Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

Integration of Ontology Transformation into Hidden Markov Model

Lazarre Warda*, Guidedi Kaladzavi, Amaria Samdalle, Kolyang

Laboratoire de Recherche en Informatique (LaRI), The University of Maroua, P.O. Box: 46 Maroua, Cameroon

* Correspondence: Lazarre Warda (lzwarda2015@gmail.com)

Received: 09-08-2022 **Revised:** 10-05-2022 **Accepted:** 10-22-2022

Citation: L. Warda, G. Kaladzavi, A. Samdalle, and Kolyang, "Integration of ontology transformation into hidden Markov model," *Inf. Dyn. Appl.*, vol. 1, no. 1, pp. 2-13, 2022. https://doi.org/10.56578/ida010102.

© 2022 by the authors. Published by Acadlore Publishing Services Limited, Hong Kong. This article is available for free download and can be reused and cited, provided that the original published version is credited, under the CC BY 4.0 license.

Abstract: The goal of this study is to suggest a method for turning an ontology into a hidden Markov model (HMM). Ontology properties (relationships between classes) and ontology classes are taken as HMM symbols and states, respectively. Knowledge is represented in many different fields using the central element of the Semantic Web dubbed ontology. The authors employed machine learning technologies like HMM to add knowledge to these ontologies or to extract knowledge from within them. The meaning obtained from ontologies is not described during this task. The ontology triples that were extracted using SPARQL queries are used in this paper to transform the ontology into an HMM in order to handle this semantic. The Pizza ontology has been used to implement this method, which is based on lightweight ontologies.

Keywords: Ontology; Triple; Learning; Sequence; HMM

1. Introduction

Ontology is a term that comes from philosophy, more specifically from the branch of metaphysics. The science of artificial intelligence then adopted ontology as a knowledge representation model for assisting reasoning in knowledge-based systems relatively early on. Ontologies are now at the heart of the architecture of the semantic web as a support for resource annotations (documents, images, videos, etc.) and facilitating communication between users and applications as well as between the apps themselves. "An ontology is an explicit specification of a conceptualization," claims Gruber [1].

The term "conceptualization" refers to an abstract representation of a certain reality phenomenon that enables the identification of the pertinent concepts for this occurrence. When a concept is described as "explicit," it means that it is clear and precise. Ontology use in applications can sometimes inspire authors to develop methods for discovering ontology properties. These methods were created using machine learning. According to Arthur Samuel, the scientific subject of machine learning enables computers to learn without explicit programming. It gives machines the ability to manage data using various statistical models and techniques [2].

Some of these algorithms are for supervised learning (decision tree, naive bayes, support vector machine, and neural networks), unsupervised learning (k-means, and hidden Markov model), semi-supervised learning (generative models, and self-training). Some authors linked these algorithms to ontologies in order to accomplish a variety of tasks. Authors suggested methods like Bayesian networks and ontologies for mapping, translation, and classification to support these tasks [3-7]. For prediction, several authors [8, 9] coupled decision trees and ontologies. For prediction, reasoning, mapping, and classification, several authors employed ontologies and neural networks [10-13] or ontologies with support vector machines [14]. Hidden Markov model (HMM) was created by Rabiner and Juang after being introduced by Rabiner and Juang [15]. It is employed for classification, prediction, comparison, and speech and pattern recognition [16-20].

Since HMM conserved semantics between elements, it is the technique that is most frequently employed for events when phenomena are sequentially and semantically linked. For ontologies represented knowledge, the main goal is to maintain semantics when managing their concepts. The HMM is therefore designed to learn ontology concepts and all of their attributes. This concept was backed by two sets of tasks: (1) using HMMs to populate ontologies [21-25]; and (2) combining HMMs and ontologies to develop systems [26-32].

The drawback of these approaches is that, although some authors used ontology ideas as symbols for HMMs, no clear relationship between ontology and HMM was established. The use of HMM to create modular ontologies is a suggestion made by Warda et al. [33]. Learning ontology properties will therefore be useful for some applications.

As a result, the goal of this study is to suggest a method for turning an ontology into an HMM. Ontology properties (relationships between classes) and ontology classes are taken as HMM symbols and states, respectively. These classes and attributes come from the triples collection that was produced by SPARQL queries on the target ontology. Ontology axioms are not handled by these queries. Only light ontologies are therefore altered. Axioms are managed in heavy ontology by removing the triples that contained them. To implement this concept, certain equations are provided. One unique feature is the ability to convert many ontologies into a single HMM. For instance, a single HMM can represent a number of ontologies depending on the domain. The Pizza ontology is used to illustrate this method.

The rest of the paper is organized as follow: Ontologies and HMMs are briefly defined in Section 2 along with the current state of the art in this area. The suggested approach is discussed in Section 3, experimental findings and discussion are covered in Section 4, and the conclusion and potential future directions are covered in Section 4.

2. State of the Art

2.1 Hidden Markov Model (HMM)

The content of this section derives from the study of Iloga et al. [16]. Formally, a HMM $\lambda = \{N, M, A, B, \pi\}$ is a given of:

(1) N: its number of states. The set of states is noted $S = \{S_1, S_2 \dots S_N\}$. Generally, at time t, the state is noted $q_t \epsilon S$.

(2) M: its number of observation symbols. The set of observation symbols is noted $V = \{v_1, v_2 \dots v_M\}$. Generally, at time t, the symbol observed by the model is noted $O_t \epsilon V$.

(3) $A = [a_{ij}]$: its state transition probabilities distributions where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i),\ 1 \leq i, j \leq N$.

(4) $B = [b_j(k)]$: its observation symbols probabilities distributions where $b_j(k) = P(v_k\ at\ time\ t \mid q_t = S_j)$ in each state $S_j,\ 1 \leq j \leq N$ and $1 \leq k \leq M$.

