Semi-automated Integration of Legacy Systems Using Linked Data

Ilya Semerhanov and Dmitry Mouromtsev[1]

ITMO University, Saint-Petersburg 197101, Russia,
ailab@mail.ifmo.ru

Abstract. A lot of valuable data is stored in standalone legacy systems inside enterprise infrastructure across different domains. It was always a big challenge to integrate such systems with each other even on structural level, but with the recent development of Semantic Web technologies its now clear that integration on semantic level could be achieved and data from different sources could be used more effectively. Its now becoming a trend to open data to the public according to Linked Data principals, but there is no common workflow that could be used with legacy systems. In this paper we propose our solution for semi-automated integration of legacy systems using Linked data principals. We analyze the motivation and current state of the art in the domain, present our method and algorithms for data extraction, transformation, mapping and interlinking. Finally we show our own implementation of the proposed solution and discuss the results.

Keywords: Semantic Web, Linked Data, Database integration, Ontology, Resource Description Framework

1 Introduction

Data integration nowadays is a big and challenging problem, due to the fact that a lot of business and personal data is stored in large amount of different computer systems: Customer Relationship Management systems, Supply Chain Management systems, Enterprise Resource Planning systems and etc. Also information is stored in web based applications such as intranet portals, blogs, personal web pages and so on. The amount of such data is growing rapidly every year and because of this industry is in need of a solution for accessing and managing distributed data. Data integration approaches can be separated in two different groups: consolidation approaches and virtualization approaches. Approaches from the first group are meant to physically transfer data from distributed data sources into one unified storage. On other hand approaches from the second group concentrate on providing a virtual way of accessing the data without any physical relocation. In this paper we understand integration according to the approach from the second group, as a unified virtual interface for accessing distributed heterogeneous data sources. But we can go with classification even further and separate virtual data integration approach into several classes: structure based and semantic based integration. The structural integration is based

on an idea that full integration can be achieved by providing a mapping between different structural parts of integrated systems. A lot of approaches and methods were developed in past years with this idea and the most popular that are used now are usually based on such technologies as Web Services, for example Service Oriented Architecture. For data representation general approach in such a case is to use a markup language, for example XML, and for data transferring - some transport network protocol, like, for example, HTTP. Thus we can say that structure based integration methods are mostly used for representing and transferring data from different systems in a common way and providing a solution for mapping one system structure to another system structure. Semantic data integration, on other hand, is dealing not only with structural mapping, but also trying to define equivalency in semantic meaning of distributed integrated data. This kind of integration, compared to the structural integration, is operating several levels above in DIKW pyramid, using not only simple data structures for integration, but also context of its usage and knowledge of how to use it [1]. This approach can bring several benefits, like for example it can increase level of automatic decision making and provide means for better analytical data search. In other words semantic integration methods can make integration tools more intelligent and help them deal more effectively with big amount of data.

This paper consists of 6 sections. Section 2 explains our motivation in this work and why we started the research. Section 3 focus on the state of the art it the area and related work. Section 4 covers our proposed solution for the problem, including method, algorithms for data integration and also implementation of it as a tool. Sections 5 and 6 concludes the paper and focus on discussion of results and future work.

2 Motivation

It is an indisputable fact that currently most of the applications and computer systems in different domains use traditional technologies such as relational databases for data storage and retrieving. From the perspective of Semantic Web they could be called legacy systems, or in other words outdated computer systems. Applying semantic integration approach to the legacy systems could take a big amount of the effort and manual work, and as a result it could be an obstacle for introduction of such approach, regardless the benefits that it could bring. There are already several solutions for mapping relational databases to Resource Description Framework (RDF) datasets available, as well as there is also a World Wide Web Consortium (W3C) R2RML language specification that intends to standardize the approaches for such mapping [2]. Nevertheless we are convinced that they all miss the unified methodology for data retrieval from different types of relational databases and publication on the Web. The level of automation in such tools is still no enough, so they require a lot of manual work through whole data integration life cycle. In this paper we present our own view on the problem and a solution for semi-automated integration of legacy systems. This solution will include the methodology and algorithms for data extraction and

transformation, as well as a prototype of a tool, that is used by us for evaluation of the results.

