:lang="en">enrollment</rdfs:comment>
    <rdfs:comment xml:lang="en">undergrad</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/numberOfStudents"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EducationalInstitution"/>
    <rdfs:comment xml:lang="en">number of postgraduate students</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/postgrad"/>
    <rdfs:comment xml:lang="en">students</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
    <rdfs:comment xml:lang="en">number of students</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/ontology/numberOfPostgraduateStudents"/>
  </rdf:Description>
```

```
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Figureskater">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">figure skater</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/FigureSkater"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/performancePerformanceid">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/performance_performanceid"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Performance"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/country">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/country"/>
  <rdfs:comment xml:lang="en">country</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Country"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/country"/>
  <rdfs:comment xml:lang="en">the countrycode value for a geoname feature is equal to the countrycode value
of the parentcountry value.</rdfs:comment>
  <rdfs:comment xml:lang="en">iso country code</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
  <rdfs:comment xml:lang="en">a two letters country code in the iso 3166 list</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/country_fips_code"/>
  <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/country_iso_alpha2"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:comment xml:lang="en">coordinates region</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#countryCode"/>
```

```
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/coordinatesRegion"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/position">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/position"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:comment xml:lang="en">currentposition</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/currentposition"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">position</rdfs:comment>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/position"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/wordnetType">
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/wordnet_type"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/thumbnail">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/thumbnail"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
  <rdfs:comment xml:lang="en">thumbnail</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/population">
```

```
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/popLatest"/>
    <rdfs:comment xml:lang="en">einwohner</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/einwohner"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latestYear"/>
    <rdfs:comment xml:lang="en">population</rdfs:comment>
    <rdfs:comment xml:lang="en">population density (/sqkm)</rdfs:comment>
    <rdfs:comment xml:lang="en">latest year</rdfs:comment>
    <rdfs:comment xml:lang="en">population date</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:comment xml:lang="en">pop latest</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/populationTotal"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/population"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/country_population"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#population"/>
    <rdfs:comment xml:lang="en">population density km</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/populationTotal"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">population estimate</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/populationEstimate"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
    <rdfs:comment xml:lang="en">population total</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/populationDensity">
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/ontology/PopulatedPlace/populationDensity"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    <rdfs:comment xml:lang="en">population density (/sqkm)</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/populationDensity"/>
    <rdfs:range rdf:resource="http://dbpedia.org/datatype/inhabitantsPerSquareKilometre"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:comment xml:lang="en">population density km</rdfs:comment>
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/populationDensityKm"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/rank">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">rank</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/rank"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Baseballplayer">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">baseball player</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/BaseballPlayer"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/gridironFootballPlayerInfo">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/draftpick"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/draftround"/>
  <rdfs:comment xml:lang="en">draftpick</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/draftPick"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/draftRound"/>
  <rdfs:comment xml:lang="en">draftround</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">draft round</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/draftPick"/>
  <rdfs:comment xml:lang="en">draft pick</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/draftRound"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Athlete"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"/>
```

```xml
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Primeminister">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">prime minister</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/PrimeMinister"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Map">
  <rdfs:comment xml:lang="en">place</rdfs:comment>
  <rdfs:comment xml:lang="en">a location</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Place"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Congressman">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">congressman</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Congressman"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/FilmCharacter">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/film_character"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/FictionalCharacter"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/othernames">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">othernames</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/othernames"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/alias"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/alias"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Architect">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">architect</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Architect"/>
```

```
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure">
    <rdfs:comment xml:lang="en">infrastructure</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Infrastructure"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/configuration">
    <rdfs:range rdf:resource="http://dbpedia.org/datatype/engineConfiguration"/>
    <rdfs:comment xml:lang="en">configuration</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/configuration"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/featureCode">
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#featureCode"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
    <rdfs:comment xml:lang="en">type of the feature, as defined in geonames taxonomy.</rdfs:comment>
    <rdfs:comment xml:lang="en">feature code</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
    <rdfs:range rdf:resource="http://www.geonames.org/ontology#Code"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Radiostation">
    <rdfs:comment xml:lang="en">a high conspicuous structure, typically much higher than its
diameter</rdfs:comment>
    <rdfs:comment xml:lang="en">tower</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.TOWR"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.STNR"/>
    <rdfs:comment xml:lang="en">radio station</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/RadioStation"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
```

<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/FilmFilmDistributorRelationship">
<fiton-prop:hasMemberClasses
rdf:resource="http://data.linkedmdb.org/resource/movie/film_film_distributor_relationship"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/length">
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/Infrastructure/length"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
<rdfs:comment xml:lang="en">length (m)</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/r2LengthM"/>
<fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/ontology/MeanOfTransportation/length"/>
<rdfs:comment xml:lang="en">r2-length-m</rdfs:comment>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:comment xml:lang="en">r1-length-m</rdfs:comment>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
<rdfs:comment xml:lang="en">length</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/r1LengthM"/>
<rdfs:range rdf:resource="http://dbpedia.org/datatype/millimetre"/>
<rdfs:range rdf:resource="http://dbpedia.org/datatype/kilometre"/>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdfs:comment xml:lang="en">length (mm)</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/length"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/length"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:comment xml:lang="en">length (km)</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballteam">
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
<rdfs:comment xml:lang="en">american football team</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/AmericanFootballTeam"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmFilmDistributorRelationshipFilmCut">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_film_distributor_relationship_film_cut"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmFilmDistributorRelationship"/>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

