International Journal of Decision Support System Technology
Volume 13 • Issue 2 • Quarterly 2021

A Semantic Knowledge-Based Framework for Information Extraction and Exploration

Abduladem Aljamel, Misurata University, Libya
https://orcid.org/0000-0002-7289-749X
Taha Osman, Nottingham Trent University, UK
Dhavalkumar Thakker, University of Bradford, UK

ABSTRACT

The availability of online documents that describe domain-specific information provides an opportunity in employing a knowledge-based approach in extracting information from web data. This research proposes a novel comprehensive semantic knowledge-based framework that helps to transform unstructured data to be easily exploited by data scientists. The resultant sematic knowledgebase is reasoned to infer new facts and classify events that might be of importance to end users. The target use case for the framework implementation was the financial domain, which represents an important class of dynamic applications that require the modelling of non-binary relations. Such complex relations are becoming increasingly common in the era of linked open data. This research in modelling and reasoning upon such relations is a further contribution of the proposed semantic framework, where non-binary relations are semantically modelled by adapting the semantic reasoning axioms to fit the intermediate resources in the N-ary relations requirements.

KEYWORDS

Information Extraction, Knowledge Representation, Knowledge-Based Approach, Machine Learning, Natural Language Processing, Non-Binary Relations, Open Linked Data, Semantic Web Technologies

1. INTRODUCTION

An increasing amount of data is being made available online. It can be exploited to inform data analytics and Decision Support Systems (DSS) for a variety of applications such as those belonging to the financial services domain. However, this online data is diverse in terms of volume and complexity, is largely unstructured and constructed in natural human languages. This makes the manual exploitation of this data by end users very difficult. Therefore, automated Information Extraction (IE) techniques are needed in order to extract useful information to be represented in a machine understandable semantic model. However, the task of transforming the largely informative unstructured text into structured knowledgebase that can be reasoned upon to infer new knowledge or predictions or decisions of interest to a specific beneficiary group is very complex. Addressing that complexity requires in-depth expertise in utilising and integrating various methods and technologies associated with Natural Language Processing (NLP), knowledge representation and Machine Learning (ML). Recently, the advantage of the achievements in the field of Semantic Web Technologies (SWT) have been extensively used in data analytics and decision-support systems in several application domains such as financial investment recommendation, a clinical management, system audit management,

DOI: 10.4018/IJDSST.2021040105

network security management, justice and legal advice, waste-water management, power consumption management and electronic issue management.

As a result, there is a pressing need for a comprehensive framework that offers an intelligent roadmap for aligning the discrepancies in knowledge presentation by various contributing information sources and deliver intelligent query methods against that extracted information and its semantic model. Such framework, in authors' view, should benefit from knowledge of the problem domain that can assist the fundamental tasks of NLP, which are Named Entity Recognition (NER) and Relation Extraction.

Domain Knowledge is knowledge about a specific field/domain of interest or subject that are understood by practitioners in that field/domain of expertise. Compiling this knowledge requires in-depth analysis of the problem domain characteristics. These characteristics could be about the grammar and the meaning of words in the context of a sentence structure or style of the language of the domain. It is crucial to comprehend these characteristics to allow engineering them as linguistic or structural features. These features can then employed in the implementation of IE systems using a variety of approaches such as rule-based or ML based (Aljamel, 2018).

In this paper, a knowledge-based framework is proposed that is based on the authors extensive research and development efforts in building a knowledge-driven financial recommender system (Aljamel, 2018). The framework adopts SWT for domain knowledge representation because they can be utilised to represent the problem domain in a highly structured knowledge model (ontology) that enables software agents to comprehend domain-related information, and thus assist in automating the extraction of concepts and relations of relevance to the domain-of-interest. The semantic ontology is formally expressed using the standardised Semantic Web languages, which are Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL), and facilitate the inference of new facts from the extracted and semantically-tagged information to support decision-making and knowledge exploration activities. Furthermore, because the targeted domain-specific knowledge is heavily represented by non-binary relations, the authors have investigated how to represent these relations in the domain-specific ontology model by using N-ary relation patterns (Hogan, 2020).

The proposed knowledge-based framework presents a comprehensive methodology for IE and exploration that comprises the processes of analysis and modelling of the domain knowledge, extracting information from unstructured data, constructing the semantic knowledgebase, enriching the semantic knowledgebase and lastly exploiting the resulting semantic Knowledgebase by intelligently exploring and processing it to support the decision making. Delivering these processes requires the integration of several diverse technologies including NLP, knowledge representation, ML and, evolutionary optimisation algorithms.

The rest of this paper is organised as follows. In section 2, some of the related works are reviewed. Section 3 introduces the proposed framework overview including the framework's motivation scenario and phases. Section 4 presents the first phase of the framework, analysing and modelling the problem domain knowledge. Section 5 presents the second phase of the framework, extracting the relevant information from unstructured data for the given problem. Section 6 presents the third phase of the framework, constructing, enriching and reasoning the semantic knowledgebase. Section 7 presents the fourth phase of the framework, applying reasoning techniques and exploiting the semantic knowledgebase. The roles of domain experts and knowledge engineers in implementing the knowledge-based framework phases are described in section 8. Finally, section 9 contains conclusions, contributions.

2. RELATED WORKS

Recently, the research community has widely acknowledged the use of SWT for knowledge representation when exploiting knowledge bases for a diversity of applications. However, as

Buranarach, et al. (2016) argue, Semantic Web based applications require frameworks to assist developers build these applications and reduce the development efforts. According to them, these applications are relatively limited and there is not enough structured information in the majority of domains. These frameworks describe the required technologies, their functionalities, and the dependencies between these technologies.

Several studies have been conducted to investigate developing frameworks based on SWT for different problem domains. For example, Wanner, et al. (2015) investigated whether the ontologies in SWT can be exploited as a core of DSS in the sense that all functions of the systems operate on ontologies which are designed to serve all modules of the system. To answer their questions, the authors proposed an environmental DSS model with an ontology-based knowledgebase as its integrative core. This system is designed to delivery environmental information for personalised decision support to a variety of different users. Environmental information webpages discovery is performed by using domain-specific search techniques. The retrieved webpages include both textual passages such as pollutant concentrations and images such as graphs and heat-maps. The discovered information includes environmental background knowledge, the characteristic features of the profile of the user, the formal description of the user request and measured or forecasted environmental data. This information is represented in a semantic knowledgebase by using SWT, ontology. This representation encodes all knowledge that is involved in a uniform format and allows applying advanced reasoning techniques on it. The architecture of the proposed Semantic Web based DSS consists of three modules and the ontology-based KB as its core. The three modules are formulation of the problem, data processing, and decision support. According to the authors, the system provides high quality environmental information for personalised decision support.

In a different study, the authors Simeonov, et al. (2016) present a decision support framework for Small and Medium Enterprises internationalisation indicators based on inference over semantically integrated data from heterogeneous web resources as a guidance to these enterprises to develop a Decision Support System for their potential investments. The authors defined internationalisation indicators for these enterprises to provide a comparative view of the countries in question and show insights based on these indicators. They grouped the indicators into four categories, products such as Product Balance, economy such as GDP growth rate, politics such as Political Stability Index and social such as Human Development Index. The information of these indicators is retrieved from semi-structured sources of specific websites such as Eurostat and WorldBank, and a specific database such as United Nations commodity trade statistics. The extracted information is represented in RDF triples by using SWT. Then, the RDF data is stored in Ontotext GraphDB. The Decision Support System is composed of these main components: Indicator information mining from the web, semantic integration of this data in a semantic knowledgebase and the decision support mechanism. According to the authors, the results of the performed evaluation show the potential of the SWT based tool in the market.

The task of transforming the unstructured online data into a machine understandable structured knowledge to satisfy the information need of variety of applications and services is very complex. This complexity regards the need of utilising and integrating various techniques and approaches in IE and Knowledge Representation fields. As a result, there is a pressing need for a comprehensive framework that offers a roadmap for aligning these techniques and approaches. It can be concluded from this literature survey that despite the enthusiasm of the research community about the Semantic Web, more effort is required to develop a unifying framework that facilitates the interoperation of intelligent agents or reasoning engines. In addition, this framework should present a semantic knowledgebase of extracted and modelled information and deliver intelligent query methods against that semantic knowledgebase. In this research, developing knowledge-based framework will be investigated to integrate exploiting semantic knowledge bases and supporting decision-making activities in specific domains.

3. FRAMEWORK OVERVIEW

According to Buranarach, et al. (2016), there are two main specific issues that are related to the implementation of semantic knowledgebase applications, semantic data publishing and semantic data consumption processes. The issue of semantic data publishing is about how to transform any kind of unstructured data into structured data and interlinking it to existing semantic datasets. The issue of semantic data consumption process is about how to discover, access and explore the structured data. In addition, the Semantic Web standards and technologies are mature enough to establish knowledge-based applications; nevertheless, they argue that these applications are relatively limited and there is not enough structured information in the majority of domains. The proposed framework is for analysing and modelling the problem domain knowledge, extracting information from unstructured data in the problem domain knowledge, constructing semantic knowledgebase, enriching the resultant knowledgebase by sourcing semi-structured and structured sources, and exploring the resultant semantic knowledgebase to support knowledge exploration in the context of decision-making activities as a use-case motivation scenario.

3.1 The Framework Motivating Scenario

This research work uses the financial information exploration and stock investment decision-making activities as motivating use-case for the proposed framework implementation. This use-case prompts the development of developing a knowledge-based application to assist in stock investment decision making.

Investors should be able to perform decision-making analysis for stock investment including analysis that describe the economic situation in the particular country and its potential influence on the profitability of stocks, the financial analysis of the individual companies from the shareholder approach and the companies' online news analysis to estimate the future earnings and profits that affect their shares prices. Logically, predicting the companies' performance changes in macroeconomic environment must be analysed first, otherwise the inconsistent assumptions could be drawn (Levišauskait, 2010; Li et al., 2014). The overview of the framework implementation scenario is depicted in Figure 1.