3 Related Work

Semantic data integration approaches are discussed for a long time by professionals in data integration domain and we can already say that the most promising solutions are based on metadata usage. Some brief survey of the current state of the art in the domain was presented by Klaus R. Dittrich and Patrick Ziegler [3]. One of the directions of development of metadata approach is closely linked to the idea to use ontology models for data integration and system interoperability. This idea is not really a new one and was already presented for example in the paper [4]. Usually there are 3 main approaches for ontology based integration: single ontology approach, multi ontology approach or hybrid approach. The idea behind single ontology is based on the assumption that its possible to create single ontology that could describe all entities and concepts in integrated systems. This is a straightforward solution that could be applied only if all systems works in one domain and it was implemented in SIMS [5]. In second approach, instead of one ontology each integrated system provide its own ontology. Compared to first solution in this case its easier to integrate new system, because ontologies could be developed independently for all integrated systems, like for example in OBSERVER [6]. But without common vocabulary its problematic to interlink concepts in different ontologies. Thats why there is a third solution that uses both ideas: for describing systems it uses several single ontologies; for interlinking them with each other global ontology is used as a vocabulary. But still there are a lot unresolved problems in this approach, for example its still not clear how to get single ontology for every system automatically and how to map them automatically with each other with the use of global ontology. The solutions for relational database mapping to RDF data sets are described in a W3C RDB2RDF Incubator Group survey [7] and in survey [8]. In this surveys several tools and techniques for data transformation are described and could be separated in three classes: direct mapping solutions, semi-automated direct mapping solutions and solutions that use domain semantic for extracting the data from relational database. Although some of them are already a mature software they still lack solutions for stable automated ontology mappings, duplicates discovery and other features that could make data integration process more friendly.

4 Implementation

4.1 Semantic integration with ontology

According to Tomas Gruber definition, ontology is an explicit specification of conceptualization [9]. It can describe concepts and relationships that can exist in one domain and basically it is a set of concepts, relations and functions that

connects them with each other:

$$O = \{T, R, F\}, where: \tag{1}$$

O – ontology of the domain;
T - set of concepts in described domain;
R - set of relationships in the domain;
F - set of functions that connects concepts and relationships inside one domain.

By using it for data integration concepts and relationships could be described in every integrated subsystem as well as in the whole subject domain. In order to do that every object that has some valuable data in the integrated systems should be described with the use of semantic metadata, in terms of one general ontology of subject domain in which all systems work. Metadata is information presented in special format that describes content of objects, it also could be called data, that describes data. In formal form we can express it as:

$$M_i = T_i \vee E_i, where: \tag{2}$$

M_i - metadata of object i;
T_i - set of concepts connected to object i;
E_i - set of concept instances in ontology.
In other words ontology of subject domain plays a role of coordinate system for all integrated applications. There are a lot of benefits of using ontology for semantic data integration:

- Instead of simple structure mapping we have a relationship mapping that describes how to use data. In other word we are moving from data level to information and knowledge level in DIKW pyramid.
- Ontology could be parsed automatically and it could provide ways for automatic decision making with the use of description logic.
- Ontology can be easily extended with new concepts and relations, thus we can integrate new subsystems without additional effort.
- One of the big advantages is that ontology can have several levels. Low level ontology could describe every individual system, middle level ontology could describe the set of integrated systems in one subject domain and high level ontology could describe integrated systems even between different domains.

In order to use such approach for integration, data that is originally stored in distributed legacy systems should be transformed to semantic friendly format first and then ontology for describing this data and systems should be created. There are numbers of methodologies for manual ontology creating, for example IDEF5, TOVE, METHONTOLOGY [10, 11]. However, there is no standard approach for semi-automatic ontology generation from legacy systems that should be later integrated, therefore we present in this paper our own solution for that.

