NOTTINGHAM TRENT UNIVERSITY

A KNOWLEDGE-BASED FRAMEWORK FOR INFORMATION EXTRACTION AND EXPLORATION

ABDULADEM ALJAMEL

School of Science and Technology

A thesis submitted in partial fulfilment of the
requirement of Nottingham Trent University
for the degree of Doctor of Philosophy

(January 2018)

I hereby declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except for included work that has formed part of jointly-authored publications. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

Part of the work in this thesis has been previously published as a full conference research paper. The details are below:

Aljamel, A., Osman, T. and Acampora, G., 2015. Domain-specific Relation Extraction: using distant supervision Machine Learning. In: The 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Lisbon, Portugal, November 12-14, 2015. pp. 92-103. SCITEPRESS.

Acknowledgements

First and foremost, I want to thank all of you who made this journey more bearable and even enjoyable sometimes. I never meant to do a PhD, but life is unpredictable and takes you to unexpected places you have never imagined.

I am extremely grateful to my main supervisor and director of study, Dr Taha Osman, for his valuable guidance, scholarly inputs and consistent encouragement I have received throughout the research work. This feat was possible only because of the unconditional support provided by him. He has always made himself available to clarify my doubts despite his busy schedules and I consider it as a great opportunity to do my doctoral programme under his supervision and to learn from his research expertise and understanding. Thank you Taha for all your help and support.

I would also like to extend thanks to my supervisors Dr Giovanni Acampora and Dr Ziqi Zhang and my independent assessor Dr Richard Cant for their advice regarding this research and Dr Autilia Vitiello for her critical review of our journal paper. Also, I would like to express my gratitude to Dr Evtim Peytchev for his generosity of allowing me use his Mac Station for my research experiments.

I believe that intensive courses like PhD are impossible to succeed without the patience and support of a caring family. I owe my loving thanks to my wife, Noha. Without her support, encouragement and understanding, it would have been impossible for me to finish this work.

I owe a lot of gratitude to my parents, who encouraged me at every stage of my life, and longed to see this achievement come true. I deeply miss them because they are not with me to share this joy. May God have mercy on their souls. Also, my special gratitude is due to my brothers, sisters and children for their support.

A heartfelt thanks to my fellow doctoral students for their feedback, cooperation and of course friendship; in particular, my friend Hussein Khalil for being part of this long journey. My friend Remi (Ayodeji Remi-Omosowon) deserves special mention for his ideas and informal discussions.

The thesis would not have come to a successful completion without the support I received from NTU staff; specially the staff of Doctoral School, DoctoratePlus Programme, library and in-sessional Academic English Support courses. I take this time to express my gratitude to all of them for their services.

After all, it has been a unique experience and without the people around me, it would not be the same.

List of Acronyms

ANNIE	A Nearly-New Information Extraction system
3N	3 Notation
API	Application Programming Interface
BBB	Bundle Branch Block
BNode	Blank Node
BORO	Business Objects Reference Ontology
CSV	Comma-Separated Values
DDM	Dividend Discount Model
DL	Description Logics
DSS	Decision Support System
EA	Evolutionary Algorithms
ECG	ElectroCardioGram
FIRST	large scale inFormation extraction and Integration infRastructure for SupporTing financial decisionmaking
FOAF	Friend Of A Friend
FOL	First-Order Logic
GA	Genetic Algorithm
GATE	General Architecture Text Engineering platform
GDP	Gross Domestic Product
GUI	Graphic User Interface
HC	Hill-Climbing algorithm
IDE	Integrated Development Environment
IE	Information Extraction
IR	Information Retrieval
JAPE	Java Annotation Patterns Engine
JSON	JavaScript Object Notation
KB	Knowledge Base (or Knowledgebase)
kbfwo	Knowledge-based Framework Ontology
KNN	K Nearest Neighbour algorithm
LHS	Left Hand Side
LOD	Linked Open Data
MKNN	Mutual K-Nearest Neighbour
ML	Machine Learning
N3	Notation3
NASDAQ	National Association of Securities Dealers Automated Quotations
NE	Named Entity
NER	Named Entity Recognition
NLP	Natural Language Processing
NYSE	New York Stock Exchange
OWA	Open World Assumption
OWL	Web Ontology Language
PAUM	Perceptron Algorithm with Uneven Margins algorithm
POS	Part Of Speech
REST	Representational State Transfer
RDF	Resource Description Framework