```
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shipFate">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">ship fate</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shipFate"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/portrayer">
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Person"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/FictionalCharacter"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">portrayer</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/portrayer"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Writer">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/writer"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Writer"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/directorDirectorid">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/director_directorid"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Director"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Film">
  <rdfs:comment xml:lang="en">film</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Film"/>
  <rdfs:comment xml:lang="en">television show</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/TelevisionShow"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/film"/>
```

```
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_des"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup">
  <rdfs:comment xml:lang="en">ethnic group</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/EthnicGroup"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/PersonFunction">
  <rdfs:comment xml:lang="en">person function</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/PersonFunction"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/color">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">color</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/color"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/populationEstimateYear">
  <rdfs:comment xml:lang="en">population estimate year</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/gdpPppYear"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/populationAsOf"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">gdp nominal year</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">gdp ppp year</rdfs:comment>
  <rdfs:comment xml:lang="en">population as of</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/gdpNominalYear"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/populationEstimateYear"/>
```

```
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/map">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/mapBackground"/>
  <rdfs:comment xml:lang="en">pushpin map caption</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/pushpinMapCaption"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/mapLegend"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">map background</rdfs:comment>
  <rdfs:comment xml:lang="en">pushpin caption</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/map"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/mapCaption"/>
  <rdfs:comment xml:lang="en">map1 caption</rdfs:comment>
  <rdfs:comment xml:lang="en">map</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/pushpinMap"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/pushpinCaption"/>
  <rdfs:comment xml:lang="en">pushpin map</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:comment xml:lang="en">map-legend</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/mapCaption"/>
  <rdfs:comment xml:lang="en">map caption</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/numberOfVariants">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Baseballplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Newspaper"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Tennisplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Musicalartist"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Architect"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/NytdDes"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.nytimes.com/elements/number_of_variants"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Basketballplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Hockeyteam"/>
```

```
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/President"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protectedarea"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Militaryunit"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountainrange"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Museum"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Philosopher"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Collegecoach"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Scientist"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Historicplace"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Britishroyalty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EducationalInstitution"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Golfplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Nascardriver"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Officeholder"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballteam"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Icehockeyplayer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Politician"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Tradeunion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Militaryperson"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Figureskater"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Congressman"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Primeminister"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Journalist"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Country"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Basketballteam"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/fam">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">fam</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/fam"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/h">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">h</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/h"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shoots">
  <rdfs:comment xml:lang="en">shoots</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Icehockeyplayer"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shoots"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/shoots"/>
  <rdfs:comment xml:lang="en">shot</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shot"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/genre">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/genre"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">genre</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/genre"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Tennisplayer">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">tennis player</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/TennisPlayer"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/wikiPageUsesTemplate">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/wikiPageUsesTemplate"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
```

```
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Producer">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/producer"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Person"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Tradeunion">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdfs:comment xml:lang="en">trade union</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/TradeUnion"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Protein">
```

```
  <rdfs:comment xml:lang="en">protein</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Protein"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Britishroyalty">
  <rdfs:comment xml:lang="en">british royalty</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/BritishRoyalty"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/U">
  <rdfs:comment xml:lang="en">undersea</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#U"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/film_cut"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/originalairdate">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">originalairdate</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/originalairdate"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Work"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Journalist">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">journalist</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Journalist"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/formationYear">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gYear"/>
```

```
  <rdfs:comment xml:lang="en">formation year</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/formationYear"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/demonym">
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/demonym"/>
  <rdfs:comment xml:lang="en">population demonym</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/populationDemonym"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">demonym</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/demonym"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre">
  <rdfs:comment xml:lang="en">music genre</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/MusicGenre"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shipPropulsion">
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/kilowatt"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/second"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">ship propulsion</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shipPropulsion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Person">
  <rdfs:comment xml:lang="en">person</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Person"/>
```

```
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/imagefile">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:comment xml:lang="en">imagefile</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/imagefile"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/V">
<rdfs:comment xml:lang="en">forest, heath, ...</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#V"/>
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Mountain">
<rdfs:comment xml:lang="en">volcano</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#T.VLC"/>
<rdfs:comment xml:lang="en">place</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#T.PK"/>
<rdfs:comment xml:lang="en">a location</rdfs:comment>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:comment xml:lang="en">an elevation standing high above the surrounding area with small summit area, steep slopes and local relief of 300m or more</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Mountain"/>
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
<rdfs:comment xml:lang="en">a conical elevation composed of volcanic materials with a crater at the top</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Place"/>
<rdfs:comment xml:lang="en">mountain, hill, rock, ...</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#T.MT"/>
<rdfs:comment xml:lang="en">mountain</rdfs:comment>
<rdfs:comment xml:lang="en">a pointed elevation atop a mountain, ridge, or other hypsographic feature</rdfs:comment>
<rdfs:comment xml:lang="en">peak</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#T"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Country">
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
<fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/country"/>
```

