Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING DBPEDIA QUALITY USING MARKOV LOGIC NETWORKS

[1] MAHMOUD ALI, [2] MOHAMMED ALCHAITA

[1]Department Of Web Sciences, Syrian Virtual University, Syria
[2]Department Of Web Sciences, Syrian Virtual University, Syria
Email: [1]mahmoud_57942@svuonline.org, [2]t_malchaita@svuonline.org

ABSTRACT

The linked data in the Web may be incomplete due to be extracted from semi-structured sources such as Wikipedia, or unstructured such as text. There are various approaches aim to complete the missing data in the linked data sets, including the statistical distributions of properties and types for enhancing the quality of incomplete and noisy Linked Data sets, which obtained good results.

In this study, we suggest using of Markov logic networks to improve the quality of the unstructured Linked Data sets without using any external knowledge. Markov Logic Networks (MLNs) is considered one of the most known and proposed methods in the field of Statistical Relational Learning (SRL). It is a first order knowledge base with attaching a weight to each formula. Markov Logic Networks generalize the First - order Logic and attach a weight to each equation. Therefore, we rely on RDF(S) and its associating entailment rules which provide a data representation model. We carry out reasoning by transforming the statements and constraints to Markov Logic and compute the most probable consistent state with respect to the defined constraints.
Results showed that the proposed algorithm could infer on correct types which the algorithm SDType couldn't. As well, the results were remarkably improved at increasing the number of the steps.

Keywords: *Dbpedia, Type Completion, Markov Logic Network, Knowledge Graph, First Order Logic.*

1. INTRODUCTION

The development and standardization of the Semantic Web technologies led to unprecedented volume of the data published on the Web as Linked Data (LD). However, it is noticeable that the quality of the data is widely disparate, ranging from extensively structured data sets to relatively low quality extracted data [1].

The Data set which published on data Web covers various group of fields such as Media, geography, life science and government…etc. However, the data on the web reveal a great disparity in the quality of the data. The data extracted from semi-structured sources such as DBpedia often contain contradictions and incomplete and noisy information [2, 3].

Many of the data sets published as linked data on the web have been constructed from structured sources as the relational database, therefore they enjoy strong structure [4]. As well, the linked data are extracted from semi-structured sources such as Wikipedia [3] or from unstructured such as text [5].

The linked Open data (LOD) consists of massive volume of the structured data published on the web. This data set is of disparate quality [6] since the linked data sets which have been extracted from the relational database contain the information type of each source. This information is existed in most of the database. This is not applicable to the data sets which have been extracted from semi-structured or unstructured sources and whose data is most probably incomplete and noisy to some extent. Those information may be either missing in the original source or the system of extracting information is unable to extract, therefore the quality of the data in the linked data in considered as object for criticizing by Pedantic web group [7].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. IMPORTANCE OF RESEARCH

Statistical relational learning techniques have not been used previously to infer missing types in knowledge bases whether DBpedia or other famous knowledge bases such as NELL and Freebase.

Th research aims to improving DBpedia using Markov logic networks. Statistical relational learning techniques reveal the semantic relationships in the available data. This information is used to predict missing statements.

3. DATA QUALITY FROM THE PERSPECTIVE OF THE APPROACH APPLIED FOR BUILDING THE KNOWLEDGE BASES

Completeness, accuracy, and data quality are important parameters that determine the usefulness of knowledge bases and are influenced by the way knowledge bases are constructed.

The Knowledge bases construction methods are classified into four main groups:

a. Curated approach: The triples are created manually by a closed group of experts.
b. Collaborative approach: The triples are created manually by an open group of volunteers.
c. Automated semi-structured approach: The triples are extracted automatically from semi-structured text (e.g., Infoboxes in Wikipedia) via hand-crafted rules or regular expressions.
d. Automated unstructured approach: The triples are extracted automatically from unstructured text via techniques of natural language processing (NLP) and machine learning.

The Table 1 clarifies the approaches applicable in constructing the most famous Knowledge base.

Table 1: The Approaches Applicable in Constructing the Most Famous Knowledge Base.

Applied Approach	Knowledge Bases
Curated approach	Cyc/OpenCyc, WordNet
Collaborative approach	Wikipedia, freebase
Automated semi-structured approach	YAGO, DBpedia, Freebase.
Automated unstructured approach	Knowledge Vault, NELL

Construction of knowledge bases by the Curated approach leads to highly accurate results, but this technique is not considered as scale well due to its dependence on human experts. While the Collaborative knowledge base construction, which was used to build Wikipedia and Freebase, is considered better but still has some limitations [8].

4. DATA QUALITY IN THE KNOWLEDGE BASES AND LINKED DATA SETS

Data Quality is not a unique scale but it is multi–dimensional including relevance, completeness and accessibility of the data. These dimensions have a varying importance depending on the its context, where the data quality is clarified by its" fitness for use", i.e. the Capability of the data to fit with requirements of a specific user by giving him a certain using case.

The Linked data sets which are constructed from semi-structured or unstructured sources is facing some problems of Data Quality which characterizing that type of the data sets. The first difference is related to completeness of the type information because of missing of type information in the semi-structured or unstructured sources or to the mistakes occurring in the process of information extraction, where the type information is usually missing in the part linked with the descriptive sources [9].

Logically, completeness in the linked data is not a problem due to the official references of RDF. The Outer world allows the existence of the missing information, but to have a complete information about the type is necessary in many using cases, for example, inquiring about all the cities of one country will give useful results if there is sufficient number of the cases which have that type Dbpedia-Owl:City.

The data Sets which are constructed from semi-structured or unstructured data sets may contain noisy. The relational database may also contain mistakes but they are usually realistic mistakes such as wrong capital or population of a country.