4.2 Linked data approach

One of the good things behind ontology based semantic integration is that it can be used to integrate not only structured data, but also semi-structured and unstructured. Using Linked Data principals as the basis for integration gives us a possibility to apply the same techniques for integrating any kind of data. Linked Data describes a method of publishing data on the web so that it can be interlinked and become more useful. It is based on W3C standard technologies, such as HTTP, RDF, Web Ontology Language (OWL), and according to Tom Heath and Christian Bizer this is one of the best solutions for publishing interlinked data on the web [12]. By defining relationships between data, in the way that it can be parsed by the computers, Linked Data gives a possibility to aggregate the information from distributed data sources, create new relations, visualize connections and also extend it by connecting to other external resources. In this approach it is also possible to use ontology and Ontology Web Language for describing relations between objects in order to achieve semantic integration goal. In other words Linked Data will play a role of unified virtual interface for accessing data, stored in distributed legacy systems. The big challenge of this research was a legacy systems data extraction, transformation and load problem. In order to overcome it we developed our own method and several algorithms for ETL (Extract, Transform, Load) procedure. As the overwhelming majority of legacy systems, such as Enterprise Resource Planning systems (ERP), Customer Relationship Management systems (CRM) and custom applications, make use of relational databases to store the data we focused on this kind of data storages.

4.3 Method

In legacy computer systems, that use relational databases, data is stored in tables, where rows represent entity instance and columns represent attribute values, describing each instance. To achieve integration with Linked Data principals data should be extracted from tables and published on the Web in appropriate format. Usually in one infrastructure there are several legacy systems, that store different kind if data, but in one subject domain. That allows us to use domain ontology for describing common relations in all systems in the domain. Furthermore some general relations are independent from the domain and could be used in any computer system. We intend to combine legacy system data model, domain ontology and upper ontology in order to automatically extract data from distributed relational databases and publish it as integrated Linked Data on the Web. For this purpose we developed a method, that is based on the IDEF5 method for ontology modeling, but instead of manual ontology creation it provide steps for semi-automated ontology generation from data storage structure. This method consists of four main steps:

1. Individual RDF structure. Extracting information about data model structure of every integrated legacy system and transforming it to RDF model;
2. Common RDF structure. Combining extracted RDF models in a common RDF model;

3. Common global ontology. Creation of global ontology model on the basis of upper ontology, subject domain ontology and extracted common RDF model of legacy systems;
4. Integrated data ontology. Extracting data from distributed data sources and presenting it as semantic metadata with the use of common global ontology and automated decision making tools.

According to the method on the first step there is a primary translation of legacy system data model to the RDF model. In case of relational databases table names transforms to RDF classes, table fields to RDF properties. Then, on the second step extracted RDF models automatically combined into one common RDF model. On this step we also provide a solution for searching for similar properties and classes and providing relations between them. On the next step generated RDF model should be enriched by different concepts and relations from subject domain ontology and upper ontology, such as Friend Of A Friend ontology (FOAF). This step is not automated and should be done manually with the use of ontology editors. On the last step, based on the common global ontology, data is extracted from distributed legacy systems and described with semantic metadata. This metadata is stored together with common global ontology in OWL format and could be published on the web as Linked Data.

Provided method make use of semantic relations, described by combined ontology, between objects in integrated systems across the domain, for automated data extraction and interlinking with other data. The benefit from this approach is that there is no need for manual interlinking between extracted data, instead semantics relations between objects will be used for this purpose. Semantic relations, on other hand, will be extracted automatically from initial data model and then extended manually by the domain experts.

4.4 Algorithms

Within the proposed method we also provide several supporting algorithms that should be used for legacy systems integration:

- Algorithm for automated common RDF model extraction from legacy systems data model;
- Algorithm for data extraction from distributed legacy systems with the use of common global ontology and publishing it on the Web as Linked Data.

We also used similarity analysis procedure in our method for comparing entities in integrated systems.

Common RDF model extraction algorithm was developed in order to extract data model from each of integrated systems as RDF, and combine it together as unified RDF data model. Given there are two legacy systems LS_1 and LS_2 in one subject domain, that use relational databases for storing the data. In this case the algorithm goal is to transform LS_1 and LS_2 database structure

into common RDF model. In relational databases for description of its structure database schema is used. This schema defines the tables, fields, relationships, views and a lot of other elements, however this is only enough for integration on structural level. In order to achieve semantic integration domain ontology should be used and in this case it will add to the extracted model relations between concepts in the subject domain. The input data for proposed algorithm is database schema of each integrated systems and domain ontology.

Legacy system LS_1 is using database schema S_1 and system LS_2 – database schema S_2:

$$S_1 = \{Tb_1, \ldots, Tb_n\}, S_2 = \{Tb_1, \ldots, Tb_k\}, where: \quad (3)$$

S_1, S_2 - database schemes,
Tb_n , Tb_k - schema tables.

$$Tb_1 = \{At_1, \ldots, At_i\}, where: \quad (4)$$

At_i – table attributes.