```
  <rdfs:comment xml:lang="en">independent political entity</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A.PCLI"/>
  <rdfs:comment xml:lang="en">dependent political entity</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A.PCLD"/>
  <rdfs:comment xml:lang="en">country</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Country"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/water">
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
  <rdfs:comment xml:lang="en">percentage of area water</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/percentWater"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/hdi"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">percent water</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:comment xml:lang="en">hdi</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/percentageOfAreaWater"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/formationDate">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/histYr"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/openingDate"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">formation</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:comment xml:lang="en">opening date</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/formationDate"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/opened"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
```

```xml
  <rdfs:comment xml:lang="en">founding date</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
  <rdfs:comment xml:lang="en">foundation</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/established"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/nflStartYr"/>
  <rdfs:comment xml:lang="en">hist yr</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/establishedDate"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/founded"/>
  <rdfs:comment xml:lang="en">nfl start yr</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/foundation"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:comment xml:lang="en">open</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/formation"/>
  <rdfs:comment xml:lang="en">opened</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">established date</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/foundingDate"/>
  <rdfs:comment xml:lang="en">established</rdfs:comment>
  <rdfs:comment xml:lang="en">founded</rdfs:comment>
  <rdfs:comment xml:lang="en">formation date</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/open"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/income">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/netIncome"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/danishKrone"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <rdfs:comment xml:lang="en">net income</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/swedishKrona"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/swissFranc"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/indianRupee"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/southKoreanWon"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/croatianKuna"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/poundSterling"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/norwegianKrone"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/operatingIncome"/>
  <rdfs:comment xml:lang="en">net income ($)</rdfs:comment>
  <rdfs:comment xml:lang="en">operating income ($)</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/philippinePeso"/>
```

```
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/japaneseYen"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/iranianRial"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/bosniaAndHerzegovinaConvertibleMarks"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/netIncome"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/usDollar"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/operatingIncome"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/ukrainianHryvnia"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/euro"/>
  <rdfs:comment xml:lang="en">operating income</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/malaysianRinggit"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Organisation"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Company"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/displacement">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <rdfs:comment xml:lang="en">displacement (m3)</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/displacement"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/language">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/anthem"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Work"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Language"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/language"/>
  <rdfs:comment xml:lang="en">anthem</rdfs:comment>
  <rdfs:comment xml:lang="en">official language</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/officialLanguage"/>
  <rdfs:comment xml:lang="en">language</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/performance">
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/performance"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
```

```
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/surname">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/surname"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/url">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">url</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/url"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Event">
    <rdfs:comment xml:lang="en">event</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Event"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/FilmFestival"/>
    <fiton-prop:hasMemberClasses
rdf:resource="http://data.linkedmdb.org/resource/movie/film_festival_event"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/designer">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">designer</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/designer"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/President">
    <rdfs:comment xml:lang="en">president</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/President"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Collegecoach">
```

```
    <rdfs:comment xml:lang="en">college coach</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/CollegeCoach"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/first">
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">first</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/first"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Protectedarea">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
    <rdfs:comment xml:lang="en">an area, often of forested land, maintained as a place of beauty, or for
recreation</rdfs:comment>
    <rdfs:comment xml:lang="en">park</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#L.PRK"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#L.RES"/>
    <rdfs:comment xml:lang="en">protected area</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/ProtectedArea"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/date">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">date</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/date"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Militaryunit">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
    <rdfs:comment xml:lang="en">military unit</rdfs:comment>
```

```
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/MilitaryUnit"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/director">
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/director"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Militaryperson">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">military person</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/MilitaryPerson"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/weight">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/weight"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/kilogram"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">weight (kg)</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/weightLb"/>
  <rdfs:comment xml:lang="en">weight</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/tonne"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/ounce"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/weight"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
  <rdfs:comment xml:lang="en">weight lb</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:comment xml:lang="en">weight (g)</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/pound"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/gram"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/Person/weight"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound">
  <rdfs:comment xml:lang="en">chemical compound</rdfs:comment>
```