(5) $\pi = [\pi_i]$: its initial state probabilities distributions where $\pi_i = P(q_1 = S_i),\ 1 \leq i \leq N$.

Given T observations symbols $O = (O_1, O_2, \dots, O_T)$. This sequence can be generated by a HMM $\lambda = \{N, M, A, B, \pi\}$ as shown in Figure 1. This representation is called generated Markov chain.

$$\begin{tikzcd}
\text{(Symbols)} & O_1 & O_2 & \dots & O_T \\
\text{(States)} & q_1 \arrow[u] \arrow[r] & q_2 \arrow[u] \arrow[r] & \dots \arrow[r] & q_T \arrow[u]
\end{tikzcd}$$

Figure 1. Markov chain representation

2.2 Ontology

Ontology is a set of concepts joined by relationships and based on some functions more especially some axioms. It is used to refer to a body of knowledge describing some domains, typically a common-sense knowledge domain, using a representation vocabulary. Given a target domain, its ontology forms the heart of any system of knowledge representation for that domain [34]. The components of ontology are: Concepts, Relations, Instances and Axioms. Concepts represent a set of entities within the domain. Relations specify the interaction among concepts. Instances indicate the concrete examples of concepts within the domain and axioms denote a statement that is always true [35]. Following the components of ontology, we have heavy ontologies and lightweight ontologies. Heavy ontologies are those which handle axioms and lightweight ontologies do not handle axioms. Lightweight ontologies give hierarchical order of classes and can precise domain and range of some properties (principally *owl: Object Property*). If axioms are avoided in heavy ontology, it becomes lightweight. For a given ontology, each class or relation is identified by its IRI (International Resource Identifier). Many tools help to build ontologies but the commonly used tool is *Protégé 2000*. Ontologies can be stored into many formalisms: RDF (Resource Description Framework), RDFS (RDF - Schema), XML (Extended Markup Language), OWL (Web Ontology language). To query ontology in *Protégé 2000*, we can use SPARQL. SPARQL is a recursive acronym that stands for SPARQL Protocol and RDF Query Language.

2.3 Related Works

Approaches proposed hitherto are divided into two categories. For the one hand, HMMs are used to build or populate ontologies and for the other hand, ontologies are mixed to HMM to build systems. Several works maintained these ideas. Valarakos et al. [21] proposed a methodology for enriching multi-lingual domain ontology using machine learning (based on HMM), principally CROSSMARC ontologies. Their approach consists to add instances of ontology concepts using machine learning techniques. HMM is trained on the ontology instances before apply it to web pages using Viterbi algorithm to recognize matches, hence locate new ontology instances and Packer and Embley followed this approach in the study [23] to propose ListReader, an approach based on HMM to populate ontology. HMM derived from OCRed, a collection of page images. Thus, the train HMM generated labeled text, which is transformed into predicates for ontology using Viterbi algorithm. For Monika and Raju [25] ontology can be obtained with another manner. They proposed an effective model integration algorithm based on HMM to build ontology. HMMs are used to capture knowledge from datasets before initialize process and each ontology concepts derived from this approach respect initial prediction. Bratus et al. [26] proposed an approach which combined HMM and CRF models to extract data using ontology-guided search. They first identified and extracted part names from unstructured data and second, they developed TCBR (Textual Case-Based Reasoning) systems for service technicians and engineers. According to this goal, Azanzi and Camara [24] proposed an approach for knowledge extraction from source code based on HMM. It was applied to EPICAM, a tuberculosis surveillance system. Ontology is code in Java language and HMM is trained to identify Java code concepts to be extracted. To classify genes, Mi et al. [27], proposed PANTHER (protein annotation through evolutionary relationship) by integrating statistical tools (HMMs). They used HMMs to capture evolutionary relationships of genes families and subfamilies and they used ontology (GO) to annotate them. The idea of classification guided Prestat et al. [22], to propose FOAM (Functional Ontology Assignments for Metagenomes). This ontology is a database of HMMs used for classification. HMMs are obtained by fetching profiles of KEGG orthologs (KOs). Pipitone and Pirrone [28] proposed an approach to automatically generate ERD (Entity Relationship Diagrams) from OWL ontology based on HMM. To construct HMM using OWL/ERD, they took ERD as hidden states. They defined grammar to determine the transitions probabilities and observation probabilities derived from OWL/ERD mapping rules. Rani et al. [29] proposed OPAESFH, an approach based on ontology for personalizing system of E-learning using Fuzzy Petri Networks (FPN) and HMM. This system used metadata of SwetoDblp, an ontology of Computer Science bibliography data. The courses and exercises of E-learning are modeled with FPN. HMM are used to updated FPN parameters and to recommend level of learner while Karmegam [30] proposed an HMM and ontology based cross-lingual question answering system for the agricultural domain. The ontology is used to map knowledge components and HMM are used to identify the most suitable resource queried by user based on semantic relations among resources. Recently, to recognize group activities based on imageries data, Elangovan [31] proposed an approach where he considered the groups of human activities as ontologies. Then, these ontologies are used as sequences. These sequences (considered as symbols) are used to train HMM and to determine the probability of sequence evaluation. In the IoT (Internet of things) domain, Muthukumar et al. [32] build a semantic-based security platform to detect the malicious attack data. Ontologies (Semantic Sensor Networks Ontology and Temporal Ontology) are used to represent sensor data. HMM is used to identify anomalies derived from clustered data using observations of HMM. These works are summarized in Table 1.