RDFS	Resource Description Framework Schema
RHS	Right Hand Side
RMHC	Random Mutation Hill-Climbing algorithm
ROC	Receiving Operator Characteristic
RQ	Research Question
RWS	Roulette Wheel Selection
S&N	Sufficient and Necessary Logic
SA	Simulated Annealing algorithm
SBI	State Bank of India
SDB	SQL triple DataBase
SPARQL	SPARQL Protocol And RDF Query Language
SQL	Structured Query Language
SUS	Stochastic Universal Sampling
SVM	Support Vector Machines algorithm
SW	Semantic Web
SWRL	Semantic Web Rule Language
SWT	Semantic Web Technology
TDB	Triple DataBase
TOVE	TOronto Virtual Enterprise ontologies
TSV	Tab Separated Values
UI	User Interface
URI	Uniform Resource Identifier
WB	World Bank
W3C	World Wide Web Consortium
XBRL	eXtensible Business Reporting Language
XML	eXtensible Mark-up Language
XSD	XML Schema Definition

Abstract

Harnessing insights from the colossal amount of online information requires the computerised processing of unstructured text in order to satisfy the information need of particular applications such as recommender systems and sentiment analysis. The increasing availability of online documents that describe domain-specific information provides an opportunity in employing a knowledge-based approach in extracting information from Web data.

In this thesis, a novel comprehensive knowledge-based framework is proposed to construct and exploit a domain-specific semantic knowledgebase. The proposed framework introduces a methodology for linking several components of different techniques and tools. It focuses on providing reusable and configurable data and application templates, which allow developers to apply it in diversity of domains. The objectives of this framework are: extracting information from unstructured data, constructing a semantic knowledgebase from the extracted information, enriching the resultant semantic knowledgebase by sourcing appropriate semi-structured and structured datasets, and consuming the resultant semantic knowledgebase to facilitate the intelligent exploration and search of information. For the purpose of investigating the challenges of extracting and modelling information in a specific domain, the financial domain was employed as a use-case in the context of a stock investment motivating scenario.

The developed knowledge-based approach exploits the semantic and syntactic characteristics of the problem domain knowledge in implementing a hybrid approach of Rule-based and Machine Learning based relation classification. The rule-based approach is adopted in the Natural Language Processing tasks associated with linguistic and structural features, Named Entity Recognition, instances labelling and feature generation processes. The results of these tasks are used to classify the relations between the named entities by employing the Machine Learning based relation classification. In addition, the domain knowledge is analysed to benefit knowledge modelling by translating the domain key concepts into a formal ontology. This ontology is employed in constructing semantic knowledgebase from unstructured online data of a specific domain, enriching the resulting semantic knowledgebase by sourcing semi-structured and structured online data sources and applying advanced classifications and inference technologies to infer new and interesting facts to improve the decision-making and intelligent exploration activities. However, most relations are non-binary in the problem domain knowledge because of its specific characteristic hence an appropriate N-ary relation patterns technique were adopted and investigated.

A serious of a novel experiments were conducted to implement and configure a Machine Learning based relation classification. The experimental evaluation evidenced that the developed knowledge-assisted ML relation classification model, which was further boosted by our implementation of GAs to reduce the feature space, has resulted in significant improvement in the process of relation extraction. The experimental results also indicate that amongst the implemented ML algorithms, SVM exhibited the best relation classification accuracy in the majority of the training datasets, while retaining acceptable levels of accuracy in the rest in the remaining training datasets.