```
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/ChemicalCompound"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Colour">
    <rdfs:comment xml:lang="en">colour</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Colour"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/EducationalInstitution">
    <rdfs:comment xml:lang="en">building(s) where instruction in one or more branches of knowledge takes
place</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.SCH"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.SCHC"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.UNIV"/>
    <rdfs:comment xml:lang="en">university</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/University"/>
            <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/College"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
    <rdfs:comment xml:lang="en">school</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/School"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/S">
    <rdfs:comment xml:lang="en">spot, building, farm, ...</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/family">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">family</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/family"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/definition">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2004/02/skos/core#definition"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

```xml
  </rdf:Description>
  <rdf:Description rdf:about="http://mid-onto.com/property/hasMemberDataTypes">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmCrewGigId">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_crew_gig_id"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCrewGig"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/birthDate">
    <rdfs:comment xml:lang="en">date of birth</rdfs:comment>
    <rdfs:comment xml:lang="en">dateofbirth</rdfs:comment>
    <rdfs:comment xml:lang="en">birth date</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/born"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:comment xml:lang="en">datebirth</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
    <rdfs:comment xml:lang="en">birthdate</rdfs:comment>
    <rdfs:comment xml:lang="en">born</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/datebirth"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/birthDate"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/dateOfBirth"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/birthdate"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/dateofbirth"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/birthDate"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/style">
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">style</rdfs:comment>
```

```
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/style"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Newspaper">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
    <rdfs:comment xml:lang="en">newspaper</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Newspaper"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Performance">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.linkedmdb.org/resource/movie/performance"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/c">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">c</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/c"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmCharacterFilmCharacterid">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_character_film_characterid"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCharacter"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shipBeam">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    <rdfs:comment xml:lang="en">the beam of a ship is its width at the widest point.</rdfs:comment>
    <rdfs:comment xml:lang="en">ship beam (m)</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/shipBeam"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Website">
    <rdfs:comment xml:lang="en">website</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Website"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
```

```xml
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/R">
    <rdfs:comment xml:lang="en">road, railroad, ...</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#R"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/type">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment xml:lang="en">type</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/type"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Golfplayer">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdfs:comment xml:lang="en">golf player</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/GolfPlayer"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/filmCrewGigFilmJob">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/film_crew_gig_film_job"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCrewGig"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/geography">
    <rdfs:comment xml:lang="en">lats</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latm"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/elevation"/>
    <rdfs:comment xml:lang="en">lon min</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longitude"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountainrange"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2003/01/geo/wgs84_pos#long"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lat"/>
    <rdfs:comment xml:lang="en">elevation min m</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
```

```
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
<rdfs:comment xml:lang="en">longs</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://www.georss.org/georss/point"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longd"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longs"/>
<rdfs:comment xml:lang="en">longitude</rdfs:comment>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdfs:comment xml:lang="en">lat</rdfs:comment>
<rdfs:comment xml:lang="en">elevation max m</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/elevationM"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lats"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latd"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Country"/>
<rdfs:range rdf:resource="http://dbpedia.org/datatype/watt"/>
<rdfs:range rdf:resource="http://dbpedia.org/datatype/foot"/>
<rdfs:comment xml:lang="en">lat d</rdfs:comment>
<rdfs:comment xml:lang="en">gemeinderatanzahl</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/northCoord"/>
<rdfs:comment xml:lang="en">longd</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longD"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/gemeinderatanzahl"/>
<rdfs:comment xml:lang="en">fläche</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longS"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latMin"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lonMin"/>
<fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/property/fl_percent_C3_percent_A4che"/>
<rdfs:comment xml:lang="en">latitude</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/long"/>
<rdfs:comment xml:lang="en">latd</rdfs:comment>
<rdfs:comment xml:lang="en">elevation m</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latM"/>
<rdfs:comment xml:lang="en">average elevation above the sea level</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lonDeg"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latitude"/>
<rdfs:comment xml:lang="en">elevation (m)</rdfs:comment>
<fiton-prop:hasMemberDataTypes
```

```
rdf:resource="http://dbpedia.org/property/gemeindeschl_percent_C3_percent_BCssel"/>
    <rdfs:comment xml:lang="en">gemeindeschlüssel</rdfs:comment>
    <rdfs:comment xml:lang="en">long m</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latDeg"/>
    <rdfs:comment xml:lang="en">lon deg</rdfs:comment>
    <rdfs:comment xml:lang="en">long s</rdfs:comment>
    <rdfs:comment xml:lang="en">latm</rdfs:comment>
    <rdfs:comment xml:lang="en">north coord</rdfs:comment>
    <rdfs:comment xml:lang="en">elevation</rdfs:comment>
    <rdfs:comment xml:lang="en">longm</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2003/01/geo/wgs84_pos#alt"/>
    <rdfs:comment xml:lang="en">lat s</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lonSec"/>
    <rdfs:comment xml:lang="en">lat sec</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/elevationF"/>
    <rdfs:comment xml:lang="en">lat min</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longM"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latS"/>
    <rdfs:comment xml:lang="en">long</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latD"/>