Table 1. Some approaches used ontologies and HMM

Authors	Ref.	Year	Main idea
Valarakos et al.	[21]	2003	Populate ontology using HMM
Bratus et al.	[26]	2011	Use HMM and CRF models for data extraction guided by ontology search
Mi et al.	[27]	2013	Use ontology to annotate concepts and HMM to capture their relationships for classification of genes
Prestat et al.	[22]	2014	Collect HMMs to build ontology such that it can be possible for classification
Pipitone and Pirrone	[28]	2014	Generate ERD from OWL ontology using HMM
Packer and Embley	[23]	2015	Populate ontology using HMM
Azanzi and Camara	[24]	2017	Extract knowledge from ontology (Java source) using HMM
Rani et al.	[29]	2017	Use ontology to personalize system of E-learning based on FPN and HMM
Karmegam	[30]	2019	Combine HMM and ontology to build answering system for agricultural domain
Monika and Raju	[25]	2019	Build ontology using algorithms based on HMMs
Elangovan	[31]	2021	Train HMM based on ontologies to compute sequence evaluation
Muthukumar et al.	[32]	2021	Use HMM to identify anomalies from data (observations) derives from ontology

Approaches hereinbefore mentioned focused on populating ontologies using HMMs, training HMMs based on ontology concepts and combining ontologies and HMMs to build a target system. No-one typically found correspondence between ontology and HMM. This limit guided us to propose this approach.

3. Proposed Approach

3.1 Main Idea

This paper aims to propose an approach to transform an OWL ontology into a HMM, in other terms learn knowledge from ontology using HMM. To achieve this goal, ontology classes are considered as hidden states of HMM and ontology properties are considered as symbols of model. Therefore, the number of ontology classes is equal to the number of HMM hidden states and the number of ontology properties is equal to the number of HMM symbols. For different concepts of ontology (classes and properties), their associated IRI can be used or only their short name depending to the user. The methodology of this approach is detailed in Section 3.2.

3.2 Methodology

Figure 2 describes steps to transform ontology into HMM. Its input is an ontology and its output is a HMM.

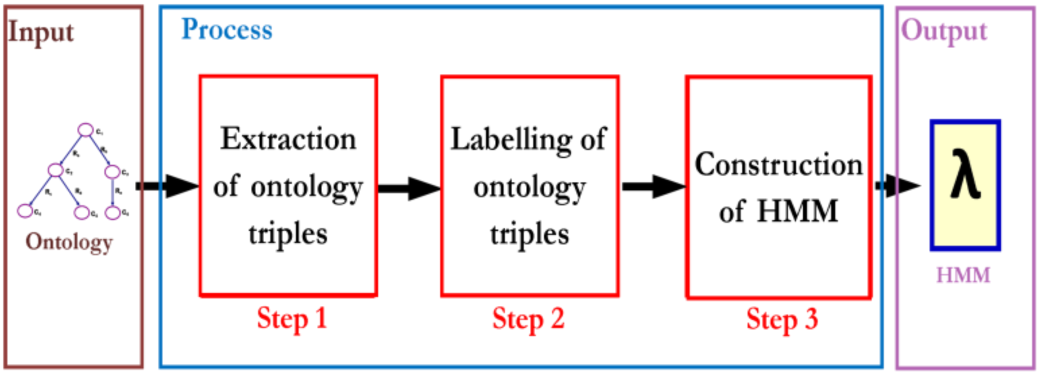


Figure 2. Process to transform ontology into HMM

3.2.1 Step 1: Extraction of ontology triples

Ontology triple is a *(subject*, *predicate*, *object)* set represented with Figure 3 In this configuration, *subject* and *object* are classes; *object* can also be an axiom according to the type of ontology. In this work, axioms are not considered since light ontologies are handled. And *predicate* denotes the relationship between *subject* and *object*.

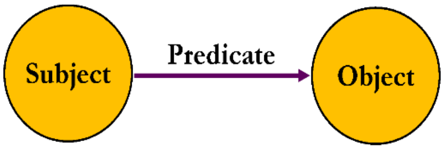


Figure 3. Representation of ontology triple

The step 1 consists to use ontology triples to replace whole ontology. The set of ontology triples can be obtained by querying ontology using SPARQL language. Here, user can define the condition of extraction. For instance, to get all ontology triples, we can use following query:

```
SELECT ?subject ?predicate ?object
WHERE { ?subject ?predicate ?object . }
```

This query means select all triples where *?predicate* is the relationship between *?subject* and *?object*. To be more precisely, some set of values can be defined on *?predicate*. It can be *owl:ObjectProperty*, *rdf:type*, *rdf:subClassOf* ... Hence, previous query can be modified like following queries:

```
SELECT ?subject ?predicate ?object
WHERE { ?subject ?predicate ?object .
        ?predicate a owl:ObjectProperty . }
```

And

```
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object . }
```

All these triples can be stored in *.txt* file for simple exploitation.

3.2.2 Step 2: Labeling of ontology triples

In this step, triples are labeled with integers. Since all ontology resources and relations are identified by IRI (which are string type), it is hard to manage them. Therefore, all concepts are replaced by integers. So, ontology triples are transformed into set of integers where each number refers to corresponding class or property. For example, if we consider this set of triples:

T = {(mango, subClassOf, fruit), (banana, subClassOf, fruit), (human, eat, fruit), (human, plant, vegetable)}

In this example, we assume that each expression is the short name of class or property (not its IRI), then each class and property are transformed as described in Table 2 and Table 3.

Table 2. Examples of labeled classes

Classes	Labels
mango	1
fruit	2
banana	3
human	4
vegetable	5

Table 3. Examples of labeled properties

Properties	Labels
subClassOf	1
plant	2
eat	3

Using Table 2 and Table 3, the labeled set of triples is:

$$T_{Labelled} = \{(1,1,2),(3,1,2),(4,2,2),(4,3,5)\}$$

3.2.3 Step 3: Construction of HMM

Let $\lambda = (N, M, A, B, \pi)$ a HMM. Construct λ refers to compute the components of matrix A and B and vector π.