The algorithm consists of five steps:

1. Structure mapping. Sequential mapping of S_1 and S_2 into RDF format.

$$Tb_{\mathrm{n}} \rightarrow T_{\mathrm{m}}, Tb_{\mathrm{k}} \rightarrow T_{\mathrm{m}}, At_{\mathrm{i}} \rightarrow A_{\mathrm{i}}$$

   where T_m - RDF classes, A_i - RDF properties.
2. Automatic relations creation. Providing semantic properties P_j , by automatic similarity analysis of database structure. Analysis is based on several measures: data types similarity, database names similarity, string similarity.
3. Enrichment. Subject domain ontology and upper ontology import with the use of OWL property *owl:import*.
4. Manual relations creation. Editing of extracted model with the ontology editor. Manual creation of relations between concepts from upper and domain level ontologies and objects in extracted model.
5. Output of created common RDF model into file or in RDF triple store.

The output is a common RDF model that describe objects and their relations in legacy systems, as well as concepts and their relations inside subject domain.

Data extraction algorithm was designed for data extraction from distributed legacy systems with the use of previously generated common RDF model, and for later publishing on the Web. The input data for this algorithm is common RDF data model and list of tables from which data should be extracted. Suppose in input there are i number of tables with data, then:

$$Tb_i = \{V_1, \ldots, V_n\}, where: \quad (5)$$

Tb_i - table database i;
V_n - table entry.

The algorithm consists of five steps:

1. Import of common global model. Import of extracted earlier common global RDF model.
2. Entries extraction. Extraction of every V_n entry from table Tb_i in each integrated database.
3. Similarity analysis. Similarity analysis of extracted entries with each other. If there is a match, one of the tree semantic properties should be applied: *skos:closeMatch*, *skos:narrowMatch*, *skos:exactMatch*.
4. Reasoning. Creation of new semantic properties by logical reasoning, that is working because of description logic, which were imported from common global model.
5. Output of created integrated data ontology in OWL format in a file or in RDF store.

The result of the algorithm is global meta model that contains objects and its relations in the subject domain and also interlinked data in RDF format.

Similarity analysis was done in both algorithms in the method. In common RDF model extraction algorithm it is used for automated creation of relations between objects in different integrated systems. In data extraction algorithm it is used for comparing extracted string values.

For automated creation of relations between objects in different systems during common model extraction, we propose to use complex method of comparison by several parameters:

- string comparison of elements names and descriptions, for example attribute names in tables of relational database.
- comparison by data type. Different relational databases use different data types, however in the research we created common mapping between all relational databases data types and XML Schema data types.
- corpus-based comparison of elements. By using subject domain ontology as thesaurus, we can find relations between objects during extraction.

For string similarity comparison there are several approaches available, for example Jaccard similarity coefficient, Tanimoto similarity coefficient, Levenshtein distance or Sorensen-Dice similarity coefficient [13]. For all approaches we propose to use it not against a single character in a string or a whole line, but against a string, separated in intersecting N-grams. N-gram is a contiguous sequence of N items from a given sequence of text and it can be of arbitrary length. We propose to select length dynamical, based on the length of the initial string and use w-shingling for tokenizing it in N-grams. For example for Sorensen-Dice similarity coefficient, that originally looks like:

$$p = \frac{2 \cdot |X \wedge Y|}{|X| + |Y|}, where: \tag{6}$$

p – similarity coefficient,
X, Y – number of characters in string X and Y.
The solution with N-grams and Sorensen-Dice coefficient should look like:

$$p = \frac{2 \cdot |Ngrams(X) \wedge Ngrams(Y)|}{|Ngrams(X)| + |Ngrams(Y)|}, where: \tag{7}$$

p – similarity coefficient,
Ngrams(X) – function calculating N-grams sets from string X.

Every parameter, that is used for complex comparison has its own weight. In the end every compared element could be described as a set of parameter similarity coefficient and its weight:

$$sim(S, E) = \{p_1 w_1, \ldots, p_k w_k\}, where: \tag{8}$$

sim(S,E) – final similarity between element S and E in integrated system,
p_k – similarity coefficient of parameter k,
w_k – weight of parameter k.
In real implementation of the algorithms the final similarity value should be computed as a leaner function or with some more effective approach, for example linear regression [14].