Web Ontology Language (OWL) reasoning and rule-based reasoning on the resultant semantic knowledgebase were applied to derive stock investment specific recommendations. In addition, SPARQL query language was employed to explore the semantic knowledgebase. Moreover, taking into consideration the problem domain's requirements for modelling non-binary relations, a relation-as-class N-ary relations pattern was implemented, and the reasoning axioms and query language were adjusted to fit the intermediate resources in the N-ary relations requirements.

In this thesis also the experience on addressing the challenges of implementing the proposed knowledge-based framework for constructing and exploiting a semantic knowledgebase were summarised. These challenges can be considered by domain experts and knowledge engineers as a novel methodology for employing the Semantic Web Technologies for the knowledge user to intelligently exploit knowledge in similar problem domains.

The evaluation of knowledge accessibility by utilising Semantic Web Technologies 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. Investigating the tasks of reasoning, accessing and querying the semantic knowledgebase evidences that Semantic Web Technologies can perform an accurate and complex knowledge representation to share Knowledge from a diversity of data sources and, improve the decision-making process and the intelligent exploration of the semantic knowledgebase.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

An increasing amount of data is being made available online. It covers a diversity of specific domains such as entertainment, financial and economy, education, politics, sports and others. I believe that there is an opportunity in extracting specific information from this data to be exploited to inform a variety of applications and services, such as recommender systems to advise financial investors about a potential business risk, sentiment analysis to inform the music industry about an emerging consumer trend or advanced data exploration engines. However, this online data is diverse in terms of volume and complexity, 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 techniques are needed in order to understand the data and extract useful information for the end users. Moreover, automatic Information Extraction can constitute a core component technology in many other Natural Language Processing applications, such as machine translation, question and answering, text summarisation, opinion mining and decision support systems.

I believe that Information Extraction efforts can benefit from the knowledge of problem domain characteristics. Analysing and understanding the domain knowledge to capture its characteristics and features can inform linguistic-based and Machine Learning based Information Extraction techniques to be more accurate in extracting useful knowledge from unstructured data sources.

The extracted information of targeted domain knowledge can be structured and represented in machine understandable semantic model by utilising Semantic-based approaches such as Semantic Web based technologies. The Semantic Web is defined as an extension to the Web where information can be understandable by machines and humans. Semantic Web Technologies present powerful tools to access, use and share information. This presents an opportunity to enrich the semantically structured data by using structured data in public datasets that adopt the same Semantic Web standards. The resultant structured data can be reasoned upon to deliver intelligent query methods against the information and the underlying metadata.

1.2 Domain Knowledge Role in Information Extraction

Applying Information Extraction techniques on unstructured data aims to represent the extracted information in a structured view. Information Extraction could be considered as a

pipeline process. In each stage of that pipeline, the tasks of Natural Language Processing are applied in order to obtain valuable information from natural language texts. An Information Extraction process usually starts in recognising the named entities; then, identifying identity relations between named entities, which is known as co-references resolution. Lastly, extracting the relation between the named entities in a certain event (Cunningham 2005, Farmakiotou, et al. 2000).

In Named Entity Recognition task, the sentence's atomic elements (words or entities) are addressed and classified into predefined types of named entities, such as organisations, place names, persons, dates and numbers. By applying Named Entity Recognition, additional descriptive information can be extracted from the text about the detected entities such as the title and gender of persons. In the entities' co-reference resolution task, the expressions in a document that refer to the same entity are identified. The co-reference relation will be marked between elements. In the Relation Extraction task, the relationships between the named entities are identified. If more than one relation are extracted and linked together, these relations comprise an event. Event extraction refers to the task of identifying events in unstructured data, which involves extracting several entities and their interrelations; for example, the location and date of opening a branch of a company (Piskorski and Yangarber 2013, Karkaletsis, et al. 2011). Usually, these entities and their interrelations are identified in accordance to the characteristics of the problem domain knowledge. I believe that domain knowledge is important for guiding Information Extraction from unstructured data.