For matrix A, the numbers of lines and columns corresponds to the number of hidden states of the model. Therefore, each a_{ij}, component of A, is the probability to move from the state i to the state j wherefore it means that there exists a relation moved from class labelled with i to one labelled with j. Eq. (1) is used to compute a_{ij}.

$$\boldsymbol{a_{ij}} = \frac{number\ of\ triples\ where\ \boldsymbol{i}\ is\ the\ subject\ and\ \boldsymbol{j}\ is\ the\ object}{number\ of\ triples\ where\ \boldsymbol{i}\ is\ the\ subject + \boldsymbol{\varepsilon}} \tag{1}$$

For matrix B, the number of lines corresponds to the number of model hidden states and the number of columns corresponds to the number of model symbols. Therefore, each $b_j(k)$, components of B, is the probability to observe symbol k at the state j wherefore it means that there exists a property labelled with k moved from the class labelled with j. Hence, each $b_j(k)$ is computed using Eq. (2):

$$\boldsymbol{b_j(k)} = \frac{number\ of\ triples\ where\ \boldsymbol{j}\ is\ the\ subject\ and\ \boldsymbol{k}\ is\ the\ predicate}{number\ of\ triples\ where\ \boldsymbol{k}\ is\ the\ predicate + \boldsymbol{\varepsilon}} \tag{2}$$

For vector π, the number of elements corresponds to the number of model hidden states. Therefore, each π_i, element of π, is the probability such that state i be the initial state of model wherefore it means that ontology can be browsed launching with the class labelled with i. Hence, Eq. (3) is used to compute each π_i:

$$\pi_i = \frac{number\ of\ triples\ where\ \boldsymbol{i}\ is\ the\ subject}{total\ number\ of\ triples + \boldsymbol{\varepsilon}} \tag{3}$$

In Eqns. (1)-(3), $\boldsymbol{\varepsilon}$ is a real positive number. It has been added to avoid division by zero and to ensure probabilities distributions of model components. To have the model, it is needed to readjust the values of *A*, *B* and π. The difference between 1 and the sum of the elements for each line of *A* and *B* is redistributed equitably to all elements of this line and the same technique is applied to the components of π. These readjustments follow the Eqns. (4)-(6) below:

For Matrix A:

$$a_{ij} = a_{ij} + \frac{\Delta_i^a}{N} \ where \ \Delta_i^a = 1 - \sum_{p=1}^{N} a_{ip} \tag{4}$$

In Eq. (4), the term Δ_i^a is the difference between 1 and the sum of elements of line number i in matrix A.

For Matrix B:

$$b_j(k) = b_j(k) + \frac{\Delta_j^b}{M} \ where \ \Delta_j^b = 1 - \sum_{o=1}^{M} b_j(o) \tag{5}$$

In Eq. (5), the term Δ_j^b is the difference between 1 and the sum of elements of line number j in matrix B.

For Vector π:

$$\pi_i = \pi_i + \frac{\Delta^\pi}{N} \ where \ \Delta^\pi = 1 - \sum_{p=1}^{N} \pi_p \tag{6}$$

In Eq. (6), the term Δ^π represents the difference between 1 and the sum of elements vector π.

For instance, according to the example of triples *T* in Section 3.2.2 transformed into labelled triples $T_{Labelled}$, the number of classes is 5 and the number of relations (properties) is 3. $a_{1,1}$ corresponds to the probability to move from state 1 to sate 1. It refers to the number of triples which have 1 as subject and 1 as object. There is one triple which has 1 as subject (1,1,2) and no one has 1 as object i.e. no move from state 1 to state 1, hence $a_{1,1} = 0$. For $a_{1,2}$ there is one move from state 1 to state 2(1,1,2) and only one move from 1 to another state outside state 2. Hence, $a_{1,2} = \frac{1}{1+\varepsilon}$. Similarly, others components of *A* are computed. $b_1(1)$ corresponds to the probability to observe symbol 1 in the state 1. It refers to the number of triples which have 1 as subject and 1 as relation. There is one triple which has 1 as state and 1 as symbol and there is two triples which has 1 as symbol. Hence $b_1(1) = \frac{1}{2+\varepsilon}$. Similarly, others components of *B* are computed. π_1 corresponds to the probability such that 1 be the initial state. It refers to the number of triples which the subject is 1 or the number of triples moved from 1. There is one triple which has 1 as subject. Hence $\pi_1 = \frac{1}{5+\varepsilon}$. Similarly, others components of π are computed. Thus, we have the following values:

$$A = \begin{bmatrix} 0 & \frac{1}{1+\varepsilon} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{1+\varepsilon} & 0 & 0 & 0 \\ 0 & \frac{1}{2+\varepsilon} & 0 & 0 & \frac{1}{2+\varepsilon} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{2+\varepsilon} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{2+\varepsilon} & 0 & 0 \\ 0 & \frac{1}{1+\varepsilon} & \frac{1}{1+\varepsilon} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\pi = \begin{bmatrix} \frac{1}{5+\varepsilon} & 0 & \frac{1}{5+\varepsilon} & \frac{2}{5+\varepsilon} & 0 \end{bmatrix}$$

Applying the Eqns. (4)-(6), we have:

For matrix A

$$\Delta_1^a = 1 - \frac{1}{1+\varepsilon} = \frac{\varepsilon}{1+\varepsilon} \quad \Delta_2^a = 1 \quad \Delta_3^a = 1 - \frac{1}{1+\varepsilon} = \frac{\varepsilon}{1+\varepsilon} \quad \Delta_4^a = 1 - \frac{2}{2+\varepsilon} = \frac{\varepsilon}{2+\varepsilon}$$

For matrix *B*:

$$\Delta_1^b = 1 - \frac{1}{2+\varepsilon} = \frac{1+\varepsilon}{2+\varepsilon} \quad \Delta_2^b = 1 \;\; \Delta_3^b = 1 - \frac{1}{2+\varepsilon} = \frac{1+\varepsilon}{2+\varepsilon} \quad \Delta_4^b = 1 - \frac{2}{1+\varepsilon} = \frac{\varepsilon-1}{1+\varepsilon}$$