On contrary, the noise which occurs in the process of information extraction usually contains different types of mistakes such as building or person is identified as a capital of a country.

In recent years, DBpedia has become one of the central data sets and the most widely usable in the Linked open data cloud according to its wide fame and comprehensiveness [4].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Although the knowledge base DBpedia provides extensive coverage but it is not free from mistakes. There are various sources for these mistakes ranging from realistic mistakes in Wikipedia resulted from the misuse of Wiki marks, as using the wrong types of InfoBox and to mistakes and weakness in the programming instructions of DBpedia data extractor. Wikipedia, itself, has mistakes.

Researchers in the Study [10] have carried out an analysis for another dimension of the quality in the Knowledge base DBpedia which is the "Completeness". But making an evaluation for the information completeness in DBpedia is difficult and may be impossible. And they tried to be close to the completeness with one specific type of the information in the Knowledge base DBpedia, we mean the direct types, they estimated number of the missing type statements in DBpedia to at least 2.7 million.

According to the design of the Knowledge base DBpedia, it cannot contain any information not available in Wikipedia abut it is allowed to provide ontology with the data to inference the missing information theoretically.

5. STATISTICAL RELATIONAL LEARNING

Many of the data sets in the real world are considered relational data sets and most of the real-world applications have embodied by existence of a complex and unspecific relational structure. Where distribution of the data is not consistent and independent.

The relational data consists of many kinds of entities. Each entity is revealed through different group of descriptors. Therefore, search fields of Statistical Relational Learning (SRL) has appeared in an attempt to show model and learning within relational field.

The Statistical Relational Learning is considered a new branch of machine learning which is trying to design a model for joint distribution through the relational data. It gathers between the statistical learning which treats uncertainty in the data and the relational learning which deals with the complex relational structures. A statistical relational model for a given database shows not only the correlations between attributes of each table, but also dependencies among attributes of different tables. The Statistical Relational Learning are usually represented with Graphical models which are considered distinct in the methods of their representation, learning and inference [11].

The Statistical Relational Learning is identified by name of Probabilistic Inductive Logic which deals with machine learning and searching for the data within relational fields. As well, it cares about with one of the most important issues in the Artificial intelligence which is combining the Probabilistic logic with machine learning and First Order Logic and relational representation. It deals with all their aspects such as Structure learning and Parameters Estimation. i.e. The statistical relational learning is combining between the logic, probability and learning (see Figure 1) [12].

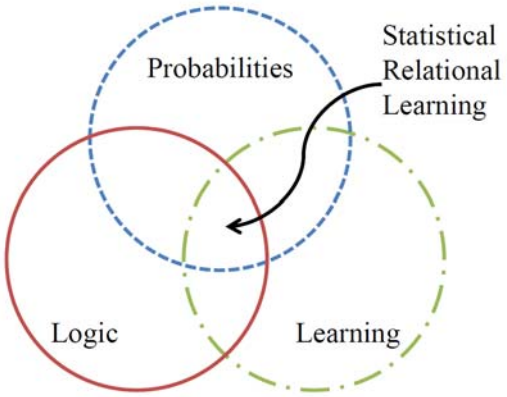

Figure 1: Statistical Relational Learning.

6. KNOWLEDGE GRAPHS

In this study, we will not depend on the traditional machine learning algorithm which deal with data matrix, where each row represents an object characterized by a feature vector of attributes, and where the main tasks are to learn a mapping from this feature vector to an output prediction of some form, or to perform unsupervised learning like clustering.

While representation of an object in the statistical relational learning may contain its relations with other objects, consequently the data may be in form of a graph, consisting of nodes (entities) and labelled edges (relations between the entities). The Knowledge graphs give organized semantic information interpretable by the computers. This characteristic is considered an important factor to build more intelligent machines, so the Knowledge

graphs have many applications on the huge data in a various group of commercial and scientific fields.

Out of the main tasks of the Knowledge graphs are:

a. Link prediction: It is also known as Knowledge graph completion which is a prediction by existence of edges (relations) in the graph or prediction with the probability of the correct type of the edges in the graphs. The link prediction is important because the existing knowledge graphs usually miss many facts and some edges may be wrong.

It has been revealed that the relational samples which take the entities relations into consideration can be significantly surpass the non-relational learning means for this task [13].

b. Entity resolution: It is also known as deduplication or object identification. It is a matter of specifying which objects in the relational data refer to the main entities themselves.

c. Link-Based Clustering: It groups together objects that have similar characteristics based on their own attributes and more importantly, the attributes of their links. i.e. the entities are not grouped according to their similarity only, but also according the similarity of their links [14].

The information in the knowledge graphs are formulated in form of entities and relations between them. This kind of knowledge representation has a long history in the Logic and artificial intelligence [15].

It has been used in the semantic web in order to create web of readable data via the machine [16]. This vision for the semantic Web is not completely achieved, only part of it.

7. MARKOV NETWORKS

It is also known as Markov random field. It models the joint distribution for a set of variables $X = (X_1, X_2, \dots, X_n)$. It consists of an undirected graph G and a set of potential functions ϕ_k for each clique k. Each variable is assigned to a node so that the joint distribution, which depends on the state of its cliques $x_{\{k\}}$, is given by

$$P(X = x) = \frac{1}{Z}\prod_k \phi_k\left(x_{\{k\}}\right) \qquad (1)$$

with the partition function $Z = \sum_{x\epsilon X}\prod_k \phi_k(x_{\{k\}})$. In log-linear models [17], the clique potential is replaced by an exponentiated weighted sum of the features of the state. A feature may be a real-valued function but it is also possible to define binary features, i.e. $f_i \in \{0,1\}$:

$$P(X = x) = \frac{1}{Z}\exp\left(\sum_j w_i f_i(x)\right) \qquad (2)$$

One key feature of the Markov Network is that each node is independent from all others given its neighbors (i.e., its Markov blanket). This enables efficient inference algorithms. A Markov Logic Network relies only on binary features.