    <rdfs:comment xml:lang="en">long d</rdfs:comment>
    <rdfs:comment xml:lang="en">lat m</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/elevationMaxM"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/elevation"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protectedarea"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/elevationMinM"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
    <rdfs:comment xml:lang="en">elevation-f</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/longm"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
    <rdfs:comment xml:lang="en">lon sec</rdfs:comment>
    <rdfs:comment xml:lang="en">lat deg</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2003/01/geo/wgs84_pos#lat"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/latSec"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
  </rdf:Description>
```

```
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/budget">
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/philippinePeso"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/southKoreanWon"/>
  <rdfs:comment xml:lang="en">budget</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/danishKrone"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/gross"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/euro"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/icelandKrona"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/canadianDollar"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/australianDollar"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/poundSterling"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/swedishKrona"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/gross"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
  <rdfs:comment xml:lang="en">budget ($)</rdfs:comment>
  <rdfs:comment xml:lang="en">gross ($)</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/budget"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/usDollar"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/budget"/>
  <rdfs:comment xml:lang="en">gross</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/typeOfFilmCut">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/type_of_film_cut"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Stadium">
  <rdfs:comment xml:lang="en">an area, often of forested land, maintained as a place of beauty, or for
recreation</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.STDM"/>
  <rdfs:comment xml:lang="en">stadium</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Stadium"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
```

```
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shipLaunched">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">ship launched</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shipLaunched"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Politician">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdfs:comment xml:lang="en">politician</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Politician"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/2000/01/rdf-schema#range">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/endDate">
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/termEnd"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://purl.org/dc/terms/date"/>
    <rdfs:comment xml:lang="en">a point or period of time associated with an event in the lifecycle of the resource.</rdfs:comment>
    <rdfs:comment xml:lang="en">date</rdfs:comment>
    <rdfs:comment xml:lang="en">term end</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/retired"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/activeYearsEndDate"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:comment xml:lang="en">retired</rdfs:comment>
    <rdfs:comment xml:lang="en">active years end date</rdfs:comment>
```

```
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://mid-onto.com/property/hasMemberClasses">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/r">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Colour"/>
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Colour"/>
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://dbpedia.org/datatype/degreeRankine"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment xml:lang="en">r</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/r"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/producerName">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/producer_name"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Producer"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/orbitalPeriod">
    <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Planet"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    <rdfs:comment xml:lang="en">orbital period (s)</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/orbitalPeriod"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/area">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/timeZone"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/lieutenancyArea"/>
    <rdfs:comment xml:lang="en">leader name</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/mountainRange"/>
    <rdfs:comment xml:lang="en">operator</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/leaderParty"/>
```

```
<rdfs:comment xml:lang="en">owning organisation</rdfs:comment>
<rdfs:comment xml:lang="en">nearest city</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/nearestCity"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/federalState"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/wikiPageDisambiguates"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/administrativeDistrict"/>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/MountainRange"/>
<rdfs:comment xml:lang="en">province</rdfs:comment>
<rdfs:comment xml:lang="en">leader title</rdfs:comment>
<rdfs:comment xml:lang="en">time zone</rdfs:comment>
<rdfs:comment xml:lang="en">lieutenancy area</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/leaderParty"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
<rdfs:comment xml:lang="en">located in area</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/postalCodeType"/>
<rdfs:comment xml:lang="en">subdivision type</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/vorwahl"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/operator"/>
<rdfs:comment xml:lang="en">area code</rdfs:comment>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/Place"/>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/River"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/county"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/related/geo"/>
<rdfs:comment xml:lang="en">county</rdfs:comment>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdfs:comment xml:lang="en">council area</rdfs:comment>
<rdfs:comment xml:lang="en">department</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/arrondissement"/>
<rdfs:comment xml:lang="en">country</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/leaderName"/>
<rdfs:comment xml:lang="en">intercommunality</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/region"/>
<rdfs:comment xml:lang="en">leader party</rdfs:comment>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/saint"/>
<rdfs:comment xml:lang="en">federal state</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/crosses"/>
<rdfs:comment xml:lang="en">ceremonial county</rdfs:comment>
```

```
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/areaCode"/>
<rdfs:comment xml:lang="en">person that first ascented a mountain</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/firstAscentPerson"/>
<rdfs:comment xml:lang="en">city</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/location"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:comment xml:lang="en">administrative district</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/country"/>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/ceremonialCounty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/archipelago"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/councilArea"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/areaCode"/>
<rdfs:comment xml:lang="en">saint</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/subdivisionType"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/leaderTitle"/>
<rdfs:comment xml:lang="en">telephone-area</rdfs:comment>
<rdfs:comment xml:lang="en">architectual bureau</rdfs:comment>
<rdfs:comment xml:lang="en">frazioni</rdfs:comment>
<rdfs:comment xml:lang="en">crosses</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/owningOrganisation"/>
<rdfs:comment xml:lang="en">location</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/province"/>
<rdfs:comment xml:lang="en">postal code type</rdfs:comment>