For vector π:

$$\Delta^\pi = 1 - \frac{4}{5+\varepsilon} = \frac{1+\varepsilon}{5+\varepsilon}$$

Hence, the components of *A*, *B* and π after readjustment are:

$$A = \begin{bmatrix} \frac{\varepsilon}{5+5\varepsilon} & \frac{5+\varepsilon}{5+5\varepsilon} & \frac{\varepsilon}{5+5\varepsilon} & \frac{\varepsilon}{5+5\varepsilon} & \frac{\varepsilon}{5+5\varepsilon} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{\varepsilon}{5+5\varepsilon} & \frac{5+\varepsilon}{5+5\varepsilon} & \frac{\varepsilon}{5+5\varepsilon} & \frac{\varepsilon}{5+5\varepsilon} & \frac{\varepsilon}{5+5\varepsilon} \\ \frac{\varepsilon}{10+5\varepsilon} & \frac{5+\varepsilon}{10+5\varepsilon} & \frac{\varepsilon}{10+5\varepsilon} & \frac{\varepsilon}{10+5\varepsilon} & \frac{5+\varepsilon}{10+5\varepsilon} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix} \quad B = \begin{bmatrix} \frac{4+\varepsilon}{6+3\varepsilon} & \frac{1+\varepsilon}{6+3\varepsilon} & \frac{1+\varepsilon}{6+3\varepsilon} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{4+\varepsilon}{6+3\varepsilon} & \frac{4+\varepsilon}{6+3\varepsilon} & \frac{4+\varepsilon}{6+3\varepsilon} \\ \frac{\varepsilon-1}{3+3\varepsilon} & \frac{2+\varepsilon}{3+3\varepsilon} & \frac{2+\varepsilon}{3+3\varepsilon} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

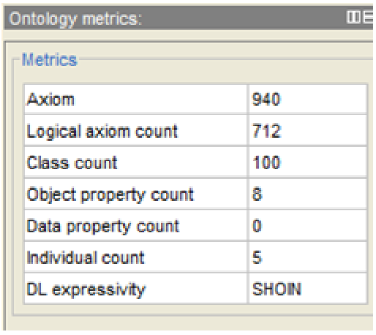
$$\pi = \begin{bmatrix} \frac{6+\varepsilon}{25+5\varepsilon} & \frac{1+\varepsilon}{25+5\varepsilon} & \frac{6+\varepsilon}{25+5\varepsilon} & \frac{11+\varepsilon}{25+5\varepsilon} & \frac{1+\varepsilon}{25+5\varepsilon} \end{bmatrix}$$

With this approach, multiple ontologies can be handled with single HMM. In this case, each ontology is transformed into set of triples. All these sets are considered to compute model parameters (A, B and π) using Eqns. (1)-(3). Nevertheless, in this case, the set of classes which should be considered is the union of the sets of classes derived from each ontology, similarly for the set of relations (properties).

4. Experiment of the Approach

4.1 Experimental Results on Pizza Ontology

This approach was experiment on Pizza ontology (https://protege.stanford.edu/ontologies/pizza/pizza.owl) (pizza.owl) developed for educational purposes by the University of Manchester, United Kingdom. It describes all concepts concerning pizza. Opened with *Protégé 2000*, it has one hundred classes and eight object properties. The partial view of pizza ontology metrics is given in Figure 4.



Ontology metrics:

Metrics

Axiom	940
Logical axiom count	712
Class count	100
Object property count	8
Data property count	0
Individual count	5
DL expressivity	SHOIN

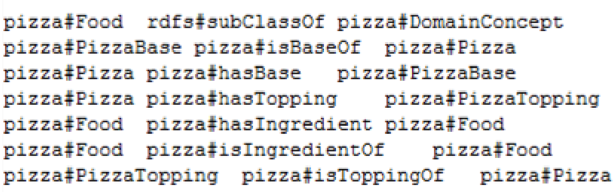
Figure 4. Partial view of pizza metrics

For this experiment, triples extracted with Python language, derived from the following SPARQL request:

SELECT ?s ?o
WHERE {?s rdfs:subClassOf ?o .

?s a owl:Class .
?o a owl:Class . }
And
SELECT ?s ?p ?o
WHERE { ?p a owl:ObjectProperty .
?p a rdfs:domain ?s .
?p a rdfs:range ?o . }

The results of these queries gave 90 triples and a part of triples is given in Figure 5. To have the view of triples, IRI were replaced with namespace (the term before symbol #). For each line in this figure, the first term corresponds to the subject of triple, the second corresponds to the predicate and the other one to the object.



```
pizza#Food  rdfs#subClassOf pizza#DomainConcept
pizza#PizzaBase pizza#isBaseOf  pizza#Pizza
pizza#Pizza pizza#hasBase   pizza#PizzaBase
pizza#Pizza pizza#hasTopping    pizza#PizzaTopping
pizza#Food  pizza#hasIngredient pizza#Food
pizza#Food  pizza#isIngredientOf    pizza#Food
pizza#PizzaTopping  pizza#isToppingOf   pizza#Pizza
```

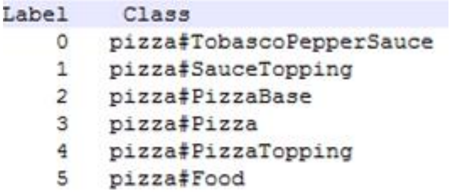
Figure 5. Partial view of triples

With this set of triples, we had metrics of triples summarized in Table 4:

Table 4. Metrics of pizza triples

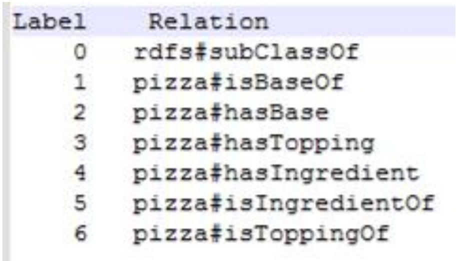
Elements	Quantity
Number of triples	90
Number of classes	85
Number of relations	07

The rest of the work is done with Java language. Classes and relations are then labeled and Figure 6 (resp. Figure 7) shows partial view of labeled classes (resp. labeled relations).