As illustrated in the diagram, the focal point of the framework is the semantic knowledgebase that is constructed by sourcing domain-relevant unstructured, semi-structured and structured data, which is aligned with the target problem domain based on a concept map (ontology. Based on the user query, the framework facilitates the intelligent exploration of the data stored in the knowledgebase or engages in rule-based reasoning to infer new knowledge that assists the decision-making process.

In achieving the above-mentioned objectives, the framework phases and tasks are highlighted with clear illustration on their respective functionality in the following section.

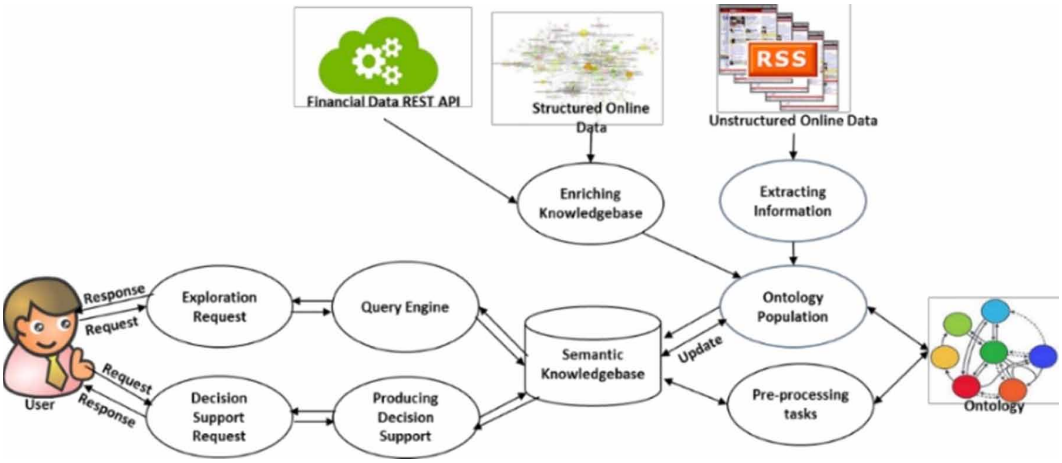
3.2 The Framework's Phases and Tasks

This framework aims to source data from structured, semi-structured and unstructured resources and aligning this complex and diverse data along a domain-specific knowledgebase that can be intelligently explored by end users. The objectives of the framework are implemented by tasks that use a diversity of algorithms, methods, approaches, techniques and tools, which are mainly related to these four disciplines, NLP, SWT, ML techniques and Evolutionary Algorithms. These tasks can be categorised into four main phases as detailed below.

3.2.1. Phase One (Analysing and Modelling the Domain Knowledge)

Analysing the problem domain to capture the syntactic and semantic characteristics to construct the knowledge map and then translating it into a formal semantic model, ontology.

Figure 1. The overview of the framework implementation scenario



3.2.2. Phase Two (Natural Language Pre-Processing, NER and Relation Classification)

Applying the Natural Language pre-Processing and NER tasks for Relation classification including relation detection, features extraction and training datasets composition then creating and applying the relations classifiers to extract relations between the targeted Named Entities. The created relation classifiers are configured and optimised by applying features selection.

3.2.3. Phase Three (Constructing and Enriching the Semantic Knowledgebase)

The optimised relation classifiers are applied on unlabelled data to recognise the Named Entities and their interrelations. Then, the recognised Named entities and their interrelations are populated into semantic knowledgebase with respect to its ontology. The last task in this phase is enriching the resulting knowledgebase by utilising public available datasets to be used to publish ground facts that are relevant to the target problem domain.

3.2.4. Phase Four (Applying Reasoning Techniques and Exploiting the Semantic Knowledgebase)

Investigate the application of Semantic Web reasoning techniques on the resulting knowledgebase in order to extract new and interesting facts to improve Intelligent Exploration of the semantic knowledgebase and to support the decision-making process.

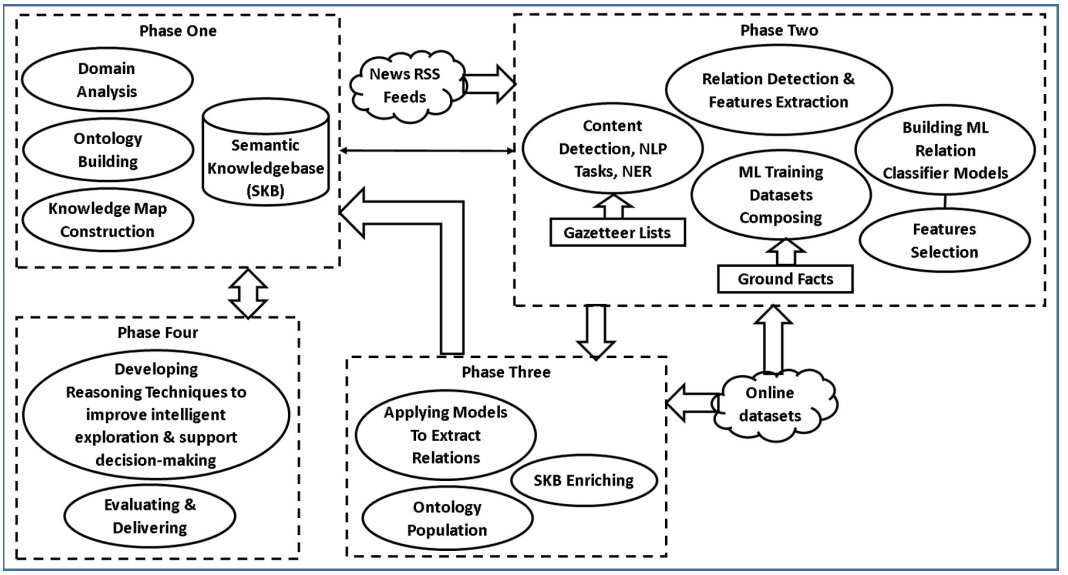
These phases are depicted in Figure 2.

The framework phases described above can be grouped into two types, tasks that can be applied on whatever the domain is and tasks that should be configured to fit every specific domain; for example, analysing the domain to specify the key domain concepts and their interrelations; accordingly, composing the training datasets for relation classification models. Nevertheless, the phases can be applied to any domain. The following sections describe the phases of the proposed knowledge-based framework phases.

4. ANALYSING AND MODELLING THE PROBLEM DOMAIN KNOWLEDGE

In this phase, the problem domain is analysed to capture the syntactic and semantic characteristics to construct the knowledge map and then translate it into a formal semantic model by utilising the SWT. The scope of the semantic model should be limited to knowledge about a particular domain of

Figure 2. The Concept Map sample of the problem domain of this work



interest rather than covering a broad range of related topics. The narrower the scope of the ontology model for the domain, the more semantic model engineering can focus on logic-based constraints to describe the details in that domain. The semantic model in this effort targets financial domain as a use-case to propose a knowledge-based framework. Specifically, the semantic model is about formulising or modelling the decision-making problem, which is the stock investment decision-making problem. This phase can be represented by reasoning module in DSS, which implements the decision support strategy. Domain conceptualisation or building the domain's knowledge map aims to create a prearranged vocabulary and semantic structure for exchanging information about that domain. The domain knowledge is modelled in terms of the problem domain's key concepts, their interrelations and the characteristics of the data as well as the interaction with the target beneficiary groups (Grimm, 2010).

As shown in Figure 3, the target domain knowledge is structured as a map of interrelated concepts that can be easily revised and improved by both the domain experts and knowledge engineers.

To perform the Semantic Web knowledge representation, ontology, formalised Semantic Web languages are employed. The main requirements of these languages are well defined syntax, efficient reasoning support, formal semantics, sufficient expressive power and convenience of expression. The Semantic Web languages include RDF, RDFS and OWL. The specifications of these languages are standardised and recommended by the World Wide Web Consortium (W3C). The concept model was implemented as a formal ontology model using OWL (Web Ontology Language) as the knowledge representation language (Allemang & Hendler, 2011).

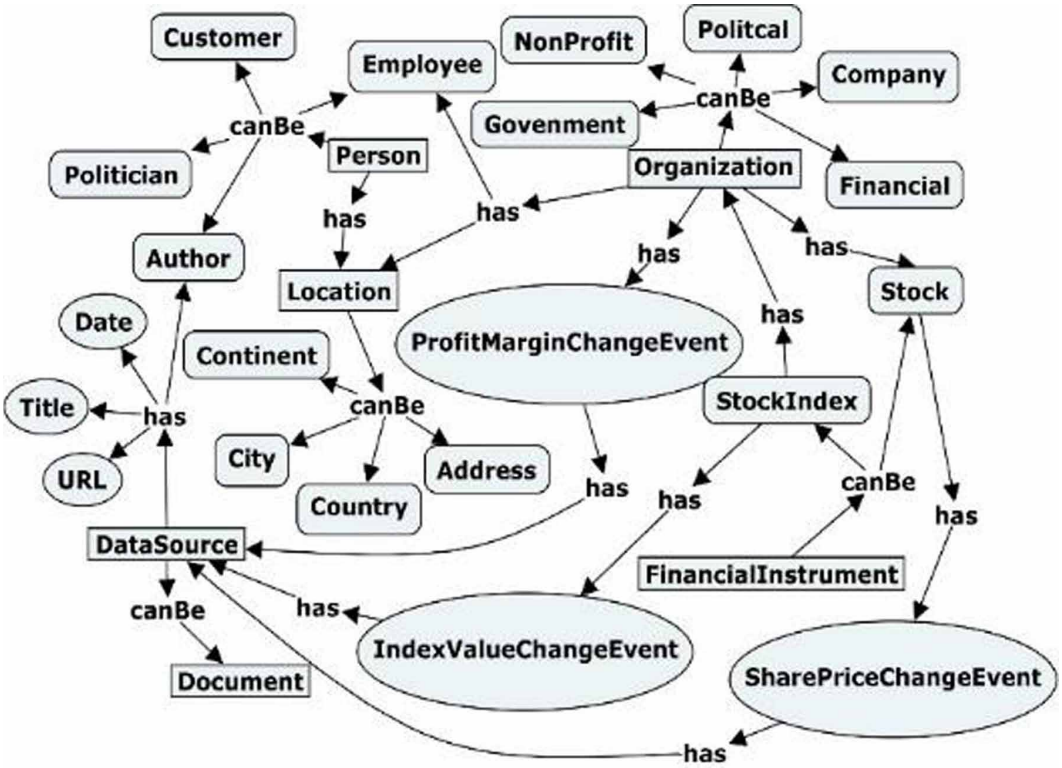
The basic structure of OWL is an RDF triple of the form:

(Subject → Predicate → Object)

The triple structure provides natural semantic units representation and it can be mapped to extracted relations from the problem domain knowledge (Grimm, 2010).