<rdfs:comment xml:lang="en">vorwahl</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/principalArea"/>
<rdfs:comment xml:lang="en">wikipage disambiguates</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Bridge"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/district"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/district"/>
<fiton-prop:hasMemberDataTypes
rdf:resource="http://dbpedia.org/ontology/associationOfLocalGovernment"/>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/Saint"/>
<rdfs:range rdf:resource="http://dbpedia.org/ontology/City"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/intercommunality"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/metropolitanBorough"/>
<rdfs:comment xml:lang="en">association of local government</rdfs:comment>
```

```
  <rdfs:comment xml:lang="en">mountain range</rdfs:comment>
  <rdfs:comment xml:lang="en">region</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/locatedInArea"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Country"/>
  <rdfs:comment xml:lang="en">arrondissement</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/telephoneArea"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/architectualBureau"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/department"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Company"/>
  <rdfs:comment xml:lang="en">metropolitan borough</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/frazioni"/>
  <rdfs:range rdf:resource="http://dbpedia.org/ontology/Person"/>
  <rdfs:comment xml:lang="en">archipelago</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/city"/>
  <rdfs:comment xml:lang="en">district</rdfs:comment>
  <rdfs:comment xml:lang="en">principal area</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/governor">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/vicepresident"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/succeeded"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/before"/>
  <rdfs:comment xml:lang="en">after</rdfs:comment>
  <rdfs:comment xml:lang="en">before</rdfs:comment>
  <rdfs:comment xml:lang="en">successor</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/after"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/predecessor"/>
  <rdfs:comment xml:lang="en">president</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/president"/>
  <rdfs:comment xml:lang="en">predecessor</rdfs:comment>
  <rdfs:comment xml:lang="en">preceded</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/preceded"/>
  <rdfs:comment xml:lang="en">governor</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/successor"/>
  <rdfs:comment xml:lang="en">vicepresident</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lieutenant"/>
  <rdfs:comment xml:lang="en">lieutenant</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/governor"/>
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">succeeded</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Americanfootballplayer">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">american football player</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/AmericanFootballPlayer"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/image">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/imageMap"/>
  <rdfs:comment xml:lang="en">photo</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment xml:lang="en">photo caption</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/image"/>
  <rdfs:comment xml:lang="en">caption</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/photo"/>
  <rdfs:comment xml:lang="en">picture</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/imageFlag"/>
  <rdfs:comment xml:lang="en">logo</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/picture"/>
  <rdfs:comment xml:lang="en">image flag</rdfs:comment>
  <rdfs:comment xml:lang="en">image</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/photoCaption"/>
  <rdfs:comment xml:lang="en">image map</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdfs:comment xml:lang="en">image caption</rdfs:comment>
  <rdfs:comment xml:lang="en">image skyline</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/imageSkyline"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/logo"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/caption"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/imageCaption"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Device">
  <rdfs:comment xml:lang="en">device</rdfs:comment>
```

```
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Device"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/postalcode">
  <rdfs:comment xml:lang="en">plz</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/postalCodes"/>
  <rdfs:comment xml:lang="en">postal code</rdfs:comment>
  <rdfs:comment xml:lang="en">postal codes</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Settlement"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#postalCode"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/postalCode"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/postalCode"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/plz"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/countryType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">country type</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/countryType"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Place"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/Stream"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/River"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/River"/>
  <rdfs:domain rdf:resource="http://dbpedia.org/ontology/BodyOfWater"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
```

```xml
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/locationIdentifier">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/faaLocationIdentifier"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/icao"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Airport"/>
  <rdfs:comment xml:lang="en">faa</rdfs:comment>
  <rdfs:comment xml:lang="en">icao</rdfs:comment>
  <rdfs:comment xml:lang="en">iata location identifier</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/joule"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/icaoLocationIdentifier"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/iataLocationIdentifier"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/faa"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/nicaraguanCórdoba"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/iata"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/kelvin"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/degreeFahrenheit"/>
  <rdfs:comment xml:lang="en">icao location identifier</rdfs:comment>
  <rdfs:comment xml:lang="en">faa location identifier</rdfs:comment>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/newton"/>
  <rdfs:range rdf:resource="http://dbpedia.org/datatype/byte"/>
  <rdfs:comment xml:lang="en">iata</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/depiction">
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
  <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Image"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/depiction"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/manufacturer">
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">manufacturer</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/manufacturer"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/NytdDes">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_des"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Historicplace">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_org"/>
  <rdfs:comment xml:lang="en">historic place</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/HistoricPlace"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#S.HSTS"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/shipArmament">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment xml:lang="en">ship armament</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shipArmament"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/name">
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/agencyName"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
  <rdfs:comment xml:lang="en">location</rdfs:comment>
  <rdfs:comment xml:lang="en">title</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/alias"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/candidate"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/givenName"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/jyutpingchinesename"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/callLetters"/>
  <rdfs:comment xml:lang="en">coordinates region</rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/birthName"/>
```