Label	Class
0	pizza#TobascoPepperSauce
1	pizza#SauceTopping
2	pizza#PizzaBase
3	pizza#Pizza
4	pizza#PizzaTopping
5	pizza#Food

Figure 6. Partial view of labelled classes



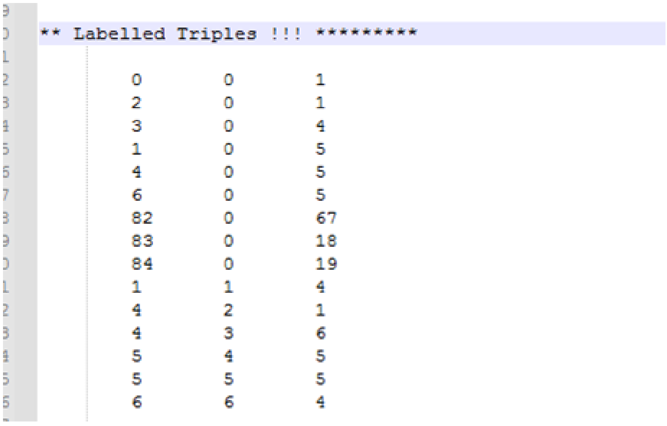
Label	Relation
0	rdfs#subClassOf
1	pizza#isBaseOf
2	pizza#hasBase
3	pizza#hasTopping
4	pizza#hasIngredient
5	pizza#isIngredientOf
6	pizza#isToppingOf

Figure 7. Partial view of labelled relations

The set of triples in Figure 5 is transformed into labeled triples according to labelled classes and labelled relations. The Figure 8 shows partial view of the corresponding labelled triples.

Full results described in Figures 5 – 8 are available in appendix A.

The set of labelled triples in Figure 8 are considered as the input for HMM construction. Since the number of classes and relations are obtained using the set of triples, the number of HMM states is 85, the number of HMM symbols is 7 and the sequence of observations to be used contained 90 observations.



```
** Labelled Triples !!! *********

        0       0       1
        2       0       1
        3       0       4
        1       0       5
        4       0       5
        6       0       5
        82      0       67
        83      0       18
        84      0       19
        1       1       4
        4       2       1
        4       3       6
        5       4       5
        5       5       5
        6       6       4
```

Figure 8. Partial view of labelled triples

After the application of proposed approach, the characteristics of HMM components are:

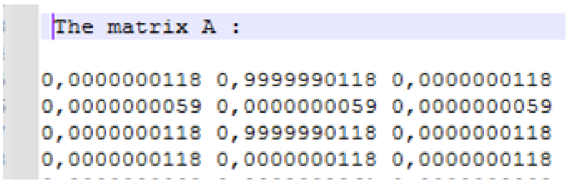
(1) Matrix A: the number of lines and the number of columns, denoted N, are equal to 85;

(2) Matrix B: the number of lines, denoted N, is equal to 85 and the number of columns, denoted M, is equal to 7;

(3) Vector π: the number of components, denoted N, is equal to 85.

The value of ε was fixed to 10^{-6}. This value can be modified by user and depends to the desired precision. To show the impact of ε, values are printed with 10 digits after coma.

For the high number of lines and columns of A, B and π, just some partial views are respectively presented in Figures 9, 10 and 11. In matrix A given by Figure 9, in the first line and first column, $a_{1,1} = 0.0000000118$, it is the probability such that there is a relation moved from the class labeled with 0 to the class labeled with 0. Since this probability reaches 0, it means that no relation exists between these two classes wherefore there is a relation between the class labelled with 0 and the one labelled with 1 because $a_{1,2} = 0.9999990118$.

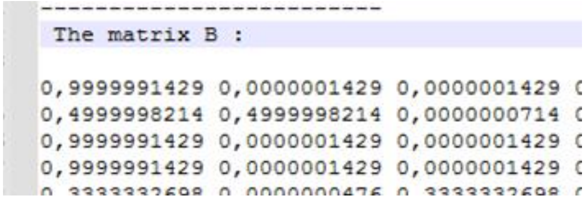


```
The matrix A :

0,0000000118 0,9999990118 0,0000000118
0,0000000059 0,0000000059 0,0000000059
0,0000000118 0,9999990118 0,0000000118
0,0000000118 0,0000000118 0,0000000118
```

Figure 9. Partial view of matrix A

In matrix B in Figure 10, $b_0(0) = 0.9999991429$. This value corresponds to the probability that the relation labelled with 0 is moved from the class labelled with 0. $b_0(1) = 0.0000001429$ means that the relation labelled with 1 is not moved from the class labelled with 0.

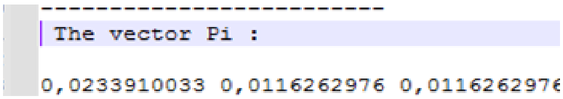


```
-------------------------
 The matrix B :

0,9999991429 0,0000001429 0,0000001429
0,4999998214 0,4999998214 0,0000000714
0,9999991429 0,0000001429 0,0000001429
0,9999991429 0,0000001429 0,0000001429
0,3333333698 0,0000000476 0,3333333698
```

Figure 10. Partial view of matrix B

In vector π given by Figure 11, we have $\pi = 0.0233910033$. This value corresponds to the probability such that pizza ontology can be browsed starting from the state labelled with 0.



```
-------------------------
 The vector Pi :

0,0233910033 0,0116262976 0,011626297
```

Figure 11. Partial view of vector π

The full parameters (Figures 9 – 11) of the obtained HMM are available in appendix A.