8. FIRST ORDER LOGIC

The first-Order logic allows construction of knowledge bases on basis of formulas which use the constants, variables, functions, and predicates. Constants represent objects in the domain of interest, variables range over the objects of the domain, functions map tuple of objects to (other) objects, and predicates model the relations between the objects. The constants and the variables may be typed, in such case variables range only over objects of the corresponding type, and constants can only represent objects of the corresponding type [18]. For example: Variable X can range on people "Mohammad, Basel, George", and the constant C can represent "Damascus" City. Based on this, the following building blocks are defined:

a. A term is an expression representing an object in the domain. It can be a constant, a variable, or a function applied to a group of terms. Example: x, Basel, $\mathrm{GreatestCommonDivisor(x, y)}$.
b. An atom (atomic formula) is defined as a predicate symbol applied to a tuple of terms.
c. A positive literal is a non-negated atom, a negative literal is a negated atom.
d. A formula is recursively constructed from atomic formulas using logical connectors as shown in the *Table 2*.

Moreover, a ground term is a term that does not contain variables, i.e., all variables are replaced by constants (grounding), and a ground atom is an atomic formula that has only ground terms. All formulas of a first-order logic knowledge base are implicitly conjoined, which leads to the requirement that a possible world must assign a positive truth value to each ground term. A possible world represents a truth assignment to each possible ground atom. We skip the rules of existential

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

quantified formulas as they are not required in the context of this work. We do also not include functions as we focus on function free first-order logic.

Table 2:Logical Connectors.

Logical connective	Example
Conjunction	$F1 \wedge F2$
Disjunction	$F1 \vee F2$
Implication	$F1 \Longrightarrow F2$
Equivalence	$F1 \Leftrightarrow F2$

9. MARKOV LOGIC NETWORKS

It is one of the most famous and proposed method in the field of statistical relational learning. The Markov Logic Networks (MLNs) combine between the probability logic and the relational logic. So, it expands Synthetically the First-Order Logic and attach a weight for each formula, but semantically it can represent the probability distribution over the possible worlds by using formulas and their corresponding weights, then the world which violates fewer formulas, it has a probability bigger than Zero. As well, Markov Logic Network is considered as a template of ordinary Markov Network [19].

Markov Logic Network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number and a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$. it defines a Markov network $M_{L,C}$ (Equations 1 and 2) as follows:

a. $M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground atom is true, and 0 otherwise.

b. $M_{L,C}$ contains one feature (i.e. an edge) for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with F_i in L.

The worlds are considered as an investment for the truth values of all probabilistic ground parts. Each stat in Markov Network represents a possible world. As well, the ground atoms which appear together in positive formula are linking by an edge(relation), its weight linking with the weight of the formula. Therefore, Markov Logic Network MLN is considered as a template of ordinary Markov Network depending on its definition and equation. The probability distribution over possible worlds x specified by the ground network is calculated by the following equation:

$$P(X = x) = \frac{1}{Z} exp\left(\sum_{i=1}^{F} w_i n_i(x)\right) \qquad (3)$$

Where F is number of the formulas in MLN and $n_i(x)$ is the number of true groundings of F_i in x, therefore the world which violates some constraints can still have a high weight. Contradictory formulas can be resolved by considering their weights. Moreover, increasing the weights to infinite makes the MLN a purely logic knowledge base [20].

To learn in Markov Logic Network, there are set of methods, the most famous one is Log-Likelihood learning which is considered a good choice for the small examples but it is not considered efficient for the Truth data and actual problems. The best choice is Pseudo-Likelihood Learning. But if the formulas prevailing in the model are of conjunction form, the best measure is Pseudo-Likelihood Learning (with Custom Grounding) because it treats the conjunction in a linear time almost instead of exponential.

The inference goes through two main phases in Markov Logic Network MLNs. In the first phase, the minimal subset is determined in the ground Markov network and many of predicates independent of the query predicates are eliminated in this phase. The result is that the inference is performed in the minimal Markov Network. In the second phase, the inference is performed in Markov Networks by using Gibbs Sampling, where the evidences nodes are observed and set to their values, then the variables which are invisible in the network are randomly conditioned and arranged. This process will be repeated for all the variables [11].

The basic inference task in Markov Logic is to determine the most probabilistic world due to some evidences. This task is called Maximum A-Posterior (MAP) Inference and it is equivalent to determining the world which increases the total of the weights.

$$\arg\max_{x} f(X = x | E = e) \qquad (4)$$

Where E is evidences variables, which is called the observed(clear) variables. Whereas the remaining

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

variables *X* is called the hidden variables. Consequently, the task of the query engine in Markov Logic is inference for the variable *x* which leads to the maximum probability [19].

One of the most famous methods of inference is Enumeration-Ask, performs exact inference by enumerating all possible worlds that are consistent with the evidence *E*. As well, it is considered a good choice for the smallest reasoning problems but it is not considered efficient for the Truth data and actual problems, and of the well- known efficient algorithm MC-SAT which do approximate inference.

10. RELATED WORK

The aim of type prediction is inference for all the type clauses in a certain case. Where there are many solutions for type inference for dataset RDF by using techniques such as machine learning, statistical methods and use of external knowledge as links for other data resources or text information. There are many approaches aiming to specify the wrong and missing data in the linked data sets, and of these emerging approaches which have been applied to Dbpedia.