OWL provides an expressive language for defining ontologies to capture and formally represent the semantics of domain knowledge. For example, it allows the expressing of individuals equality

Figure 3. The four phases of the general framework



(owl:sameAs), the expressing of equivalent or disjoint classes and properties (owl:equivalentClass, owl:equivalentProperty, owl:disjointWith, owl:propertyDisjointWith), or the expressing of distinguishing between resource and literal values properties, (owl:DatatypeProperty) and (owl:ObjectProperty) (Polleres, Hogan, Delbru, & Umbrich, 2013).

The target use case belongs to a problem domain that have a rich set of non-binary relations. Despite several research efforts, the semantic representation of non-binary (n-ary) relations is not fully resolved. In next section, the modelling and implementation of those relations are investigated.

4.1 Non-Binary Relations Problem

The de-facto semantic knowledge representation Language, OWL, adopts the RDF standard for describing relations in Description Logic style.

Predicate (Subject, Object) (1)

The above RDF triple model can be used to represent relations with just unary and binary predicates. A common problem in data modelling occurs when it is necessary to make statements about relationships as exemplified by the ground facts below from the problem domain.

kbfwo:shareDecreasedBy(kbfwo:apple, “5.86”^^xsd:string)
kbfwo:shareDecreaseDate(kbfwo:apple, “Friday, 12/12/2016”^^xsd:datetime)

Where "kbfwo" is a prefix of the name space of this research ontology. It stands for Knowledge-Based FrameWork Ontology.

The two triples above state that the stock price of Apple company has decreased by 5.86 percent on Friday, 12/12/2016. Assuming that Apple has another share decrease by 1.5 percent on Monday 26/12/2016 as described by RDF triples below:

kbfwo:shareDecreasedBy(kbfwo:apple, "1.5"^^xsd:string)
kbfwo:shareDecreaseDate(kbfwo:apple, "Monday, 26/12/2016"^^xsd:datetime)

It is clear that there is no link between the dates and the price decrease statements in those triples, therefore, for instance, the date Friday, 12/12/2016 could be associates with either the 5.86 decrease or for 1.5 decrease. Moreover, it is difficult to add more details about these facts such as the source of these facts or add details related to the IE technique used to extract them.

The problem of logically representing facts that involve more than two entities, usually called N-ary relations, is a known issue in formal languages as it is the case in Semantic Web languages and most Description Logics (Hoekstra, 2009; Krieger & Willms, 2015; Segaran, Evans, & Taylor, 2009). The non-binary relations could be represented in a general form as below:

$$predicate(subject_1, subject_2, subject_3, ..., subject_m, object_1, object_2, object_3 ....., object_n) \quad (2)$$

Form (2) above is a general form that can be simplified to represent the common relation between one subject and several objects as in the form (3) below:

$$predicate(subject, object_1, object_2, object_3, ... object_n) \quad (3)$$

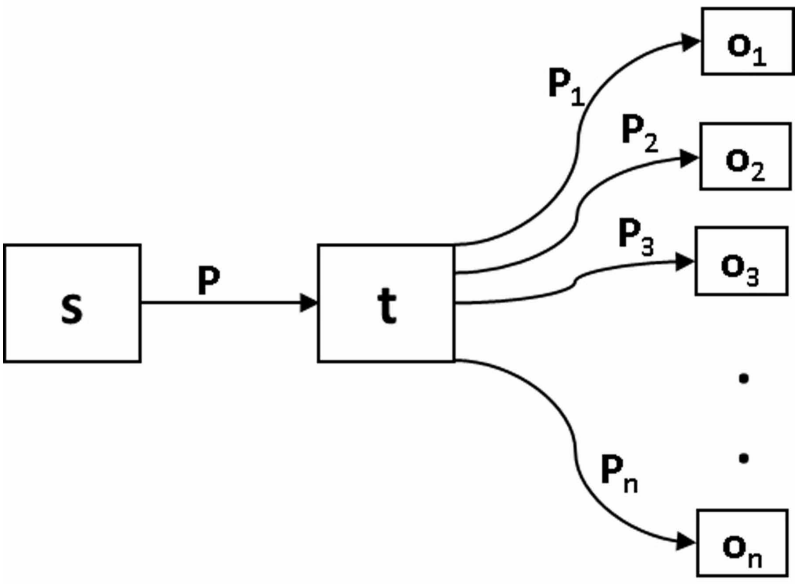
The two common approaches for N-ary relation modelling are reification and relation-as-class. Reification is the process of representing (subject-predicate-object) statement as a subject in other statements. Although the RDF standard supports reification and it has a built-in vocabulary for reifying triples, the instances of these vocabularies and relations are designed to add more information about triples rather than relations. This additional information affect OWL and RDFS reasoners and increases the complexity of the ontology which leads to a complexity in querying the resulting RDF data. A potential disadvantage of RDF reification is that there is no connection between the original statement and the reified statement. If one of them is modified, the other is not automatically modified and therefore W3C no longer supports the reification approach (Noy, Rector, Hayes, & Welty, 2006; Vinu, Sherimon, Krishnan, & Saad Takroni, 2014).

The relation-as-class N-ary relation pattern is about creating an intermediate resource to represent the original or main N-ary predicate as a class with "N" properties that provides additional information about the relation instance rather than the triple (or statement) itself. Individual instance of that classes corresponds to instances of the relation. Additional properties provide binary links to each argument of the relation. In this pattern solution, the N-ary relation is transferred into multi-binary relations (Hoekstra, 2009; Noy et al., 2006). To illustrate this pattern, return back to the pattern that is introduced in the form number (3) above. It can be represented in terms of the intermediate resources and the arguments of the relation as in the form number (4) below:

$$P(s, o1, o2, \ldots, on) \Rightarrow P(s,t) \times P1(t,o1) \times P2(t,o2) \times P3(t,o3) \times \ldots\ldots\ldots \times Pn(t,on) \quad (4)$$

Where:

Figure 4. N-ary relation pattern



P: the main predicate of the N-ary Relation.
s: the subject individual member of the domain class of the main predicate of the N-ary relation
t: an intermediate individual member of the intermediate class of the N-ary relation. Every N-ary relation has its own relation class to generate intermediate individual for every N-ary relation.
o1, o2,….. on: the objects individual members of the range classes of the properties that are participate in the N-ary Relation. Each individual represents an argument of the N-ary relation.
P1, P2, ….. Pn: the properties of the binary relations used to represent the N-ary relation as a multi-binary relation.

Graphically, N-ary relation patterns can be represented as in Figure 4:

Also, the form number (4) above can be expressed in terms of domain and range classes as in form number (5) below.

$$P(C, D1, D2, ..., Dn) \Rightarrow P(C, RC) \times P1(R_{C,} D1) \times P2(RC, D2) \times P3(_{R}C, D3) \times \ldots\ldots_{.} \times Pn_{(}RC, Dn) \quad (5)$$

Where:

P: the main predicate of the N-ary Relation.

C: a domain class for P predicate.
P1, P2, ….. Pn: the properties of the binary relations used to represent the N-ary relation as multi-binary relation.
RC: an intermediate class of the N-ary relation. It is a range class for the main predicate P and the domain class for all other properties of the binary relations, P1, P2, ….. Pn

D1, D2,….. Dn: the range classes for all properties of the binary relations, P1, P2, ….. Pn

4.2 The Proposed Approach to Implementing N-ary Relation Pattern

The above ground facts of Apple's shares increase could be formulated by using the simplified N-ary form number (3) above to be as in the relation form below:

kbfwo:sharePriceChange (kbfwo:apple, "5.86%"^^xsd:string, "Friday, 12/12/2106"^^xsd:datetime)

Where (kbfwo:sharePriceChange) is the main N-ary predicate.

Also, the online news document source details of the ground fact could be added. In addition to that resources are richly described in N-ary relations, adding information about data sources such as authorship of a data, its currency, its date and its licensing terms could encourage reusing the datasets. As highlighted by Heath and Bizer (2011), this metadata provides the consumers of the data clarity about the provenance and relevance of a datasets. The details about the data sources could support the information in the triples; for example, the date that is stated in triple will be clearer if it is related to the date of the news article. Also, there are more information about the document could be linked to the document such as the URL link, the author and the title of the document. After adding the date of the data source resources to the N-ary relation ground fact above, it will be as in the N-ary relation ground fact below:

kbfwo:sharePriceChange(kbfwo:apple, "5.86%", "Friday, 12/12/2106", kbfwo:158b_gone_the_apple)

Where, (kbfwo:158b_gone_the_apple) is the URI resource name of data source document of the ground facts triples.