4.2 Discussion

The proposed approach focuses on learning ontology concepts through HMM, in other words the transformation of ontology into HMM. This technique handled only lightweight ontology and it is the limit of this approach. Nevertheless, a heavy ontology can be used however triples which contain axioms will be avoided. Hence, sometimes, the set of relations can be reduced to *rdfs:subClassOf* and then the HMM will be initialize using only one symbol. The sensibility of this approach is based on the value of ε. Since authors [21-32] based on populating ontologies through HMM and mixed HMM and ontologies, in this work, a strictly relationship is outlined between ontology and HMM. As precise in Section 3.2, in the case of multiple ontologies, a single HMM can represent them using this approach. For the main goal of ontology is to represent knowledge – and knowledge is based on semantic –, HMM consolidated this semantic because Eqns. (1)-(3) are based on the triples obtained via SPARQL querying on target ontology. With this approach, comparing two ontologies can be referred to comparing the two corresponding HMMs since they captured ontologies properties. Some challenges are outlined concerning the modularization of ontology [33]. This technique can be a possible issue to extract ontology modules. Another challenge is to transform a HMM into ontology.

5. Conclusions

This paper proposed a method to transform ontology into HMM. For this, ontology is transformed into triples using SPARQL querying, then triples are transformed into labelled triples using labelled classes and labelled relations and HMM parameters are initialized with these labelled triples. This approach was experimented on Pizza ontology and results are presented and discussed. This approach does not handle heavy ontologies and this constitutes its drawback. Since the main purpose of machine learning is to help computers to learn from several data sources, this approach can contribute to ensure this goal concerning data represented by ontologies. The future trends are to apply HMM algorithms and tasks (prediction, classification, clustering ...) and to perform the results in ontology engineering.

Data Availability

The data (Pizza ontology, an owl file) supporting our research results is deposited in https://protege.stanford.edu/ontologies/pizza/pizza.owl.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] R. T. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Syst. Lab.*, vol. 5, no. 2, pp. 199-220, 1993. http://dx.doi.org/10.1006/knac.1993.1008.

[2] B. Mahesh, “Machine learning algorithms-a review,” *Int J. Sci. Res.*, vol. 9, no. 1, pp. 381-386, 2020. http://dx.doi.org/10.21275/ART20203995.

[3] H. P. M. Wai, P. P. Tar, and P. Thwe, “Ontology based web page classification system by using enhanced C4.5 and naïve Bayesian classifiers,” In 2018 International Conference on Intelligent Informatics and Biomedical Sciences, (ICIIBMS 2018), Bangkok, Thailand, 21-24 October 2018, IEEE, pp. 286-291. https://doi.org/10.1109/ICIIBMS.2018.8549994.

[4] A. Ramírez Noriega, R. Juárez Ramírez, J. J. Tapia, V. H. Castillo, and S. Jiménez, “Construction of conditional probability tables of Bayesian networks using ontologies and Wikipedia,” *Comput. Sist.*, vol. 23, no. 4, pp. 1275-1289, 2019. http://dx.doi.org/10.13053/cys-23-4-2705.

[5] R. Pan, Z. Ding, Y. Yu, and Y. Peng, “A Bayesian network approach to ontology mapping,” In International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Berlin: Springer, pp. 563-577. https://doi.org/10.1007/11574620_41.

[6] S. Li, S. Chen, and Y. Liu, “A method of emergent event evolution reasoning based on ontology cluster and Bayesian network,” *IEEE Access*, vol. 7, pp. 15230-15238, 2019. https://doi.org/10.1109/ACCESS.2019.2894777.

[7] Y. Wen, Y. Zhang, L. Huang, C. Zhou, C. Xiao, F. Zhang, X. Peng, W. Zhan, and Z. Sui, “Semantic modelling of ship behavior in harbor based on ontology and dynamic Bayesian network,” *ISPRS Int J. Geo-Inform.*, vol. 8, no. 3, pp. 107-107, 2019. https://doi.org/10.3390/ijgi8030107.

[8] R. Kühn, J. Mitrovic, and M. Granitzer, “Grhoot: Ontology of rhetorical figures in German,” In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 2022, European Language Resources Association, pp. 4001-4010.

[9] H. El Massari, N. Gherabi, S. Mhammedi, Z. Sabouri, and H. Ghandi, “Ontology-based decision tree model for prediction of cardiovascular disease,” *Indian J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 851-859, 2022. https://doi.org/10.21817/indjcse/2022/v13i3/221303143.

[10] M. Mao, Y. Peng, and M. Spring, “An adaptive ontology mapping approach with neural network based constraint satisfaction,” *J. Web Semant.*, vol. 8, no. 1, pp. 14-25, 2010. https://doi.org/10.1016/j.websem.2009.11.002.

[11] C. Breen, L. Khan, and A. Ponnusamy, “Image classification using neural networks and ontologies,” In Proceedings 13th International Workshop on Database and Expert Systems Applications, Aix-en-Provence, France, 10 December 2002, IEEE, pp. 98-102. https://doi.org/10.1109/DEXA.2002.1045883.

[12] P. Hohenecker and T. Lukasiewicz, “Ontology reasoning with deep neural networks,” *J. Artif. Intell. Res.*, vol. 68, pp. 503-540, 2020. https://doi.org/10.48550/arXiv.1808.07980.

[13] L. Korel, A. S. Behr, N. Kockmann, and M. Holena, “Using artificial neural networks to determine ontologies most relevant to scientific texts,” CEUR Workshop Proceedings, vol. 3226, pp. 44-54, 2022.

[14] L. Zhu, G. Hua, H. M. Baskonus, and W. Gao, “Svm-based multi-dividing ontology learning algorithm and similarity measuring on topological indices,” *Front. Phys-Lausanne*, vol. 8, Article ID: 547963, 2020. https://doi.org/10.3389/fphy.2020.547963.