One of the first approaches to type classification in relational data has been discussed by Neville and Jensen (2000). The authors train a machine learning model on instances that already have a type, and apply it to the untyped instances in an iterative manner. The authors report an accuracy of 0.81, treating type completion as a single-class problem (i.e., each instance is assigned exactly one type) [21].

Giovanni et al. (2012) exploit types of resources derived from linked resources, where links between Wikipedia pages are used to find linked resources. As DBpedia only exploits links within Wikipedia infoboxes. For each resource, they use the classes of related resources as features, and use k nearest neighbors for predicting types based on those features. The authors report a recall of 0.86, a precision of 0.52, and hence an F-measure of 0.65, on DBpedia [22].

Sleeman and Finin (2013) try to predict the type of instances, given the attributes of that instance. They use a labeled training set for training an approach for type prediction. They evaluate their approach on three example classes (Person, Place, and Organization) on Freebase, reporting an F-measure near 1.0 for places, while the F-measure for persons and organizations is around 0.6 [23].

Pohl (2012) addresses a slightly different problem, i.e., the mapping DBpedia resources to the category system of OpenCyc. They use different indicators – infoboxes, textual descriptions, Wikipedia categories and instance-level links to OpenCyc – and apply an a posteriori consistency check using Cyc's own consistency checking mechanism. The authors report a recall of 0.78, a precision of 0.93, and hence an F-measure of 0.85 [24].

Aprosio et al. (2013) introduce an approach which first exploits cross-language links between DBpedia in different languages to increase coverage, e.g., if an instance has a type in one language version and does not have one in another language version. Then, they use nearest neighbor classification based on different features, such as templates, categories, and bag of words of the corresponding Wikipedia article. On existing type information in DBpedia, the authors report a recall of 0.48, a precision of 0.91, and an F-measure of 0.63 [25].

The approach we examined in Paulheim (2013) depending on statistical distributions, which use the features that link between two sources as indictors for their types, and the main idea is using all the internal and external features of a source as indicator for type of this source. The statistical distribution of the type is used for a feature in the object site and subject of the feature in order to figure out the types of the case. As well, SDType can be considered as a way for the weighed voting since each feature can be voted for its objects types by using the statistical distribution of the weight of the votes [9, 10].

In this study, we suggest using of Markov logic networks to improve the quality of the unstructured Linked Data sets without using any external knowledge. Markov Logic Networks (MLNs) is considered one of the most known and proposed methods in the field of Statistical Relational Learning (SRL). It is a first order knowledge base with attaching a weight to each formula. Markov Logic Networks generalize the First - order Logic and attach a weight to each equation. Therefore, we rely on RDF(S) and its associating entailment rules which provide a data representation model. We carry out reasoning by transforming the statements and constraints to Markov Logic and compute the most probable consistent state with respect to the defined constraints, and we evaluate the proposed approach in order to show its practicality and flexibility and then compare with the results of Statistical distribution algorithms of the features and types in order to complete the type SDType.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

11. MEHODOLOGY AND PROPOSED METHOD

The system is treating the dataset extracted from DBpedia in order to build Model and Evidence in form of logical expressions of first order logic, then the treatment is by the programming tool ProbCog which is included in the system. This tool is supporting the inference algorithm MC-SAT which we will use in this research, then treatment of the tool output in order to complete the statements of the missing type.

10.1 Building of The Model for Type Completion System (MlnType)

The most important step in building a successful model. The model consists of three main parts (see Figure 2):

a. Domain declarations: this part includes the domains of the classes and properties which are extracted from the Ontology file of the Knowledge base DBpedia, it is as "rdfs:Class = {dbo[1]:Agent, dbo:Person,…, dbo:Place}".

b. Predicate declaration: this part includes the predicates used in the whole system, which are extracted from RDF(S) vocabulary and transformed into First order logic expressions, for example, a type predicate is expressed by rdf: type(rdfs: resource, rdfs: Class). The Table 3 shows the predicates used in the Type completion system corresponding the version 3.8 of DBpedia.

Table 3: The Predicates Used in Type Completion System.

Predicates
rdf:type(rdfs:Resource, rdfs:Class)
rdfs:subClassOf(rdfs:Class, rdfs:Class)
owl:equivalentClass(rdfs:Class, rdfs:Class)
owl:equivalentProperty(rdf:Property, rdf:Property)
rdfs:domain(rdf:Property, rdfs:Class)
rdfs:range(rdf:Property, rdfs:Class)
domainRes(rdf:Property, rdfs:Resource)
rangeRes(rdf:Property, rdfs:Resource)

c. Rules and Constraints: There are two kinds of constraints in Markov Logic networks, hard constraints which are not allowed to be violated in any way, where Markov Logic Network calculates weights allow of non-violation of these rules. The other kind of the constraints is soft constraints, in which each rule has a weight. The soft constraints are preceded by this weight, while the hard constraints are terminated by a point (".").

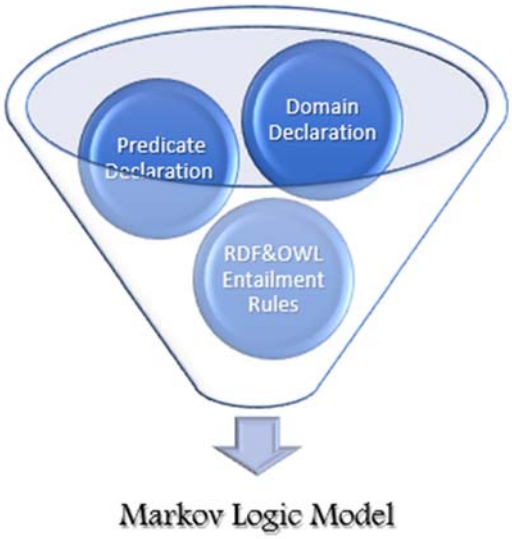

Figure 2: Main Parts of The Model.