This N-ary relation ground fact could be expressed in terms of domains and ranges classes of the main N-ary relation predicate. It will be as in the N-ary relation ground fact below:

kbfwo:sharePriceChange(kbfwo:Company, Literal^^xsd:string, Literal^^xsd:dateTime, kbfwo:OnlineNews)

Where (kbfwo:Company) class is the domain of the main predicate of N-ary relation and the (Literal^^xsd:string), (Literal^^xsd:dateTime) and (kbfwo:OnlineNews) are ranges.

This N-ary relation ground fact above can be modelled by using the pattern presented and explained in the form number (4) and Figure 4 by transferring *the N-a*ry relation ground fact into multi-binary ground facts. Firstly, a new Class (kbfwo:SharePriceChange) is created to represent N-ary relation's main predicate (kbfwo:sharePriceChange). Secondly, an individual of this association class is created, (kbfwo:sharedecrease_1). Then, N-ary relation subject is linked with this individual. Lastly, the individual (kbfwo:sharechange_1) is linked with the other properties' values that describe the N-ary relation. For example, the percentage value of share's decrease and the decrease's date and the data source of this information. These binary ground facts triples are shown in DL form below:

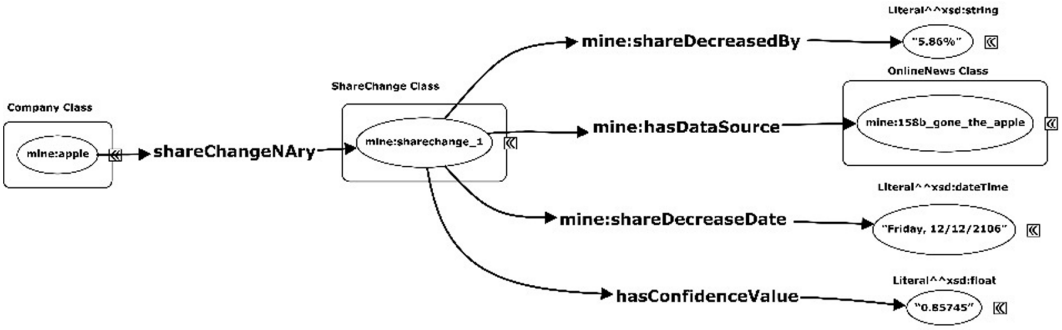
rdf:type (kbfwo:sharechange_1, kbfwo:SharePriceChange)
kbfwo:sharePriceChange (kbfwo:apple, kbfwo:sharechange_1)
kbfwo:shareDecreasedBy (minr:sharechange_1, "5.86%"^^xsd:string)
kbfwo:shareDecreaseDate (kbfwo:sharechange_1, "Friday, 12/12/2016"^^xsd:dateTime)
kbfwo:hasDataSourc (sharechange_1, kbfwo:158b_gone_the_apple)

Also, the information about the data source document could be added such as URL link, title, creator name and date as shown in the following binary ground facts:

kbfwo:hasTitle (kbfwo:158b_gone_the_apple, “$158b gone! Apple crash gets ugly”^^xsd:string)
kbfwo:hasURL (kbfwo:158b_gone_the_apple, “http://www.msn.com/..?srcref=rss”^^xsd:string)
kbfwo:hasDate (kbfwo:158b_gone_the_apple, “21/8/2015”^^xsd:dateTime)
kbfwo:hasCreator (kbfwo:158b_gone_the_apple, kbfwo:matt_krantz)

Graphically, the above N-ary relation example is depicted in Figure 5:

Figure 5. N-ary Relation Example



However, there are some considerations when introducing a new intermediate class for an N-ary relation. Firstly, meaningful names to instances of properties should be given or to the classes used to represent instances of N-ary relations. Secondly, defining inverse properties with N-ary relations. Lastly, expressing the N-ary relation in terms of OWL axioms (Noy et al., 2006). The next subsections will present these considerations in detail.

4.2.1 The N-ary Relations’ Intermediate Classes

As explained above, N-ary relation pattern requires introducing a new class for all relation properties as an intermediate class of each N-ary relation. It is recommended when introducing a new intermediate class to provide meaningful name for it, for its individual instances, and to the main predicate of the N-ary relation. The individual members of the intermediate classes are required to serve as an intermediate resource linking the subject to the objects of the N-ary relation (see form number (4) and Figure 4: N-ary r*elation pattern). In fac*t, it is common to use of blank nodes to represent these intermediate resources in the most of N-ary relations patterns representations.

However, the authors claim that these intermediate resources are important resources and they should be identified by URI reference because of two reasons. Firstly, the negative impact of blank nodes on the representation of the Semantic Web data such as the problems of merging RDF graphs and publishing Linked Data. Hogan (2020), in his book, discussed in more detail the issues arising from the presence of blank nodes in RDF datasets. According to him, the problem arises when merging data from different scopes, a blank node with the same label in two distinct scopes could naively cause a “blank node clash”. Secondly, all parts of the N-ary relations should be considered as one component and all parts of this component should have a globally resolvable name; specifically, when the N-ary relation represents an event as shown in the example in Figure 4. This claim is in agreement with the opinion of Krieger and Declerck (2015) who showed that the negative impact of

blank nodes can often be avoided by generating unique URI reference names from information that is accessible through the new individual properties. As a result, in this research implementation of N-ary relation pattern, the unique URI reference names are generated for the individual instances of the intermediate classes. These individuals' instances are accessible through the other individuals' properties of the N-ary relation.

Dynamic events are common in the dynamic financial problem domain; hence it was necessary in the proposed implementation to consider N-ary relations representation as intermediate classes for event relations. For example, the event of share price change has intermediate class (kbfwo:SharePriceChange) and has a main predicate or property for the N-ary relation (kbfwo:sharePriceChange). To clarify this example more, the sentence example is presented below:

Zoomlion's Hong Kong-traded shares closed up 3.31 percent on Monday.

This sentence is retrieved from the online news document of title "Zoomlion says 2014 profit may have fallen"

This sentence contains the following entities. The entity "Zoomlion" is for company name. The entity "3.31 percent" is a percentage number. It can be defined as a typed literal of float value "3.31". The entity "Monday" is a date value. It can be defined as a URI resource of "monday_1234567" and its correct date value can be found by using the date of the document data source as a reference. From the entities recognised in this sentence and its data source, the following individuals can be extracted:

kbfwo:zoomlion → rdf:type → kbfwo:Company
kbfwo:monday_1234567 → rdf:type → kbfwo:Date
kbfwo:zoomlion_says_2014_profit_may_have_fallen → rdf:type → kbfwo:OnlineNews

However, the fact that the shares of "Zoomlion" is increased by "3.31%" on "Monday" is presented according to its data source in N-ary relation as in following triples.

First, an intermediate individual member of the intermediate class (kbfwo:SharePriceChange) is created. This individual should be identified by URI as in the triple below:

kbfwo:sharepricechange_8901234 → rdf:type → kbfwo:SharePriceChange

Then, the binary triples that represent N-ary relation are linked to the instance of the intermediate class as follows:

kbfwo:zoomlion → kbfwo:sharePriceChange →kbfwo:sharepricechange_8901234
kbfwo:sharepricechange_8901234 → kbfwo:shareIncreasedBy → "3.31"^^xsd:float
kbfwo:sharepricechange_8901234 → kbfwo:shareIncreaseDate → kbfwo:monday_1234567
kbfwo:sharepricechange_8901234 → kbfwo:hasDataSource → kbfwo:zoomlion_says_2014_profit_may_have_fallen

4.2.2 Inverse N-ary Relations

Defining inverse properties with N-ary relations by using OWL requires more work than with binary relations. An inverse must be specified for each of the properties participating in the N-ary relation. For example, the following N-ary relation triples:

kbfwo:xerox_technology → kbfwo:employerOfNAry → kbfwo:employerofnary_1234567

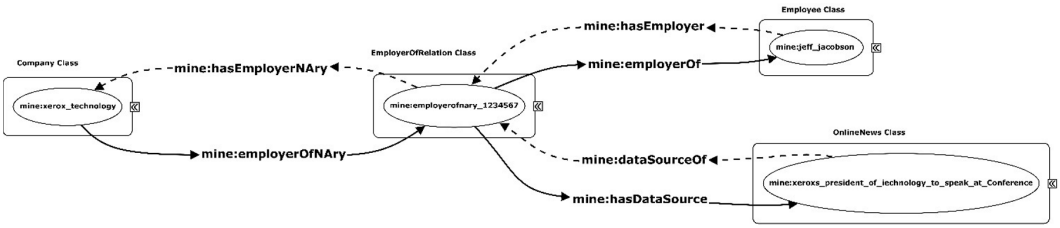
kbfwo:employerofnary_1234567 → kbfwo:employerOf → kbfwo:jeff_jacobson
kbfwo:employerofnary_1234567 → kbfwo:hasDataSource →
kbfwo:xeroxs_president_of_technology_to_speak_at_Conference

The inverse of this N-ary relation can be expressed by creating the inverse of all properties that participate in the N-ary relation, the main predicate (kbfwo:employerOfNAry) and the other properties (kbfwo:employerOf) and (kbfwo:hasDataSource). These inverse properties are:

(kbfwo:hasEmployerNAry) is an inverse of the property (kbfwo:employerOfNAry)
(kbfwo:hasEmployer) is an inverse of the property (kbfwo:employerOf)
(kbfwo:dataSourceOf) is an inverse of the property (kbfwo:hasDataSource)

The invers N-ary relation of the above N-ary relation by using the inverse properties is shown in the triples and Figure 6:

Figure 6. Inverse N-ary Relation Example



kbfwo:jeff_jacobson → kbfwo:hasEmployer → kbfwo:employerofnary_1234567
kbfwo:xeroxs_president_of_technology_to_speak_at_Conference →
kbfwo:dataSourceOf → kbfwo:employerofnary_1234567
kbfwo:employerofnary_1234567→ kbfwo:hasEmployerNAry → kbfwo:xerox_technology

It is also worth pointing out that the invers N-ary relation uses the same intermediate individual of the original N-ary relation.