What is domain knowledge?

Domain Knowledge is a knowledge about a specific field/domain of interest or subject that are understood by practitioners in that field/domain of expertise. In fact, domain-specific knowledge is required to identify specific Information Extraction tasks. It requires to be analysed to understand its 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 be engineered in linguistic or structural features. These features should use the relevant knowledge of the problem domain to reflect its characteristics. Once the analysis of the knowledge of the problem domain is performed to understand its characteristics, it can be employed in the implementation of Information Extraction systems by using multiple Information Extraction approaches such as rule-based or Machine Learning based (Gao and Zhang 2003, Song and Roth 2017).

Identifying the relevant domain knowledge is about capturing the key and relevant domain concepts and the relationships between these domain concepts. This includes the common-sense knowledge, which refers to general knowledge of the problem domain. The role of relevant and common-sense knowledge of the domains are important in relating

different concepts, arguments, facts, and events to create an inference model and employing semantic relations between concepts to decode information hidden in texts. Furthermore, this relevant knowledge is useful in classifying the entities and their interrelations according to the engineered linguistic and structural features and the captured key concepts of the problem domain. It is also effective for addressing a clear definition of decision-making problems to be modelled and supported by using rule-based reasoning techniques. The captured key concepts, vocabularies and interrelations in the targeted domain knowledge should cover all the data involved in the Information Extraction tasks (Gao and Zhang 2003, Song and Roth 2017, Chen, Yin and Pang 2017).

Knowledge-based approaches originate from understanding the syntactic and semantic characteristics of the domain knowledge to be formalised into semantic model or metadata. Then, discovering and mapping the entities in data onto elements of metadata. For example, the concepts “Organization” and “Person” can be separated from the entities “Apple”, “Tim Cook”, “Satya Nadella” and “Microsoft”. The domain knowledge can be modelled by using the key concepts “Organization” and “Person” including the relation between them. In addition, the entities and their interrelation can be classified according to domain knowledge model and the related features engineering. As a result, “Tim Cook” and “Satya Nadella” will be classified to belong to “Person” concept and “Apple” and ”Microsoft” will be classified to belong to “Organization” concept. In fact, domain-specific knowledge can be modelled and clearly separated from other components of the Information Extraction tasks to be adapted to other domains (Gao and Zhang 2003, Song and Roth 2017).

There are several external domain knowledge resources that comprise lexicons and ground facts. Building lexicons includes collecting the most relevant domain terms and entities and their synonyms. They are large dictionaries of person names, location names, temporal expressions and others. These are often referred to as gazetteer lists. These gazetteers are used for different tasks such as recognising the Named Entities. The ground facts represent structured information of the relation between entities which can be used as reference knowledge for other tasks such as slot filling and relation extraction. The opportunity of adopting knowledge-based approach for Information Extraction and Knowledge Representation derived from the common sources of these external domain knowledge resources, which are gazetteers and ground facts. These resources adopt the same Semantic Web standards and available in the Linked Open Datasets (LOD) such as DBPedia and Freebase[1] (Mendes, Jakob and Bizer 2012, Weikum and Theobald 2010, DBpedia Team 2015, Freebase Metaweb 2014, LOD 2018).

Understanding domain knowledge in knowledge-based approaches aides Information Extraction process and knowledge representation activities such as reasoning about objects

[1] https://developers.google.com/freebase/. This page provides access to the last available data dump. I downloaded its last data dump and mainly used Freebase data as a distant source for constructing ML training datasets. The original Freebase API was completely shut-down on 31 August 2016.

related to that domain. The next section presents one of the knowledge representation approaches that is based on Semantic Web Technologies.