The rules which are in conformity with the state of Knowledge base DBpedia 3.8 are included and formed from two sets which are RDFS Entailment rules[2] and OWEL LD entailment rule set[3].

Since DBpedia contains noise and at the same time it has been built in conformity with the standard RDF(S), therefore the rules have been used as hard constraints. The use of rules as soft constraints, it means the permission to violate these rules which leads to increase of inconsistency of the results. Noting that the world which violates few number of the constraints has a probability more than Zero. Table 4 shows the used rules after being transformed into first order logic.

[1] The prefix **dbo** denotes for classes which start with http://dbpedia.org/ontology

[2] https://www.w3.org/TR/2004/REC-rdf-mt-20040210/#RDFSRules

[3] http://semanticweb.org/OWLLD/#Rules

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: The Rules Used in The Model Relate to The Version Dbpedia 3.8 Of Type Completion System.

Rule Set	Formulas
RDFS	domainRes(a, y) ^ rdfs:domain(a, x) => rdf:type(y, x).
RDFS	rangeRes(p, o) ^ rdfs:range(p, c) => rdf:type(o, c).
RDFS	rdfs:subClassOf(x, y) ^ rdf:type(z, x) => rdf:type(z, y).
RDFS	rdfs:subClassOf(x, y) ^ rdfs:subClassOf(y, z) => rdfs:subClassOf(x, z).
OWL	owl:equivalentProperty(a, b) ^ domainRes(a, x) => domainRes(b, x).
OWL	owl:equivalentProperty(a, b) ^ rangeRes(a, y) => rangeRes(b, y).
OWL	owl:equivalentProperty(a, b) ^ domainRes(b, x) => domainRes(a, x).
OWL	owl:equivalentProperty(a, b) ^ rangeRes(b, y) => rangeRes(a, y).
OWL	owl:equivalentClass(x, y) ^ rdf:type(z, x) => rdf:type(z, y).
OWL	owl:equivalentClass(x, y) ^ rdf:type(z, y) => rdf:type(z, x).
OWL	owl:equivalentClass(x, y) => rdfs:subClassOf(x, y) ^ rdfs:subClassOf(y, x).
OWL	rdfs:subClassOf(x, y) ^ rdfs:subClassOf(y, x) => owl:equivalentClass(x, y).

10.2 Building of Evidence for Type Completion System (MlnType)

Building of the data should be in a way not violating the definitions existed in the model and getting benefits from the ontology file of the knowledge base DBpedia. This file is in formula XML-Rdf. The system is analyzing this file and extracting the important structures from the ontology file to be transformed into first order Logic expressions and used in the Markov Logic Networks by ProbCog.

In addition to the domains which are extracted from the ontology file for building the model, the following information is extracted for building data of the Evidence from the ontology file of the version 3.8 of knowledge base DBpedia:

a. The hierarchy of the classes through expressions rdfs:subclassOf, where these expressions are transformed into First order logic and in conformity with the their definition in the model to be as form rdfs:subclassOf(rdfs:class, rdfs,class).

b. The Equivalent properties with transforming them into first order logic and in conformity with the their definition in the model to be as form owl:equivalentProperty(rdf:Property,rdf:Property).

c. The Equivalent Classes with transforming them into first Order logic and in conformity with the their definition in the model to be as form owl:equivalentClass(rdfs:Class,rdfs:Class).

d. The ontology file contains definitions of all the relations used in the properties file through expressions rdfs:range, rdfs:domain for each property . The property rdfs:domain clarifies the class used in the subject position for a relation and the property rdfs:range clarifies the class used in the object position for a relation. The relations are extracted from the data set of the properties in a similar way in which the definition of the relations is extracted from the ontology file. i.e. we follow the principle of separation between the object and the subject and we define two vocabularies for this aim in the model within the part related to the predicates definition. These two vocabularies are:

domainRes(rdf:Property, rdfs:Resource)

rangeRes(rdf:Property, rdfs:Resource)

The property domainRes clarifies the resource in the subject position while the property rangeRes clarifies the resource in the object position. The following relation is taken as example:

dbr[4]:LeonardNimoy dbp:starredIn dbr:StarTrek

This relation becomes after separation as follows:

domainRes(dbp:starredIn, dbr:LeonardNimoy)

rangeRes(dbp:starredIn , dbr:StarTrek)

Although in this search we don't treat the literal data or the time periods as in algorithm of SDType, but this separation helps us to get benefit from the part related to the subject site in these expressions, as the definitions of the subject part are extracted

[4] The prefix **dbr** denotes for resources which start with http://dbpedia.org/resource

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

from the ontology file. The data of the subject position are extracted with disregarding the data of the object position which are literal from the data set of the properties. This information helps in enriching the inference of the resources types in the subject position. The following relation is taken as example:

dbr:Daunorubicin dbp:iupacName "acetyl"@en

After disregarding the part related to the object position, we get benefit from the part related to the subject position as follows:

domainRes(dbp:iupacName, dbr:Daunorubicin)

Where the following expression is extracted from the ontology file:

rdfs:domain(dbp[5]:iupacName, dbo:Drug)

Consequently we can infer that the resource dbr:Daunorubicin is of type dbo:Drug. The model and the prepared data are input for the programming tool ProbCog which executes Markov Logic Networks for discovering the types of the untyped sources, where from the input of the tool, types are extracted for the untyped resources according to the specified threshold. These types are written in a new n-triple file.