4.2.3 OWL Axioms and Reasoning for N-ary Relations

kbfwo:xerox_technology → kbfwo:employerOfNAry → kbfwo:employerofnary_1234567
kbfwo:employerofnary_1234567 → kbfwo:employerOf → kbfwo:jeff_jacobson
kbfwo:employerofnary_1234567 → kbfwo:hasDataSource →
kbfwo:xeroxs_president_of_technology_to_speak_at_Conference

The Xerox Technology (kbfwo:Company class) is the employer of the Jeff Jacobson (kbfwo:Employee class) as mentioned in the online news article of title "Xeroxs President of Technology to speak at Conference" (kbfwo:OnlineNews class). The individual (kbfwo:xerox_technology) has a property (kbfwo:hasEmployerNAry) that has another object (kbfwo:employerofnary_1234567, an instance of the class (kbfwo:EmployerOfRelation) as its value. The individual (kbfwo:employerofnary_1234567) in the example represents a single object encapsulating both the employee (kbfwo:jeff_jacobson, a

specific instance of kbfwo:Employee) and the data source of the information (kbfwo:xeroxs_president_of_technology_to_speak_at_Conference, a specific instance of kbfwo:OnlineNews).

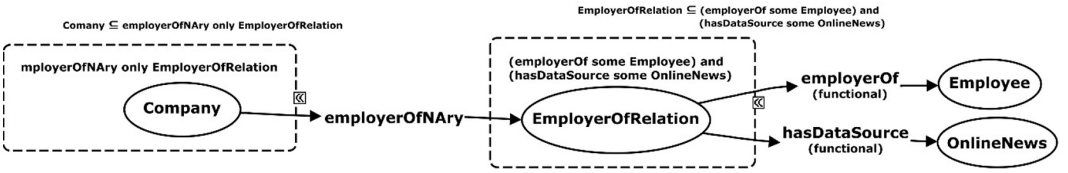
The components of the N-ary relation above contain the information held in the original sentences arguments, which are "What is the company?", "Who is the employee?" and "What is the data source of this information?". This N-ary relation example can be expressed in terms of domains and ranges classes of all properties that participate in the N-ary relation by using the form number (5) above. It is as shown in the relation from below:

kbfwo:employerOfNAry(kbfwo:Company, kbfwo:Employee, kbfwo:OnlineNews) ⇒
kbfwo:employerOfNAry(kbfwo:Company, kbfwo:EmployerOfRelation)×
kbfwo:employerOf(kbfwo:EmployerOfRelation, kbfwo:Employee)×
kbfwo:hasDataSource(kbfwo:EmployerOfRelation, kbfwo:OnlinNews)

Also, this N-ary relation can be casted into OWL axioms by representing the combination of restrictions. In the definition of the (kbfwo:Company) class, which the individual (kbfwo:xerox_technology) belongs to, a property (kbfwo:hasEmployerNAry) is specified with the range restriction going to the (kbfwo:EmployerOfRelation) class, which the individual (kbfwo:employerofnary_1234567) belongs to. The OWL restrictions should be defined on the properties of the N-ary relations. For example, both (kbfwo:employerOf) and (kbfwo:hasDatasource) have been defined as functional properties, thus requiring that each instance of (kbfwo:HasEmployerRelation) class has exactly one value for (kbfwo:Employee) class and one value for (kbfwo:OnlineNews) class. The OWL axioms of N-ary relation example are shown in the formulas below:

Company → employerOfNAry **only** EmployerOfRelation
EmployerOfRelation → (employerOf some Employee) **and** (hasDataSource **some** OnlineNews)
The axioms above are depicted in Figure 7:

Figure 7. OWL axioms for N-ary relations example



When applying reasoning tasks, the intermediate classes and their individual members should be considered; for example, if the stock holders are required to be classified in a specific class, (kbfwo:StockHolder), the OWL existential restrictions can be applyed. They are represent a property value restriction to specify (owl:Restriction) class by using (owl:someValuefrom) property restriction. Existential restrictions describe the set of individuals that have at least one specific kind of relationship to individuals which are members of a specific class. Because N-ary representation is being used, represent this restriction by using Manchester syntax will be as in the formula below:

Class: D **EquivalentTo:** P **some** (P_1**some** C)

Where P is the main N-ary relation and P1 is one of the properties that is used to link between the instances of mediate class and the instance of other class involved in the N-ary relation.

For the example above, the Manchester syntax will be as below:

Class: kbfwo:StockHolder **EquivalentTo:** kbfwo:hasStockNAry **some** (kbfwo:hasStock **some** kbfwo:Stock)

Similarly, when applying Rule-based reasoning tasks, the intermediate classes and their individual members should be considered. To make rule for N-ary relations example by using Jena rules, an example about supporting a stock investor for buying or holding stocks will be used according to some information related to the targeted company exist in the semantic knowledgebase. Suppose that the information which is exist in the semantic knowledgebase include, the targeted company for stock investment (kbfwo:microsoft), the current stock price (Price="65.22"^^xsd:float) of the targeted company. These pieces of information is represented in the semantic knowledgebase in N-ary relation pattern as in the triples below:

kbfwo:microsoft → kbfwo:hasSharePriceNAry → kbfwo:hassharepricerelation_1
kbfwo:hassharepricerelation_1 → kbfwo:hasSharePrice → "65.22"^^xsd:float
kbfwo:hassharepricerelation_1→ kbfwo:hasSharePriceDate → kbfwo:2932017_1
kbfwo:2932017_1 → kbfwo:hasDateValue → "2017-3-29"^^xsd:date

where, (kbfwo:hassharepricerelation_1) is the intermediate individual member of the intermediate class (kbfwo:HasSharePriceRelation).

The other piece of information is, the calculated intrinsic value or valuation of the stock price (Valuation="70.01"^^xsd:float) of the targeted company. These pieces of information are represented in the semantic knowledgebase in N-ary relation pattern as in the triples below:

kbfwo:microsoft → kbfwo:hasStockPriceValuationNAry → kbfwo:hasstockpricevaluationrelation_1
kbfwo:hasstockpricevaluationrelation_1 → kbfwo:hasStockPriceValuationValue → "70.01"^^xsd:float
kbfwo:hasstockpricevaluationrelation_1 → kbfwo:hasStockPriceValuationDate → kbfwo:stockpricevaluationdate_4
kbfwo:stockpricevaluationdate_4 → kbfwo:hasDateValue → "2017-3-29"^^xsd:date

where, (kbfwo:hasstockpricevaluationrelation_1) is the intermediate individual member of the intermediate class (kbfwo:HasStockPriceValuationRelation).

The investment decision will be taken according to the fact that whether the stock is under valuated or not. In other words, if the current stock price (?Price) is less than the intrinsic value of the stock (?Valuation), the decision should be to buy or hold the stock; otherwise, sell the stock. This decision can be converted into rules by using Jena rules syntax. One example of these rules will be as follows:

```
[ruleName:
(kbfwo:investorRequestID kbfwo:hasTargetedCompany ?TargetedCompany),
(?TargetedCompany kbfwo:hasSharePriceNAry ?NAryIntermediateSharePrice),
(?NAryIntermediateSharePrice kbfwo:hasSharePrice ?price),
(?TargetedCompany kbfwo:hasStockPriceValuationNAry ?NAryIntermediateValuation),
(?NAryIntermediateValuation kbfwo:hasStockPriceValuationValue ?value),
lessThan(?price, ?value)
->
```

(kbfwo:investorRequestID kbfwo:hasDecisionConclusion 'buy or keep the stock because it is under valuated.'^^xsd:string)
]

It should be noted from the rule above the variables (?NAryIntermediateSharePrice) and (?NAryIntermediateValuation), that represent the intermediate individual members of the intermediate classes.

After applying this rule one the semantic knowledgebase with the above information by the rule reasoning engine, the following statement would be derived:

kbfwo:investorRequestID → kbfwo:hasDecisionConclusion →
'buy or keep the stock because it is under valuated.'^^xsd:string

The information can be delivered to the investor by using an appropriate technique in an appropriate style.

4.2.4 Discussion

The authors have adopted the relation centred or relation-as-class pattern as a N-ary relation pattern to represent domain-specific non-binary relations in the problem domain. This pattern is about creating an intermediate resource to represent the original or main N-ary predicate as an intermediate class with "N" properties that provides additional information about the relation instance. Individual instances of that intermediate class correspond to instance of the relation. By using the intermediate resources, the N-ary relation is transferred into multi-binary relations and could allow the representation of non-binary relations work around the limitations of the direct binary predicates. Furthermore, creating an intermediate resource for the relationship allows much more flexibility in describing the relationships between resources because any number of additional properties may be used to annotate the relation in this pattern.

The authors have investigated the N-ary relation patterns considerations when introducing a new intermediate class for a relation. Firstly, meaningful names should be given to instances of properties or to the classes used to represent instances of N-ary relations. Secondly, in defining the inverse of N-ary relation, inverse properties should be defined for all properties involved in the N-ary relation. Lastly, the intermediate resources should be considered when expressing the N-ary relation in terms of OWL axioms.

In comparison to state-of-the-art N-ary relation modelling by using relation-as-class pattern, the proposed approach of N-ary relation pattern implementation does not use the blank nodes in identifying the intermediate resources of the N-ary relation. In fact, the unique URI reference names are generated for the individual's instances of the intermediate classes because these intermediate resources are important resources. They should not be identified by blank nodes because of the negative impact of blank nodes on the representation of the Semantic Web data. Moreover, all parts of the N-ary relations should be considered as one component and all parts of this component should have a globally resolvable name; specifically, when the N-ary relation represents an event.

Representing N-ary relation in semantic knowledgebase is clearly domain independent and can be applied across multiple application domains; for example, in the context of sale data analysis, relations crossing items, customers, dates, and regions can be easily acquired.