4.5 Semantic data integration tool

For practical implementation and testing of our method and algorithms we developed a prototype of semantic data integration tool. The high-level architecture of the tool, called R2RMapper, is illustrated on figure 1. The tool consists of four main modules: R2RMapperCore, R2RMapperWi, R2RMapperBatch and MatchingDB. R2RMapperCore is the main module that is used for data extraction from distributed legacy systems. It can work like a standalone library or inside R2RMapper tool. Communication with integrated data sources is achieved by web services or by direct JDBC connection. R2RMapperWi module is a web frontend that gives access for the user to the main features. R2RMapperBatch is a module for scheduling of different tasks, for example nightly synchronization with integrated systems. MatchingDB is a Redis based memory caching mechanism that caches different information, such as similarity analysis results, in memory during ontology extraction. The extracted OWL ontology is stored as RDF in Jena TDB storage from where it is published by Jena Fuseki server on the web as a linked data cloud. This linked data cloud works as a virtual interface, with which client systems or users can access data in distributed legacy systems by SPARQL queries. As the linked data is backed up by the real ontology with the sets of concepts and relations, all benefits of ontology based integration are provided.

For testing of the tool we used a CentOS 6.2 server with 16GB RAM and Intel i5 processor 3.10 GHz. We executed it against Oracle 11g database with 50 000 entries. The results of the execution are presented on figure 2. Due to use

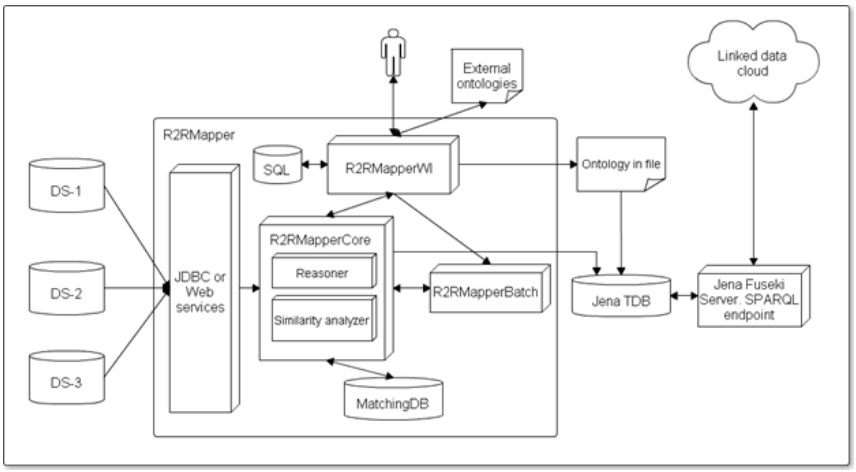

Fig. 1. R2RMapper architecture

of Redis as a caching mechanism we managed to decrease time for extraction of data from relational databases two times.

As a result of integration, user can work with integrated data through one endpoint and more effectively. For example there is a possibility to do an analytical semantic search query which will also include in search results related information about the required resource. Thus we can say that by using such approach users can work not only with the integrated data, but with integrated semantic knowledge of how to use this data.

5 Discussion

Today providing of an easy access to the big amount of stored data is becoming a trend, no matter whether it is inside one organisation infrastructure or distributed between different organisations and domains. It is equally important for big enterprise companies, open communities and research centers to open the data and make it available for other internal systems, external clients and of course users. In the research we were developing a unified method that could be applied to every domain for data extraction, transformation, publication and integration. The main goal is to automate the process as much as possible by using hybrid multilevel ontology approach, natural language processing and machine learning for automated decision making during integration. Although the method and its implementation is an already working solution it still has a room for improvement, for example in the area of semi-structured and unstructured data extraction. We also attempt to make global ontology extraction procedure more convenient for the users and domain experts.