1.3 Utilising Semantic Web Technologies in Knowledge Representation and Exploration

Knowledge representation approaches aim to represent knowledge that can be reasoned upon and interpreted and explored by machines. Semantic knowledge-based applications should have a computational model for the targeted domain of interest, to represent part of the real world such as physical objects, relationships or events to be understandable and processable by machines. Semantic knowledge bases store the symbols of the computational model of the domain in statement form for machines to perform reasoning. The reasoning procedures aim to derive implicit statements from a given knowledgebase or check the consistency of a particular entailed statement in the knowledgebase (Grimm, Hitzler and Abecker 2007). Knowledge representation approaches represent knowledge in machine understandable semantic model by utilising approaches of semantic-based technologies such as Semantic Web Technologies.

Semantic Web Technologies are based on different forms of knowledge representation, which are semantic networks, rules and logic. A semantic network is a graph whose nodes represent concepts and whose arcs represent relations between these concepts. They provide a structural representation of statements about a domain of interest. Rules reflect the notion of consequence. They allow the expression of various kinds of complex statements. Logic, on the other hand, has been used by semantic networks and rules to be precisely semantically formalised. Without such precise formalisation, they will be uncertain for computational inferencing purposes (Grimm, Hitzler and Abecker 2007).

First-order predicate logic is an important knowledge representation formalism because all current symbolic knowledge representation formalisms can be understood in their relation to first-order logic. It also provides a notion of universal truth, in the sense that a logical statement can be universally valid truth regardless of any preconditions. Description Logics (DL), on the other hand, is a field of research that has studied a particular decidable fragment of First Order Logic. It is expressive enough such that they have become a major knowledge representation paradigm, in particular; for use within Semantic Web Technologies (Grimm, Hitzler and Abecker 2007).

The Semantic Web offers a powerful logical and standardised technologies to represent, share and process knowledge such as inference and validation. It is based mainly on graph theory and Description Logics. One of the core components of Semantic Web is ontology.

Ontology is a formal explicit description of the targeted domain knowledge and it plays a key role in Semantic Web knowledge representation. It is recommended that separating data from metadata or ontology because it provides robustness, scalability, and efficiency for the semantic knowledgebase storage system (Davies, Studer and Warren 2006, Grimm, Hitzler and Abecker 2007, Taye 2010, Hebeler, et al. 2011).

The formalisation of semantic knowledge bases by using ontology could include multiple axioms, definitions, rules, facts, statements, and any other primitives. The main components of the ontology are concepts, relations, instances and axioms and can be characterised in 4-tuple formula below (Davies, Studer and Warren 2006, Grimm, Hitzler and Abecker 2007):

$$O = \langle C, R, I, A \rangle$$

Where:

O is the ontology.

C is a set of classes representing concepts that are relevant in the domain of interest. They are mapped to the generic nodes in semantic networks, or to unary predicates in logic, or to concepts as in Description Logics.

R is a set of relations that semantically connect concepts and instances to specify their interrelations. They are mapped to arcs in semantic networks, or to binary predicates in logic, or to roles in Description Logics.

I is a set of instances that represent the named and inferred individuals. They can be linked to each other by relations which are classified by concepts or axioms. They are mapped to individual nodes in semantic networks or to constraints in logic.

A is a set of axioms that are used to apply constraints on the properties of the concepts and individuals. Axioms are expressed in logic.

Knowledge representation by using ontology in Semantic Web Technologies should be performed by employing formalised Semantic Web languages. The main requirements of these languages are: well defined syntax, efficient reasoning support, a formal semantics, sufficient expressive power and convenience of expression (Ameen, Khan and Rani 2014a). The next subsections presents an overview for these languages and their reasoning support.

1.3.1 Semantic Web Technologies Languages

In the Semantic Web context, ontology is modelled by utilising Semantic Web Technologies Languages. They are: Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL). The specifications of these languages are standardised

and recommended by the World Wide Web Consortium (W3C) (W3C 2018). Brief details of these languages are presented below.