5. EXTRACTING THE RELEVANT INFORMATION FROM UNSTRUCTURED DATA FOR THE GIVEN PROBLEM

In this phase, information from unstructured data sources is retrieved and structured it into domain-specific semantic knowledgebase that can be seamlessly explored by end users.

The information retrieval process starts by recognising the named entities. The next step is the identification of identity relation between named entities (co-references resolution) before extracting the relation between the named entities in a certain event. In fact, E pipeline process is divided into two stages.

For Relation Extraction, a hybrid approach integrating Rule-based and ML based techniques was adopted. Supervised ML based approaches have been widely adopted in information extract from unstructured text, chiefly in NER and Relation Extraction (Aljamel, Osman, Acampora, Vitiello, & Zhang, 2019).

The relation classification models were further boosted by authors' implementation of Genetic Algorithms as wrapper approach to reduce the feature space for ML. Authors' implementation of GAs has resulted in significant improvement in the accuracy of ML based Relation Extraction process. Figure 8 illustrates the flow of feature subsets selections as wrapper approach.

The research and development of the NLP, NER and Relation Classification phase authors' implementation of GA Algorithms as wrapper approach are documented in authors' previous paper (Aljamel et al., 2019). The output of this phase is documents with annotated named entities and their interrelations. They will be used to construct the initial semantic knowledgebase, which will be described in the next section.

6. CONSTRUCTING AND ENRICHING THE SEMANTIC KNOWLEDGEBASE

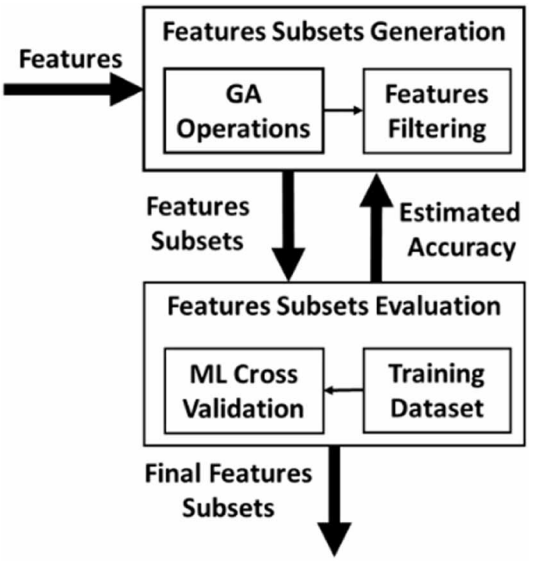
The previous sections of this research are concern about the core technologies that underpin IE and the knowledge representation. These technologies have been utilised in IE and semantic modelling the problem domain knowledge. The SWT is employed to develop semantic data model (or ontology) as a unifying structure to describe a common representation for semantics of the extracted information. Then, these technologies are utilised to construct the semantic knowledgebase from unstructured, semi-structured and structured data. Once this unifying structure for heterogeneous information sources is represented in the semantic knowledgebase, it can be exploited to improve the performance of accessing the semantic knowledgebase by developing a semantic web application (Jambhulkar & Karale, 2016).

Constructing and enriching the semantic knowledgebase tasks can be represented by information module in DSS, which allows to store the information into a semantic knowledgebase to be processed by the DSS. The information that can be used to support stock investment decision-making process could be as follows:

1. Company's information: the current stock price and the historic dividend values.
2. Country's Economic indicators: GDP, inflation and unemployment rates.
3. Company's events in online news: profit margin increase/decrease or share prices increase/decrease.
4. Other information that could be retrieved from other sources, such as companies' products and employees.

The process of constructing the knowledgebase was implemented in three stages, information retrieval, ontology population and knowledgebase enrichment. technologies in IE and semantic modelling the problem domain knowledge have been utilised.

Figure 8. GA feature subsets selection as Wrapper Approach



The main task of knowledge-based applications is performing the inference task on the semantic knowledgebase because it draws conclusions from that knowledgebase. The inference mechanism can be achieved by utilising the SWT in the knowledge-based applications to solve complex problems and to provide effective decision support (Davies, Studer, & Warren, 2006).

The details of constructing the semantic knowledgebase stages and their implementation are presented in Figure 9 and explained in next subsections.

6.1 Populating the Extracted Information

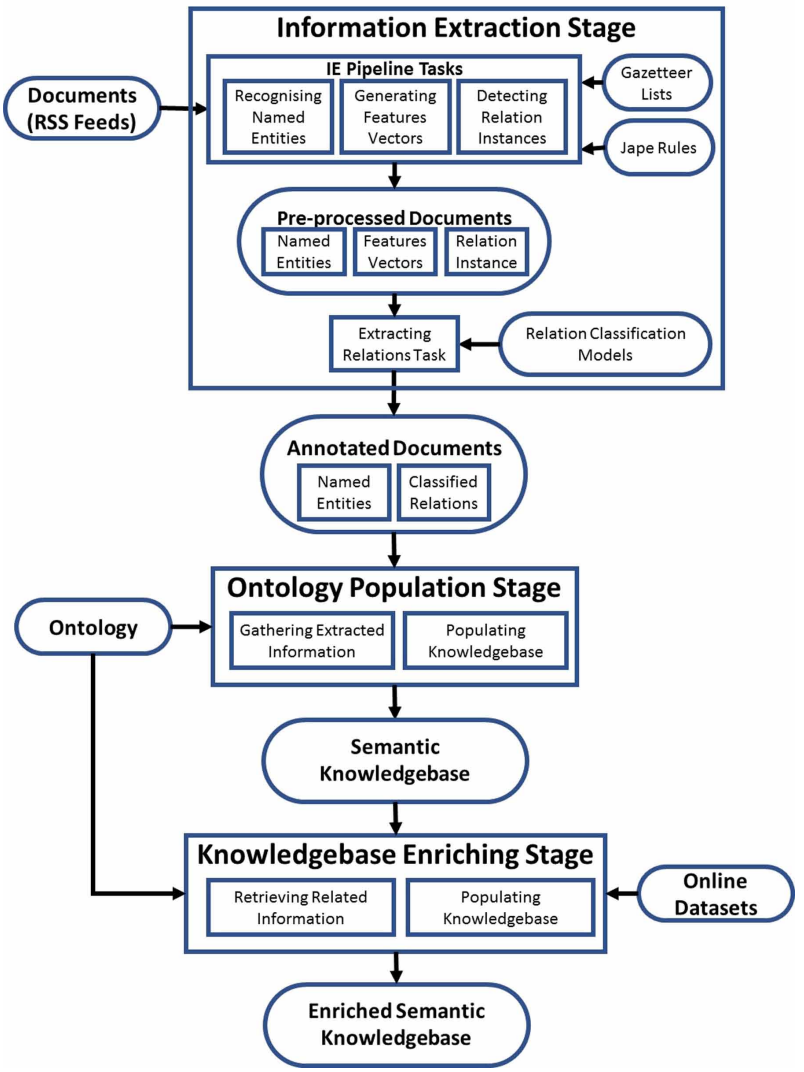
Ontology population is a knowledge acquisition activity that transforms unstructured data into instances of the concepts and relationships defined in the ontology. It is a crucial part of knowledgebase construction because it relates text to ontologies. The annotated named entities and their interrelations in the documents are transferred to the Semantic Web RDF model using authors' domain-related Semantic ontology. The named entities are related to appropriate concepts as instances in the semantic Knowledgebase. Then, mapping the relations between those named entities to the suitable property, data type or object, in the ontology as relation instances in the semantic Knowledgebase (Kumar & Zayaraz, 2015; Saat & Noah, 2016).

6.2 Enriching the Semantic Knowledgebase by Sourcing Online Datasets

Ontology Population occurs in both knowledgebase that has no instances and those already has been populated. When ontology population is performed on those that already have instances, the process is known as enriching the knowledgebase. It can be used for a variety of exploration scenarios to provide aid in a specific subject matter.

In this research, the semantic knowledgebase, which is initially populated with semantically tagged information instances that where extracted from the problem domain documents, is further enriched by utilising a diversity of structured data sources such as the Linked Open Data cloud and semi-structured data sources such as API endpoints that provide access to different economic datasets; for example, Crunchbase dataset, Yahoo Finance web service API and World Bank Linked Data endpoint. The enriched information is about the companies and their countries profiles. The formalism

Figure 9. The process of constructing semantic knowledgebase stages



in modelling the semantic knowledgebase provides a great opportunity to leverage domain-relevant facts that are published in structured and semi-structured data sets.

6.3 Domain-Specific Data Requirements for Decision-Making Process

In some domains, the retrieved information requires an additional process to apply reasoning tasks for decision-making activities such as numeric calculations for some of the decision's factors. These activities are domain-specific and required for decision-making models; however, they might not be mandatory in other domains. The new devised information is inserted directly to the knowledgebase to be used in exploration and decision-making activities (Wanner et al., 2015). In the problem domain use-case scenario, the companies' performance rates are calculated by using other existing numeric rates and inserted in the knowledgebase as appropriate. The output of this stage is an enriched semantic knowledgebase which is understandable by machines.

For storing the resultant semantic knowledgebase, the RDF triple stores such as Jena Triple Database (TDB) can be employed for storing and managing semantic facts, which are published in RDF triple model (Bunakov, 2015).

7. APPLYING REASONING TECHNIQUES AND EXPLOITING THE SEMANTIC KNOWLEDGEBASE

In this stage of the proposed framework, the semantic knowledgebase is intelligently exploited to support the stock investment decision making process by adopting a Semantic Web based method to deliver inferred facts to end users. This framework integrates the semantic knowledgebase exploration and DSS tasks. The interrogation of the Knowledgebase was according to two use-case scenarios. The first use-case scenario is that the investors request support in making a stock investment decision in a specific company. The second use-case scenario is that the investors explores the semantic knowledgebase to make the decision by themselves. The application receives and processes the user's request, responding with either the recommended decision or exploring the semantic Knowledgebase.