[15] L. R. Rabiner and B. H. Juang, “An introduction to hidden markov models,” *IEE ASSP Mag.*, vol. 3, no. 1, pp. 4-16, 1986. http://dx.doi.org/10.1109/MASSP.1986.1165342.

[16] S. Iloga, O. Romain, and M. Tchuenté, “An accurate hmm-based similarity measure between finite sets of histograms,” *Pattern. Anal. Appl.*, vol. 22, pp. 1079-1104, 2019. http://dx.doi.org/10.1007/s10044-018-0734-z.

[17] Y. Htet, T. T. Zin, P. Tin, H. Tamura, K. Kondo, and E. Chosa, “HMM-based action recognition system for elderly healthcare by colorizing depth map,” *Int J. Env. Res. Pub. He.*, vol. 19, no. 19, Article ID: 12055, 2022. https://doi.org/10.3390/ijerph191912055.

[18] F. Liu, H. Zhang, X. Li, Z. Li, and H. Wang, “Intrusion identification using GMM-HMM for perimeter monitoring based on ultra-weak FBG arrays,” *Opt. Express*, vol. 30, no. 10, pp. 17307-17320, 2022. https://doi.org/10.1364/OE.452418.

[19] R. Srivastava and D. Pandey, “Speech recognition using HMM and soft computing,” Mater. Today: Proc., vol. 51, pp. 1878-1883, 2022. https://doi.org/10.1016/j.matpr.2021.10.097.

[20] Z. Su and B. Yi, “Research on HMM-based efficient stock price prediction,” Mob. Inf. Syst., vol. 2022, Article ID: 8124149, 2022.

[21] A. Valarakos, G. Sigletos, V. Karkaletsis, and G. Paliouras, “A methodology for semantically annotating a corpus using a domain ontology and machine learning,” In Proceedings of the International Conference in Racent Advances in NLP, Borovest, Bulgaria, September 10-12, 2003, Berlin: Springer, pp. 495-499.

[22] E. Prestat, M. M. David, J. Hultman, N. Tas, R. Lamendella, J. Dvornik, R. Mackelprang, D. D. Myrold, A. Jumpponen, S. G. Tringe, E. Holman, K. Mavromatis, and J. K. Jansson, “Foam (functional ontology assignments for metagenomes): a hidden markov model (hmm) database with environmental focus,” *Nucleic Acids Res.*, vol. 42, no. 19, pp. e145-e145, 2014. http://dx.doi.org/10.1093/nar/gku702.

[23] T. L. Packer and D. W. Embley, “Unsupervised training of hmm structure and parameters for ocred list recognition and ontology population,” *ACM Trans. Knowl. Discov. Data.*, vol. 22, pp. 23-30, 2015. http://dx.doi.org/10.1145/0000000.0000000.

[24] F. J. Azanzi and G. Camara, “Knowledge extraction from source code based on hidden Markov model: application to epicam,” In 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications, Hammamet, Tunisia, March 12, 2018, IEEE, pp. 1478-1485. http://dx.doi.org/10.1109/AICCSA.2017.99.

[25] P. Monika and G. Raju, “Effective model integration algorithm for improving prediction accuracy of healthcare ontology,” In International Conference on Computational Vision and Bio Inspired Computing, January 7, 2020, Springer, pp. 1203-1212. http://dx.doi.org/10.1007/978-3-030-37218-7_126.

[26] S. Bratus, A. Rumshisky, A. Khrabrov, R. Magar, and P. Thompson, “Domain-specific entity extraction from noisy, unstructured data using ontology-guided search,” *Int J. Doc. Anal. Recog.*, vol. 14, no. 2, pp. 201-211, 2011. http://dx.doi.org/10.1007/s10032-011-0149-5.

[27] H. Mi, A. Muruganujan, J. T. Casagrande, and P. D. Thomas, “Large-scale gene function analysis with the panther classification system,” *Nat. Protoc*, vol. 8, no. 8, pp. 1551-1566, 2013. http://dx.doi.org/10.1038/nprot.2013.092.

[28] A. Pipitone and R. Pirrone, “A hidden Markov model for automatic generation of er diagrams from owl ontology,” In 2014 IEEE International Conference on Semantic Computing, Newport Beach, CA, USA, 25 August 2014, IEEE, pp. 135-142. http://dx.doi.org/10.1109/ICSC.2014.19.

[29] M. Rani, R. Vyas, and O. Vyas, "Opaesfh: Ontology-based personalized adaptive e-learning system using FPN and HMM," In TENCON 2017-2017 IEEE Region 10 Conference, Penang, Malaysia, 21 December 2017, IEEE, pp. 2441-2446. http://dx.doi.org/10.1109/TENCON.2017.8228271.

[30] N. Karmegam, "An HMM and ontology based cross-lingual question answering system for the agricultural domain," PhD Dissertation, Informatics Institute of Technology, Westminster University, UK, 2019.

[31] V. Elangovan, "Indoor group activity recognition using multi-layered hmms," *Comput. Vis. Patt. Recog.*, vol. 2021, pp. 1-8, 2021. https://doi.org/10.48550/arXiv.2101.10857.

[32] B. Muthukumar, K. Sathiyamurthy, and A. Jagan, "Semantic based security for health data over sensor network," *J. Electr. Eng.*, vol. 21, no. 1, pp. 13-13, 2021.

[33] L. Warda, "Modular ontology design, a state-of-art of diseases ontology modelling and possible issue," Revue d'Intelligence Artificielle, vol. 3, no. 3, pp. 497-501, 2022. https://doi.org/10.18280./ria.360319.

[34] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them," IEEE Intell. Syst. Th. App., vol. 14, no. 1, pp. 20-26, 1999. http://dx.doi.org/10.1109/5254.747902.

[35] R. Subhashini and J. Akilandeswari, "A survey on ontology construction methodologies," *Int J. Enterp. Comput. Bus. Syst.*, vol. 1, no. 1, pp. 60-72, 2011.