12. EXPERIMENTS

We have carried out an evaluation similar to the one of the study related to the statistical distributions [10], in the evaluation, the statement which have types have been used benefitting from the type information as standard. We randomly samples two datasets and then tried to reconstruct their types. We will discuss the execution of the two samples by using the reference algorithm SDType and the proposed algorithm MlnType.

Figure 3 shows pieces of background knowledge about the movie dbr:StarTrek in RDF triples, which can be obtained from Dbpedia. A RDF triple consists of a subject, a property and an object. As we can see from the figure, there can be incoming knowledge, e.g. dbr:LeonardNimoy dbp:starredIn dbr:StarTrek where dbr:StarTrek is used as an object, as well as outgoing knowledge such as dbr:StarTrek → dbp:genre → dbr:scienceFiction where dbr:StarTrek is a subject.

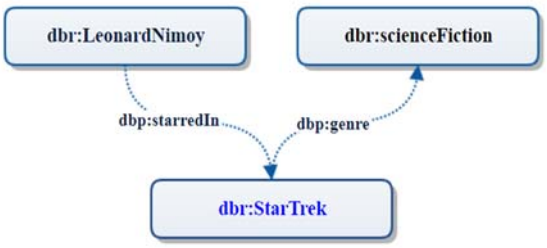

Figure 3: An Example About The Movie Dbr:StarTrek From Dbpedia.

In this evaluation we depended on the incoming properties because the inference through the outgoing properties is considered relatively easy.

In the first experiment, the proposed algorithm for type completion "MlnType" could discover 149 new type triples and achieve 82% by F-measure at executing 150 steps, While It could achieve 90% by F-measure at executing 1000 steps, at specifying the threshold 0.80. Figure 4 shows the results of executing the proposed algorithm MlnType on the first sample with 150 and 1000 steps.

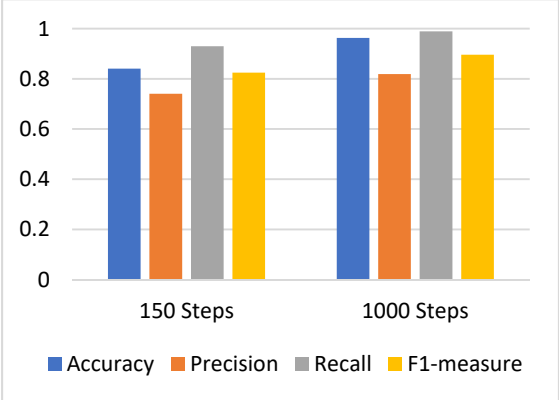

Figure 4: The Results of Executing the Proposed Algorithm MlnType on the First Sample.

While the reference algorithm SDType could discover 80 new type triples and achieve 89% by F-measure at the threshold 0.15 and it couldn't do the required task at specifying higher threshold. Noting that the proposed algorithm MlnType could work on 15 different classes while the work of the reference algorithm SDType was restricted only on 5 classes.

[5] The prefix **dbp** denotes for properties which start with http://dbpedia.org/ontology

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 5 shows the results of executing the reference algorithm SDType. Table 5 presents the compered results Between SDType and MlnType at executing 1000 Steps.

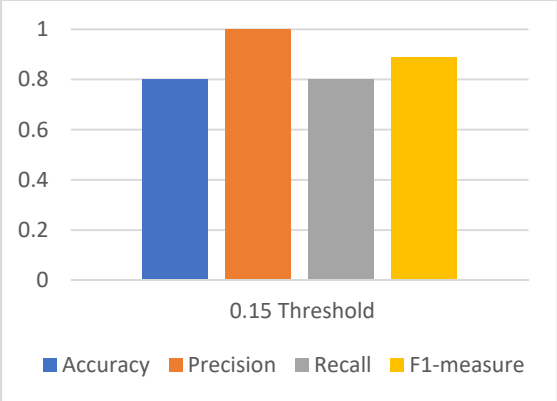

Figure 5: The Results of Executing the Algorithm SDType On the First Sample.

Table 5: Comparison Results of the First Sample.

Class	Algorithm	A	P	R	F
Owl:Thing	SDType	1	1	1	1
	MlnType	1	1	1	1
Schema_RiverBodyOfWater	SDType	1	1	1	1
	MlnType	1	1	1	1
Schema_BodyOfWater	SDType	1	1	1	1
	MlnType	0.94	0	1	0
dbo_BodyOfWater	SDType	1	1	1	1
	MlnType	0.94	0	1	0
Schema_Place	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Place	SDType	1	1	1	1
	MlnType	1	1	1	1
Schema_Country	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Country	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_PopulatedPlace	SDType	1	1	1	1
	MlnType	0.89	1	0.89	0.94
dbo_City	SDType	0	1	0	0
	MlnType	1	1	1	1
dbo_Settlement	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_NaturalPlace	SDType	0	1	0	0
	MlnType	0.94	1	0.94	0.97
dbo_River	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Mountain	SDType	0	1	0	0
	MlnType	0.72	0.28	1	0.44
dbo_Stream	SDType	1	1	1	1
	MlnType	1	1	1	1
Average	SDType	0.8	1	0.8	0.89
	MlnType	0.96	0.82	0.98	0.90

In the second experiments, the algorithm proposed for type completion MlnType could discover 141 new type triples and achieve 86% by F-measure at executing 150 steps, While the algorithm could discover 126 new type triples and achieve 94% by F-measure at executing 1000 steps, at specifying the threshold 0.80. Figure 6 clarifies the results of executing the proposed algorithm MlnType on the second sample with 150 and 1000 steps.