Figure 10 illustrates the workflow for exploring semantic knowledgebase and the integrated Decision Support System. The workflow starts with applying ontology reasoning techniques on the Semantic Knowledgebase component, user request submission component, the recommended decision production component and it ends with exploring the semantic knowledgebase.

7.1 Semantic Knowledgebase Reasoning

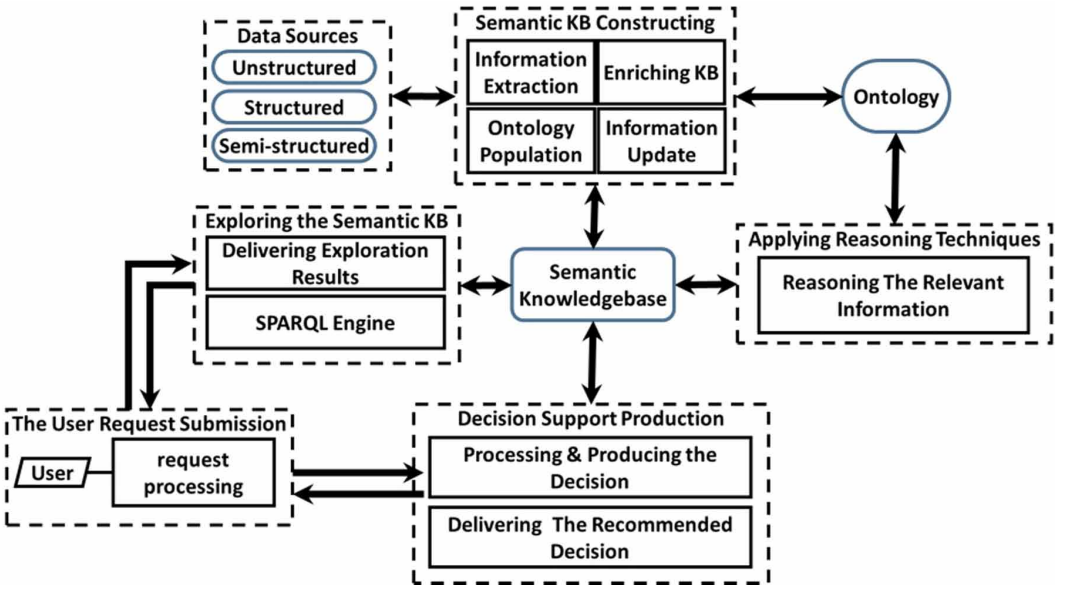
The SWT is adopted to model and reason the problem domain knowledge and develop a knowledge-based application. The use-case focuses on the finance and economic domain, which is the information about stock market investments. In addition to describing the concepts and their interrelation in the problem domain knowledge, the representation of domain knowledge by sematic ontology also facilitate reasoning, where the information in the semantic knowledgebase can be further exploited to infer new statements based on existing statements. The facilitation of reasoning on semantically-structured knowledge is at the heart of building intelligent knowledge bases that deliver sophisticated data exploration and decision support activities. The challenge of developing semantic reasoning includes compiling classification rules that are hard-wired into the ontology such as first predicate logic's Necessary & Sufficient conditions and axioms to classify events. As a part of the framework, OWL reasoning has been applied to achieve many tasks such as automatic class subsumption and automatic Individuals classification. Moreover, the rule-based reasoning was utilised to develop decision-rules based on the ontology and execute them to derive stock investment specific recommendations. Moreover, in consideration of adopting N-ary relations patterns requirements to represent non-binary relations in the problem domain, the reasoning axioms was adapted to fit the intermediate resources in the N-ary relations requirements (Wang, Zhang, Gu, & Pung, 2004).

7.2 Exploring the Semantic Knowledgebase

Intelligently exploring the semantic knowledgebase and retrieval of appropriate information are very important techniques for semantic knowledgebase applications. SWT allows for systemic and standardised modelling and compilation of knowledge for the targeted problem domain that provides deep understanding of published (processed) semantic data. In addition, SWT is capable of supporting advanced exploration scenarios and solve complex information needs such as supporting the decision-making process.

For instance, in authors' use-case scenario, where investors explore the semantic knowledgebase to make financial decisions, the knowledge interrogation component initially checks whether the requested information is available in the semantic knowledgebase. If the requested information it is available, the system presents the relevant information to the user, otherwise, the system attempts to extract that information from the relevant unstructured, semi-structured or structured data sources.

Figure 10. The main components of exploring semantic knowledgebase and the integrated Decision Support System



The query languages' specifications for exploring the semantic knowledgebase should be capable of exploring that knowledge representation standard. Due to the fact that the resultant semantic knowledgebase is constructed and stored be using Semantic Web standards, SPARQL query language was utilised to explore it. SPARQL standard being the W3C's recommendation works by allowing users to express query patterns across diverse RDF data sources to retrieve the required information. In addition, SPARQL queries was adapted to fit the intermediate resources in the N-ary relations requirements.

For example, the SPARQL query below retrieves the stock price and its date of (kbfwo:microsoft) company. The graph pattern, which is used in this query, is N-ary relation pattern because N-ary pattern is adopted in in the semantic knowledgebase.

```
SELECT DISTINCT ?StockPrice ?PriceDate
WHERE {
kbfwo:microsoft kbfwo:hasSharePriceNAry ?NAry.
?NAry kbfwo:hasSharePrice ?StockPrice .
?NAry kbfwo:hasSharePriceDate ?DateResource.
?DateResource kbfwo:hasDateValue ?PriceDate.
}
```

The query shown above would select all unique values of the variables (?StockPrice) and (?PriceDate), where there is a triple that matches any objects of (kbfwo:hasSharePrice) and (kbfwo:hasDateValue) respectively which apply the other constrains of the properties (kbfwo:hasSharePriceNAry) and (kbfwo:hasSharePriceNAry). SPARQL engine accept queries and then issue them against the semantic knowledgebase to produce a result set in either RDF or tabular form. The produced results should reflect the contents of the knowledgebase. They can be processed by the system to be presented to the user in appropriate style (Harris, Seaborne, & Prud'hommeaux, 2013). The tabular form of the result of the above query example is below:

```
-------------------------------------------------------------
----
| StockPrice                  | PriceDate                   |
```

```
=======================================
| "65.22"^^xsd:float          | "2017-3-29"^^xsd:date     |
-------------------------------------------------------------
----
```

Users can access and explore the semantic knowledgebase and retrieve RDF data by executing the SPARQL queries via SPARQL endpoints or via SPARQL engines by user interfaces of semantic web application. The query results that are returned from these endpoints or interfaces can be processed by machines to be presented to end users in an appropriate format.

8. THE ROLES OF DOMAIN EXPERTS AND KNOWLEDGE ENGINEERS IN IMPLEMENTING THE KNOWLEDGE-BASED FRAMEWORK PHASES

This framework refers to a generic architectural paradigm of IE, integration and exploitation. During the implementation of this framework, most of the problems were investigated such as the problem of optimising the features of the relation classifiers and the issue of representing the non-binary relations by using Semantic Web languages. In fact, a number of challenges in employing SWT in modelling and intelligent exploration of semantic knowledge bases were addressed including:

1. Modelling the targeted domain-specific knowledge of the sourced data into a machine-comprehensible Semantic Web ontology.
2. Transforming the extracted information into a structured data by mapping it into a semantic knowledgebase by using the semantic model, ontology.
3. Integrating the resultant knowledgebase with other semi-structured and structured data from a diversity of sources to enrich the knowledgebase.
4. Developing inference techniques to be applied on the semantic knowledgebase to infer new information and classify events that might be of importance to end users.
5. Exploiting the resultant semantic knowledgebase for intelligent exploration of information and decision-making support.

These challenges should be considered by domain experts and knowledge engineers as a roadmap for employing the SWT for the knowledge user to intelligently exploit knowledge in similar problem domain.

The primary aspects to be realised by Semantic Web engineering are, knowledge representation, knowledge accessibility and application integration; moreover, the knowledge sources quality, which is the striking features for any knowledge-based application (Hebeler, Fisher, Blace, & Perez-Lopez, 2011). Once the knowledge user needs are specified, the Semantic Web application should achieve these aspects considering the framework application requirements according to a use-case scenario.

Implementing the phases of the proposed framework requires the development and integration of processes that utilise a number of constantly evolving technologies ranging from using NLP in IE to ontology engineering and intelligent inferencing in knowledge representation. The framework allows the application developers to focus on domain problems rather than the tools, techniques and approaches of the application. The workflow and the structure of the framework's tasks makes the applicability to other domains only requires the one-off effort in constructing most of the tasks.

The usual method of constructing semantic knowledgebase in a machine understandable format involves domain experts and knowledge engineers. The domain expert analyses the problem domain knowledge to describe its characteristics including the key concepts and their interrelations, which are required to produce the knowledge map. The knowledge map is produced by domain expert alone or as a main contributor with the knowledge engineer. The knowledge engineer translates the knowledge map into a machine comprehensible and usable format and stores it in a semantic knowledgebase. This format can link this knowledgebase to other knowledge sources to be enriched. These activities

should be accomplished according to the requirements of the knowledge user. However, the knowledge engineers are required for updating the knowledge model (Bach, 2017).

9. CONCLUSION

An ever-increasing amount of data is being made available online. It can be exploited to inform data analytics and decision support systems for a variety of applications many disciplines. However, the scale and variety of the data makes manual exploitation difficult and pose significant challenges for traditional data management technologies. This paper documents the investigation and development of innovative technologies that facilitate automated information extraction.

The semantic and syntactic characteristics of domain data were exploited in improving NLP tasks associated with the instances labelling and feature generation processes in authors' implementation of ML based relation classification. In addition, the structure characteristics in knowledge modelling were exploited by translating them into a formal ontology. This ontology is required for: constructing a semantic knowledgebase from unstructured online data of a specific-domain, enriching the resulting semantic knowledgebase by sourcing semi-structured and structured online data sources, mapping that knowledge to other public datasets and employing advanced classifications and inference technologies to infer new and interesting facts about the problem domain.