<rdfs:comment xml:lang="en">district</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/thisAlbum"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/editing"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/district"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/R"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/background"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/teamName"/>
<rdfs:comment xml:lang="en">native name</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Producer"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/constituencyWestminster"/>
<rdfs:comment xml:lang="en">company name</rdfs:comment>
<rdfs:comment xml:lang="en">playername</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
<rdfs:comment xml:lang="en">artist</rdfs:comment>
<rdfs:comment xml:lang="en">subdivision type</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/stateName"/>
<rdfs:comment xml:lang="en">agency name</rdfs:comment>
<rdfs:comment xml:lang="en">civil parish</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shireDistrict"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/label"/>
<rdfs:comment xml:lang="en">name</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/simpchinesename"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
<rdfs:comment xml:lang="en">alias</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/nameLocal"/>
<rdfs:comment xml:lang="en">screenplay</rdfs:comment>
<rdfs:comment xml:lang="en">order</rdfs:comment>
<rdfs:comment xml:lang="en">birth name</rdfs:comment>
<rdfs:comment xml:lang="en">birthname</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/writer"/>
<rdfs:comment xml:lang="en">religion</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/subdivisionType"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/occupation"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Writer"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#officialName"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/country_name"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/intercommunality"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/state"/>

```
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Settlement"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/nickname"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/nick"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/story"/>
<rdfs:comment xml:lang="en">nativename</rdfs:comment>
<rdfs:comment xml:lang="en">municipality</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/nativeName"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/pinyinchinesename"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCharacter"/>
<rdfs:comment xml:lang="en">state name</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#alternateName"/>
<rdfs:comment xml:lang="en">starring</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/alternativeNames"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/order"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/nativename"/>
<rdfs:comment xml:lang="en">municipality name</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/realname"/>
<rdfs:comment xml:lang="en">distributor</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/welshName"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
<rdfs:comment xml:lang="en">subject name</rdfs:comment>
<rdfs:comment xml:lang="en">call letters</rdfs:comment>
<rdfs:comment xml:lang="en">welsh name</rdfs:comment>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
<rdfs:comment xml:lang="en">label</rdfs:comment>
<rdfs:comment xml:lang="en">othername</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/birthname"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#name"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://purl.org/dc/terms/title"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/birthName"/>
<rdfs:comment xml:lang="en">lake name</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/commonName"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/region"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/synonym"/>
<rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
```

```
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCrewGig"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/coordinatesRegion"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/cube"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/distributor"/>
<rdfs:comment xml:lang="en">gaeilge</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/residence"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/title"/>
<rdfs:comment xml:lang="en">shire county</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
<rdfs:comment xml:lang="en">common name</rdfs:comment>
<rdfs:comment xml:lang="en">shire district</rdfs:comment>
<rdfs:comment xml:lang="en">alternative names</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
<rdfs:comment xml:lang="en">a name in an official local language</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Colour"/>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/operator"/>
<rdfs:comment xml:lang="en">realname</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/title"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/gaeilge"/>
<rdfs:comment xml:lang="en">fullname</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/alias"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/gaelicName"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/screenplay"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/arrondissement"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/subdivisionName"/>
<rdfs:comment xml:lang="en">background</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/director"/>
<rdfs:comment xml:lang="en">cinematography</rdfs:comment>
<rdfs:comment xml:lang="en">music</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://data.linkedmdb.org/resource/movie/actor_name"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/fullname"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
<rdfs:comment xml:lang="en">occupation</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/religion"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/companyName"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/PersonFunction"/>
```