As one of the first big practical use cases of presented method we intend to develop a platform for publishing of open science data from universities as

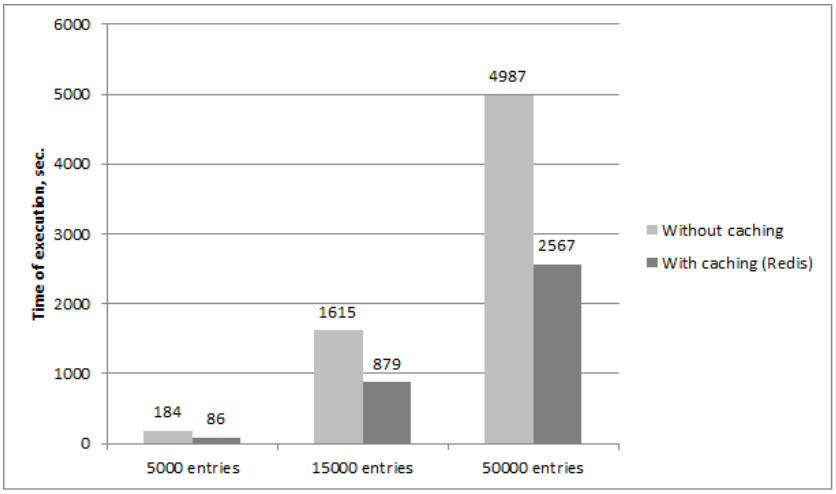

Fig. 2. Time of data extraction with R2RMapper tool

Linked Data. The first candidates for opening data by this platform are ITMO University and Leipzig University.

6 Conclusions and future work

In the presented paper we discussed semantic data integration approach, based on ontology usage and linked data technology. We showed the main benefits of this idea and presented our solution for providing distributed data in a semantic friendly linked data format. We also presented a developed prototype of a semantic data integration tool, which implements our solution. Our method currently works only for the structured data, but we are working on an extension of this method in the direction of working with semi-structured data. In future work we plan to research possibility of applying our solution also for unstructured data in order to parse it and represent as Linked Data.

In collaboration with AKSW [1] group from Leipzig we intend to improve the solution and use it in a generic platform for open science data integration, knowledge acquisition and collaboration beteen universities. Together with legacy system integration using ETL procedures this platform will be used for integration of existing RDF datasets by the use of SPARQL federated queries. This topic will be discussed in our future papers.

[1] http://aksw.org/

References

1. Rowley J. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science*, 33(2):163–180, 2007.
2. R2RML: RDB to RDF mapping language. http://www.w3.org/TR/r2rml/. Accessed: 2015-03-20.
3. Dittrich K. R. Ziegler P. Three decades of data integration - all problems solved? *World Computer Congress - IFIP*, pages 3–12, 2004.
4. Visser U. Stuckenschmidt . Wache H., Vogele . Ontology-based integration of information - a survey of existing approaches. *IJCAI-01 proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 4–10, aug 2001.
5. Yigal Arens, Chun-Nan Hsu, and Craig A. Knoblock. Readings in agents. chapter Query Processing in the SIMS Information Mediator, pages 82–90. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
6. Eduardo Mena, Arantza Illarramendi, Vipul Kashyap, and AmitP. Sheth. Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.
7. Sebastian Hellmann Kingsley Idehen Ted Thibodeau Jr Sren Auer Juan Sequeda Ahmed Ezzat Satya S. Sahoo, Wolfgang Halb. A survey of current approaches for mapping of relational databases to RDF. Technical report, W3C RDB2RDF Incubator Group, jan 2009.
8. Catherine Faron-Zucke Franck Michel, Johan Montagnat. A survey of RDB to RDF translation approaches and tools. Under review in Semantic Web Journal, 2014.
9. Gruber T. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
10. Perakath C. Benjamin, Christopher P. Menzel, Richard J. Mayer, Florence Fillion, Michael T. Futrell, Paula S. deWitte, and Madhavi Lingineni. *IDEF5 Method Report*, September 1994.
11. Uschold M. King M. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing.*, 1995.
12. Bizer C Heath T. Linked data: Evolving the web into a global data space (1st edition). *Synthesis Lectures on the Semantic Web: Theory and Technology*, pages 1–136, 2011.
13. Safa'a I. Hajeer. Comparison on the effectiveness of different statistical similarity measures. *International Journal of Computer Applications*, (8), 2012.
14. Anhai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to map between ontologies on the semantic web. *WWW-02 Proceedings of the 11th international conference on World Wide Web*, pages 662–673, 2002.