The main novel outcome of this paper is the knowledge-based framework for Information Retrieval from domain-specific unstructured data. The framework contributes to the body of knowledge in modelling the problem domain into a semantically-structured knowledgebase that can be enriched by utilising a diversity of structured and semi-structured online data sources and in preparation for its exploration in the context of supporting the decision-making process. The experience in addressing the challenges of implementing the proposed knowledge-based framework were summarised to be as a road map that could be considered by domain experts and knowledge engineers as for employing SWT to intelligently explore knowledge in other and similar problem domains.

During the research implementation of the tasks of the proposed knowledge-based framework, some of the challenges and problems related to these tasks were investigated. These investigations show the importance of understanding the characteristics of the domain data in addressing these challenges. Analysing and understanding the syntactic and semantic characteristics of the domain data with SWT benefited the process of IE. The meaningful representation of data also aided the NLP tasks by improving the efficiency of the automating or semi-automating instance labelling process. For instance, in authors' implementation of ML based relations classification, domain-specific knowledge is used to compile some of authors' training datasets by drawing on relation mentions that are featured as ground facts in public online datasets such as DBPedia and Freebase. This alleviates the manual annotation effort for relation extraction, which can be a time-consuming and cumbersome task to undertake. Furthermore, this research findings highlighted that importance of using N-ary relation patterns to model non-binary relations in a variety of problem domains; this paper contributes to the modelling of non-binary relations by adapting the reasoning axioms to fit the intermediate resources in the N-ary relations requirements.

Fundamentally, this paper has proposed a domain-specific knowledge-based framework for exploiting domain knowledge in constructing a semantic knowledgebase for a target problem domain. The semantic knowledgebase allows for intelligent inference and advanced interrogation of information from the target domain. Then, a knowledge-based application was developed for investigating the implementation challenges of that framework. The evaluation of the knowledge accessibility by utilising SWT in the developed application includes the ability of data retrieval to obtain either the entire or some portion of the data from the semantic knowledgebase for a particular use-case scenario.

REFERENCES

Aljamel, A. (2018). *A Knowledge-Based Framework For Information Extraction And Exploration*. Nottingham Trent University.

Aljamel, A., Osman, T., Acampora, G., Vitiello, A., & Zhang, Z. (2019). Smart Information Retrieval: Domain Knowledge Centric Optimization Approach. *IEEE Access, 7*, 4167–4183. 10.1109/ACCESS.2018.2885640

Allemang, D., & Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in RDFS and OWL* (2nd ed.). Morgan Kaufmann, Elsevier.

Bach, K. (2017). Knowledge Engineering for Distributed Case-Based Reasoning Systems. In G. J. Nalepa & J. Baumeister (Eds.), *Synergies Between Knowledge Engineering and Software Engineering. Advances in Intelligent Systems and Computing (AISC), Volume (626)* (pp. 129–147). Springer., doi:10.1007/978-3-319-64161-4_7

Bunakov, V. (2015). Use Cases for Triple Stores and Graph Databases in Scalable Data Infrastructures. In DAMDID/RCDL (pp. 37–40). Academic Press.

Buranarach, M., Supnithi, T., Thein, Y. M., Ruangrajitpakorn, T., Rattanasawad, T., Wongpatikaseree, K., Lim, A. O., Tan, Y., & Assawamakin, A. (2016). OAM: An ontology application management framework for simplifying ontology-based semantic web application development. *International Journal of Software Engineering and Knowledge Engineering*, *26*(1), 115–145. doi:10.1142/S0218194016500066

Davies, J., Studer, R., & Warren, P. (2006). *Semantic Web technologies: trends and research in ontology-based systems*. John Wiley & Sons. doi:10.1002/047003033X

Grimm, S. (2010). Knowledge Representation and Ontologies. In M. M. Gaber (Ed.), *Scientific Data Mining and Knowledge Discovery: Principles and Foundations* (pp. 111–137). Springer-Verlag.

Harris, S., Seaborne, A., & Prud'hommeaux, E. (2013). *SPARQL 1.1 query language. W3C Recommendation, 21 March 2013*. Retrieved from https://www.w3.org/TR/sparql11-query/

Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology (Vol. 1). Morgan & Claypool Publishers. doi:10.2200/S00334ED1V01Y201102WBE001

Hebeler, J., Fisher, M., Blace, R., & Perez-Lopez, A. (2011). *Semantic web programming*. John Wiley & Sons.

Hoekstra, R. (2009). Ontology Representation: Design Patterns and Ontologies that Make Sense. In *Proceedings of the 2009 conference on Ontology Representation: Design Patterns and Ontologies that Make Sense*. IOS Press.

Hogan, A. (2020). Resource Description Framework. In *The Web of Data* (pp. 59–109). Springer. doi:10.1007/978-3-030-51580-5_3

Jambhulkar, S. V., & Karale, S. J. (2016). Semantic web application generation using protégé tool. In *Online International Conference on Green Engineering and Technologies (IC-GET)*. Coimbatore, India: IEEE. doi:10.1109/GET.2016.7916686

Krieger, H.-U., & Declerck, T. (2015). An OWL Ontology for Biographical Knowledge. Representing Time-Dependent Factual Knowledge. In BD (pp. 101–110). Academic Press.

Krieger, H.-U., & Willms, C. (2015). Extending OWL ontologies by Cartesian types to represent N-ary relations in natural language. *Language and Ontologies*, 1.

Kumar, G. S., & Zayaraz, G. (2015). Concept relation extraction using Naı ¨ ve Bayes classifier for ontology-based question answering systems. *Journal of King Saud University - Computer and Information Sciences, 27*(1), 13–24. 10.1016/j.jksuci.2014.03.001

Levišauskait, K. (2010). Investment Analysis and Portfolio Management. Leonardo da Vinci programme project. Kaunas, Lithuania: Leonardo da Vinci Programme Project, Vytautas Magnus University.

Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., & Chen, Y. (2014). The effect of news and public mood on stock movements. *Information Sciences*, *278*, 826–840. doi:10.1016/j.ins.2014.03.096

Noy, N., Rector, A., Hayes, P., & Welty, C. (2006). Defining n-ary relations on the semantic web. *W3C Working Group Note, 12*(4).

Polleres, A., Hogan, A., Delbru, R., & Umbrich, J. (2013). RDFS and OWL reasoning for linked data. In Reasoning Web. Semantic Technologies for Intelligent Data Access (pp. 91–149). Springer. doi:10.1007/978-3-642-39784-4_2

Saat, N. I. Y., & Noah, S. A. M. (2016). Rule-based Approach for Automatic Ontology Population of Agriculture Domain. *Information Technology Journal*, *15*(2), 46–51. doi:10.3923/itj.2016.46.51

Segaran, T., Evans, C., & Taylor, J. (2009). *Programming the semantic web* (1st ed.). O'Reilly Media, Inc.

Simeonov, B., Alexiev, V., Liparas, D., Puigbo, M., Vrochidis, S., Jamin, E., & Kompatsiaris, I. (2016). Semantic integration of web data for international investment decision support. In *International Conference on Internet Science* (pp. 205–217). Springer. doi:10.1007/978-3-319-45982-0_18

Vinu, P. V., Sherimon, P. C., Krishnan, R., & Saad Takroni, Y. (2014). Pattern Representation Model for N-ary Relations in Ontology. *Journal of Theoretical and Applied Information Technology*, *60*(2).

Wang, X. H., Zhang, D. Q., Gu, T., & Pung, H. K. (2004). Ontology based context modeling and reasoning using OWL. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, 14-17 March 2004* (pp. 18–22). Orlando, FL: IEEE. doi:10.1109/PERCOMW.2004.1276898

Wanner, L., Rospocher, M., Vrochidis, S., Johansson, L., Bouayad-Agha, N., Casamayor, G., & Moumtzidou, A. et al. (2015). Ontology-centered environmental information delivery for personalized decision support. *Expert Systems with Applications*, *42*(12), 5032–5046. doi:10.1016/j.eswa.2015.02.048

Abduladem Aljamel received his Ph.D. degree in knowledge-based information extraction and exploration from Nottingham Trent University. He is currently a Lecturer at the school of Information Technology in Misurata University. Prior to joining Misurata University, he was a lecturer and trainer at the Libyan Higher Institution of Science and Technology. His research interests include information extraction, and knowledge representation and exploration. He is a member in the Computational Linguistics research group. The goal of this group includes the formulation of the grammatical and semantic frameworks for characterising Arabic Language in ways enabling computationally tractable implementations of syntactic and semantic analysis.

Taha Osman (PhD) started exploring the utilisation of the semantic web technology in the intelligent composition of Web services. The application of semantic technologies in Dr Osman's research further expanded to intelligent information retrieval and knowledge management, which culminated in the collaboration with the Press Association – the UK's prime multimedia content and news provider, where Dr Osman's research team helped the company to develop a semantic-based image indexing and retrieval system that improves the accuracy and recall of PA Photos images search engine.

Dhaval Thakker (PhD) is a Senior Lecturer in Computer Science at the University of Bradford. Prior to joining Bradford, he worked as a Research Fellow at the University of Leeds from 2011 to 2015 and was leading semantic web related research in several EU projects. Before Leeds, he worked in the industry with UK's national news agency (Press Association) as a Research & Development Consultant to provide strategic and technical leadership in implementing Semantic Web and Linked data related projects to improve access to their media repositories. At Bradford, he leads the work on the Internet of Things (IoT), Smart Cities and Semantic Web research stream with currently funded projects from European Commission (Smart Cities and Open Data REuse (SCORE)), two Innovate UK projects, and one GCRF project. Dhaval supervises 8 Ph.D. students in these research areas and has an extensive publication track record with 80+ publications.