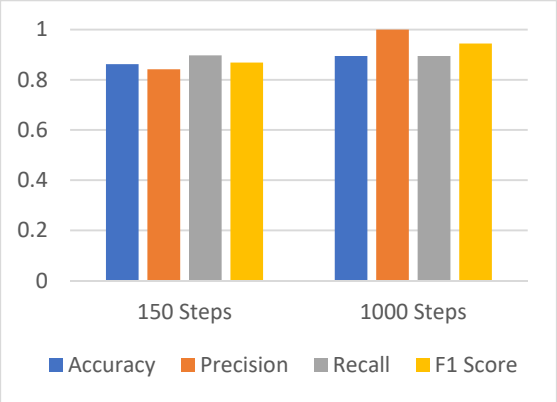

Figure 6: The Results of Executing the Proposed Algorithm MlnType On the Second Sample.

The algorithm SDType could discover 50 new type triples and achieve 83% by F-measure at the threshold 0.15 and it couldn't do the required task at specifying higher threshold. Noting that the proposed algorithm MlnType could work on 19 different classes while the work of the reference algorithm SDType was restricted only on 10 classes. Figure 7 clarifies the results of executing the reference algorithm SDType. Table 6 presents the compered results.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

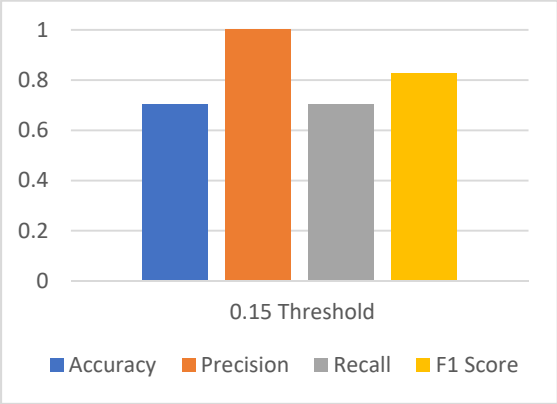

Figure 7: The Results of Executing the Algorithm SDType On the Second Sample.

Table 6: Comparison Results of the Second Sample.

Class	Algorithm	A	P	R	F
Owl:Thing	SDType	0.47	1	0.47	0.64
	MlnType	1	1	1	1
foaf_Person	SDType	0.47	1	0.47	0.64
	MlnType	1	1	1	1
dbo_Person	SDType	0.47	1	0.47	0.64
	MlnType	1	1	1	1
Schema_Person	SDType	0.47	1	0.47	0.64
	MlnType	1	1	1	1
dbo_Agent	SDType	0.47	1	0.47	0.64
	MlnType	1	1	1	1
dbo_SoccerPlayer	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_MusicalArtist	SDType	0	1	0	0
	MlnType	0	1	0	0
dbo_Artist	SDType	0	1	0	0
	MlnType	0	1	0	0
Schema_MusicGroup	SDType	0	1	0	0
	MlnType	1	1	1	1
dbo_Athlete	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Country	SDType	1	1	1	1
	MlnType	1	1	1	1
Schema_Country	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Organization	SDType	1	1	1	1
	MlnType	1	1	1	1
Schema_Organization	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Band	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Place	SDType	1	1	1	1
	MlnType	1	1	1	1
Schema_Place	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_PopulatedPlace	SDType	1	1	1	1
	MlnType	1	1	1	1
dbo_Settlement	SDType	1	1	1	1
	MlnType	1	1	1	1
Average	SDType	0.70	1	0.70	0.82
	MlnType	0.89	1	0.89	0.94

It can be observed that the proposed algorithm for type completion is working well on the datasets. As the results has been revealed that the inference by using the proposed algorithm MlnType is richer.

We notice from these two experiments that the reference algorithm SDType could achieve 100% by precision measure, which means the ability of the algorithm to not recall any wrong results, but the second experiment has reflected the decreasing of the sensitivity of the reference algorithm as it achieved 70% by Recall measure, i.e. that it couldn't recall all the correct results.

We have analyzed how well MlnType is suitable for adding type information to untyped resources. It can be observed that on the first sample, 7.27 types per instance can be generated at executing 1000 steps. While 5 types per instance can be generated by the reference algorithm SDType. Table 7 shows that MlnType is richer than SDType.

Table 7: Comparison results of MlnType and SDType for assigning types to untyped instances.

	First Sample		Second Sample	
	SDType	MlnType	SDType	MlnType
Newly Type instances	80	131	50	126
Avg. types per instance	5	7.27	5	6
Distinct Classes	15		19	

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SDType Algorithm surpasses the proposed algorithm MlnType in the runtime, where the runtime for each of the two samples hasn't exceeded five minutes by using SDType, while the runtime by using MlnType has lasted nearly 15 hours for 1000 steps, and nearly two hours and half for 150 steps for each of the two samples.

The reason of the small size of the samples and the few number of the executed steps is attributed to constraints related to the capabilities of the computer which the proposed algorithm is executed by. Execution of the algorithm is done by using a Laptop of 2.00GHZ processor and Random memory 4GB.

Noting that execution of a system called "mandolin" similar to the technique of the proposed algorithm MlnType and using Markov Logic Networks to discover new truths has lasted nearly 24 hours on the whole data of DBpedia version 2015 and on a high-tech server which has 64-core with 256GB RAM [26].

13. CONCLUSION

In this research, we proposed an approach to improve the quality of the unstructured Linked Data sets without using any external knowledge, depending on the means of Statistical relational learning through Markov Logic Networks which merges Markov network with First- order logic. Markov Logic is suitable for our approach as it supports weighted and hard (unweighted) constraints. Thus, it fulfills all requirements with respect to the essential types of statements and constraints. In order to infer new type statements in datasets, we rely on Markov Logics' MC-SAT inference. We integrated the Markov Logic solver ProbCog in our application but it is mentionable that our formalism does not rely on this specific system. Hence, it is possible to use our formalism with other Markov Logic solvers that support MC-SAT inference.