<rdfs:comment xml:lang="en">operator</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/producer"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/starring"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/verwaltungsgemeinschaft"/>
<rdfs:comment xml:lang="en">editing</rdfs:comment>
<rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/almaMater"/>
<rdfs:comment xml:lang="en">intercommunality</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/music"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
<rdfs:comment xml:lang="en">studio</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/name"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/artist"/>
<rdfs:comment xml:lang="en">writer</rdfs:comment>
<rdfs:comment xml:lang="en">jyutpingchinesename</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/V"/>
<rdfs:comment xml:lang="en">gaelic name</rdfs:comment>
<rdfs:comment xml:lang="en">producer</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/municipality"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/tradchinesename"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/otherName"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Director"/>
<rdfs:comment xml:lang="en">cube</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/office"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2000/01/rdf-schema#label"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/location"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
<rdfs:comment xml:lang="en">conventional long name</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/cinematography"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/narrator"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://www.w3.org/2004/02/skos/core#prefLabel"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/U"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/otherNames"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/stadiumName"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/municipalityName"/>
<rdfs:comment xml:lang="en">name local</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/lakeName"/>

```
<rdfs:comment xml:lang="en">order in office</rdfs:comment>
<rdfs:comment xml:lang="en">state</rdfs:comment>
<rdfs:comment xml:lang="en">narrator</rdfs:comment>
<rdfs:comment xml:lang="en">other name</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmFilmDistributorRelationship"/>
<rdfs:comment xml:lang="en">verwaltungsgemeinschaft</rdfs:comment>
<rdfs:comment xml:lang="en">constituency westminster</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
<rdfs:comment xml:lang="en">nickname</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/civilParish"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Performance"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/owner"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
<rdfs:comment xml:lang="en">region</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
<rdfs:comment xml:lang="en">alma mater</rdfs:comment>
<rdfs:comment xml:lang="en">synonym</rdfs:comment>
<rdfs:comment xml:lang="en">subdivision name</rdfs:comment>
<rdfs:comment xml:lang="en">pinyinchinesename</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Film"/>
<rdfs:comment xml:lang="en">stadium name</rdfs:comment>
<rdfs:comment xml:lang="en">candidate</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/studio"/>
<rdfs:comment xml:lang="en">official name</rdfs:comment>
<rdfs:comment xml:lang="en">simpchinesename</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/officialName"/>
<rdfs:comment xml:lang="en">owner</rdfs:comment>
<rdfs:comment xml:lang="en">tradchinesename</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/playername"/>
<rdfs:comment xml:lang="en">residence</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/shireCounty"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://xmlns.com/foaf/0.1/name"/>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/orderInOffice"/>
<rdfs:comment xml:lang="en">postalabbreviation</rdfs:comment>
<rdfs:comment xml:lang="en">arrondissement</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/postalabbreviation"/>
<rdfs:comment xml:lang="en">director</rdfs:comment>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/S"/>
<rdfs:comment xml:lang="en">other names</rdfs:comment>
```

```
  <rdfs:comment xml:lang="en">this album</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/subjectName"/>
  <rdfs:comment xml:lang="en">story</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/othername"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/L"/>
  <rdfs:comment xml:lang="en">office</rdfs:comment>
  <rdfs:comment xml:lang="en">team name </rdfs:comment>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/FilmCut"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/conventionalLongName"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/childrenFeatures">
  <rdfs:range rdf:resource="http://www.geonames.org/ontology#RDFData"/>
  <rdfs:comment xml:lang="en">children features</rdfs:comment>
  <fiton-prop:hasMemberDataTypes rdf:resource="http://www.geonames.org/ontology#childrenFeatures"/>
  <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/River">
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Stream"/>
  <rdfs:comment xml:lang="en">river</rdfs:comment>
  <rdfs:comment xml:lang="en">a large natural stream</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/River"/>
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
  <rdfs:comment xml:lang="en">a body of running water moving to a lower level in a channel on
land</rdfs:comment>
  <rdfs:comment xml:lang="en">stream</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#H.STM"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Officeholder">
  <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
  <rdfs:comment xml:lang="en">office holder</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/OfficeHolder"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Currency">
  <rdfs:comment xml:lang="en">currency</rdfs:comment>
  <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Currency"/>
```

```
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure">
    <rdfs:comment xml:lang="en">anatomical structure</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/AnatomicalStructure"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Mountainrange">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
    <rdfs:comment xml:lang="en">mountains</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#T.MTS"/>
    <rdfs:comment xml:lang="en">a chain of mountains bordered by highlands or separated from other mountains by passes or valleys.</rdfs:comment>
    <rdfs:comment xml:lang="en">mountain range</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/MountainRange"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/related">
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/related"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment xml:lang="en">related</rdfs:comment>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/related"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/wikiPageWikiLink">
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/ChemicalCompound"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Colour"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/wikiPageWikiLink"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/OlympicResult"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Organisation"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Website"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Beverage"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Language"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Award"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Protein"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MeanOfTransportation"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/LegalCase"/>
```

```
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Species"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Event"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Infrastructure"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Device"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Drug"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Activity"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AnatomicalStructure"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Actor"/>
    <rdfs:comment xml:lang="en">link from a wikipage to another wikipage</rdfs:comment>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Map"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Disease"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Lake"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/MusicGenre"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Currency"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Person"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/EthnicGroup"/>
    <rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Mountain"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/Basketballplayer">
    <fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_per"/>
    <rdfs:comment xml:lang="en">basketball player</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/BasketballPlayer"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/AdministrativeRegion">
    <rdfs:comment xml:lang="en">third-order administrative division</rdfs:comment>
    <rdfs:comment xml:lang="en">seat of a first-order administrative division (pplc takes precedence over
ppla)</rdfs:comment>
    <rdfs:comment xml:lang="en">section of populated place</rdfs:comment>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A.ADM1"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
    <fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPLA3"/>
    <rdfs:comment xml:lang="en">seat of a second-order administrative division</rdfs:comment>
    <rdfs:comment xml:lang="en">fourth-order administrative division</rdfs:comment>
    <rdfs:comment xml:lang="en">first-order administrative division</rdfs:comment>
    <rdfs:comment xml:lang="en">a subdivision of a first-order administrative division</rdfs:comment>
```

```
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P"/>
<rdfs:comment xml:lang="en">capital of a political entity</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A.ADM3"/>
<rdfs:comment xml:lang="en">country, state, region ...</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/Place"/>
<rdfs:comment xml:lang="en">administrative region</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPLX"/>
<rdfs:comment xml:lang="en">a primary administrative division of a country, such as a state in the united
states</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPLA"/>
<rdfs:comment xml:lang="en">populated place</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A"/>
<rdfs:comment xml:lang="en">place</rdfs:comment>
<rdfs:comment xml:lang="en">a city, town, village, or other agglomeration of buildings where people live
and work</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A.ADM2"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:comment xml:lang="en">a subdivision of a third-order administrative division</rdfs:comment>
<rdfs:comment xml:lang="en">a subdivision of a second-order administrative division</rdfs:comment>
<rdfs:comment xml:lang="en">seat of a third-order administrative division</rdfs:comment>
<rdfs:comment xml:lang="en">second-order administrative division</rdfs:comment>
<rdfs:comment xml:lang="en">a location</rdfs:comment>
<rdfs:comment xml:lang="en">city, village,...</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPL"/>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#A.ADM4"/>
<fiton-prop:hasMemberClasses rdf:resource="http://dbpedia.org/ontology/AdministrativeRegion"/>
<rdfs:comment xml:lang="en">seat of a first-order administrative
    division</rdfs:comment>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPLA2"/>
<fiton-prop:hasMemberClasses rdf:resource="http://data.nytimes.com/elements/nytd_geo"/>
<fiton-prop:hasMemberClasses rdf:resource="http://www.geonames.org/ontology#P.PPLC"/>
</rdf:Description>
<rdf:Description rdf:about="http://ri-www.nii.ac.jp/fiton/ontology/releaseDate">
<rdfs:comment xml:lang="en">first aired</rdfs:comment>
<fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/ontology/releaseDate"/>
<rdfs:domain rdf:resource="http://ri-www.nii.ac.jp/fiton/ontology/Work"/>
<rdfs:comment xml:lang="en">released</rdfs:comment>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gMonthDay"/>
<rdfs:comment xml:lang="en">release date</rdfs:comment>
```

```
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/released"/>
    <fiton-prop:hasMemberDataTypes
rdf:resource="http://data.linkedmdb.org/resource/movie/initial_release_date"/>
    <fiton-prop:hasMemberDataTypes rdf:resource="http://dbpedia.org/property/firstAired"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </rdf:Description>
</rdf:RDF>
```