In this research, since we don't study the intervals, we suggest to work with a treatment of such type of data, and specifically those related to the wrong time periods such as date of death and date of birth, as an example identical to what we have done; taking into consideration adding new rules, discussing when the events happened such as an event had happened before another one, and ending an event within another one. Results showed that the proposed algorithm could infer on correct types, the algorithm SDType couldn't. this is because of the good coverage of the classes in the ontology. As well, the results were remarkably improved at increasing the number of the steps. In the future, we intend to expand the size of samples, and we plan to improve our approach by using fuzzy logic instead of first order logic.

REFERENCES

[1] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, "Quality Assessment for Linked Data: A Survey," *Semantic Web Journal,* vol. 7, no. 1, pp. 63-93, 2016.

[2] M. Morsey, J. Lehmann, S. Auer, C. Stadler and S. Hellmann, "DBpedia and the Live Extraction of Structured Data from Wikipedia," *Program electronic library and information systems ,* vol. 46, no. 2, pp. 157-181, 2012.

[3] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. v. Kleef, S. Auer and C. Bizer, "DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia," *Semantic Web Journal,* vol. 6, no. 2, pp. 167-195, 2015.

[4] C. Bizer, T. Heath and T. Berners-Lee, "Linked Data - The Story So Far," *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 5, no. 3, pp. 1-22, 2009.

[5] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellmann, "DBpedia - A crystallization point for the Web of Data," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 7, no. 3, pp. 154-165, 1 September 2009.

[6] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen and A. Zaveri, "Test-driven Evaluation of Linked Data Quality," in *Proceedings of the 23rd international conference on World Wide Web*, New York, NY, USA, 2014.

[7] A. Hogan, A. Harth, A. Passant, A. Polleres and S. Decker, "Weaving the Pedantic Web," in *(LDOW2010) 3rd International Workshop on Linked Data on the Web, in conjunction with 19th International World Wide Web Conference, CEUR*, Raleigh, USA, 2010.

[8] M. Nickel, K. Murphy, V. Tresp and E. Gabrilovich, "A Review of Relational Machine Learning for Knowledge Graphs," in *Proceedings of the IEEE*, 2016.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] H. Paulheim and C. Bizer, "Improving the Quality of Linked Data Using Statistical Distributions," *International Journal on Semantic Web & Information Systems,* vol. 10, no. 2, pp. 63-86, 1 April 2014.

[10] H. Paulheim and C. Bizer, "Type Inference on Noisy RDF Data," in *12th International Semantic Web Conference*, Sydney, NSW, Australia, 2013.

[11] H. Khosravi and B. Bina, "A Survey on Statistical Relational Learning," in *Proceeding AI'10 Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence*, Berlin, 2010.

[12] L. D. Raedt and K. Kersting, "Statistical Relational Learning," in *Encyclopedia of Machine Learning*, C. Sammut and G. Webb, Eds., Springer US, 2010, pp. 916-924.

[13] B. Taskar, M.-F. Wong, P. Abbeel and D. Koller, "Link Prediction in Relational Data," in *Advances in Neural Information Processing Systems (NIPS) 16*, Cambridge, 2004.

[14] Y. Wang and M. Kitsuregawa, "Link based clustering of web search results," in *International Conference on Web-Age Information Management (WAIM)*, Berlin, Heidelberg, 2001.

[15] R. Davis, H. Shrobe and P. Szolovits, "What is a Knowledge Representation?," *AI Magazine,* vol. 14, no. 1, pp. 17-33, 1993.

[16] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Scientific American,* vol. 284, no. 5, pp. 34-43, May 2001.

[17] D. Koller and N. Friedman, "Probabilistic Graphical Models Principles and Techniques - Adaptive Computation and Machine Learning," 2009.

[18] M. R. Genesereth and N. Nilsson , "Logical foundations of artificial intelligence," vol. 9, 1987.

[19] P. Domingos and D. Lowd, "Markov Logic: An Interface Layer for Artificial Intelligence," *Synthesis Lectures on Artificial Intelligence and Machine Learning,* pp. 1-155, 2009.

[20] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning,* vol. 62, no. 1-2, pp. 107-136, 2006.

[21] J. Neville and D. Jensen, "Iterative Classification in Relational Data," in *Proceedings of the Workshop on Learning Statistical Models from Relational Data, Seventeenth National Conference on Artificial Intelligence*, Austin, TX, 2000.

[22] A. Giovanni, A. Gangemi, V. Presutti and P. Ciancarini, "Type inference through the analysis of wikipedia links," *Linked Data on the Web (LDOW),* 2012.

[23] J. Sleeman and T. Finin, "Type Prediction for Efficient Coreference Resolution in Heterogeneous Semantic Graphs," in *IEEE Seventh International Conference on Semantic Computing (ICSC)*, Irvine, CA, 2013.

[24] A. Pohl, "Classifying the Wikipedia Articles in the OpenCyc Taxonomy," in *In Web of Linked Entities Workshop (WoLE 2012)*, 2012.

[25] A. P. Aprosio, C. Giuliano and A. Lavelli, "Automatic Expansion of DBpedia Exploiting Wikipedia Cross-Language Information," in *10th Extended Semantic Web Conference (ESWC 2013)*, Montpellier, France, 2013.

[26] T. Soru, D. Esteves, E. Marx and A.-C. Ngonga Ngomo, "Mandolin: A Knowledge Discovery Framework for the Web of Data," *CoRR,* vol. abs/1711.01283, 2017.