1.3.1.1 Resource Description Framework (RDF)

RDF is a graph-based language that allows data within a domain to be linked through named relationships. It is a simple triple structure and a natural method that extends the current data of WEB into a highly structured model to assist machines to describe processed data. In RDF standards, RDF triples are encoded as a set of nodes, subject, predicate and object. These nodes could be resources that are identified by URI references, literals that denote values such as numbers or strings and blank nodes that represent unnamed or anonymous resources that are not assigned URI references. The subject of an RDF triple may be a URI reference or a blank node, the predicate must be a URI reference, and the object may be of all three kinds (URI references, literals, blank nodes). When combined together, RDF triples form a direct, labelled graph. Subjects and objects of RDF triples become nodes in an RDF graph, and predicates become arcs connecting them. URI is stand for Uniform Resource Identifier. By URIs, resources are uniquely identified throughout the web, which allows for a decentralised organisation of knowledge about commonly referenced resources. The resource URI names can be partitioned into URI namespaces and entity names. The semantic statement can be modelled by using RDF triple based on an ontology (Lord 2010, Taye 2010, Cao, et al. 2012, Henson 2013, Grimm, Hitzler and Abecker 2007).

If several triples are linked to each other, they form an RDF graph. The graph's nodes are URI resources and the arcs are properties. From a knowledge representation view, an RDF graph can be seen as a semantic network. The triples in an RDF graph can originate from different dataset sources with the idea that anybody can state anything about any resource. In this sense, RDF is designed to capture knowledge and meta data that is spread over the web (Lord 2010, Taye 2010, Cao, et al. 2012, Henson 2013, Grimm, Hitzler and Abecker 2007). However, RDF graphs can use blank nodes, which are not assigned URI references. Next subsection will introduce the blank nodes in details.

Blank Nodes

As aforementioned, blank nodes represent unnamed resources in RDF triples that are not assigned URI references. Also, the subject nodes in the RDF triples can be a URI resource or blank node and the object node can be a URI resource, literal or blank node. In the Semantic Web contexts, blank nodes are also known as anonymous resources or bnode. According to RDF standards, blank nodes are utilised to express the existence of a particular thing without using an URI to identify it (Tzitzikas, Lantzaki and Zeginis 2012, Chen, et al. 2012).

In fact, blank nodes are one of the core aspect of Semantic Web technology. They are included in several W3C standards and tools. Also, they are heavily used in several Linked datasets across the Web. However, blank nodes are not always utilised with the same meaning or propose. For example, some publishers use them to describe multi-component structures or represent complex attributes without having to name explicitly the auxiliary node or offer protection of the inner sensitive information of the customers from the browsers (Lantzaki, et al. 2014, Hogan, et al. 2014).

Although blank nodes bring some facilities for expressing the information of the resources and the relationships between them in a semantic knowledgebase, they also bring some troubles to manage and process that knowledgebase. There have been several studies in literature reported the negative impact of blank nodes on the representation of Semantic Web data. They have examined the relationship between the usefulness of datasets and the quantity of blank nodes in them. They have found that the datasets are more useful when the quantity of blank nodes in them is at a minimum. The authors Heath, et al. in (Heath and Bizer 2011), Chen, et al. in (Chen, et al. 2012), Tzitzikas, et al. in (Tzitzikas, Lantzaki and Zeginis 2012), Mallea, et al. in (Mallea, et al. 2011), Hogan, et al. in (Hogan, et al. 2014), Lantzaki, et al. in (Lantzaki, et al. 2014) and Booth in (Booth 2013) have revealed that the blank nodes bring some issues to manage and process knowledge bases. These problems can be summarised into the following four aspects:

1- The problem of merging RDF graphs. Because the scope of blank nodes is limited to the dataset, it is impossible to create external links to their triples. This will reduce the potential interlinking between different Linked Data sources. Consequently, merging data from different datasets becomes very difficult as there is no URI reference to identify the blank node.
2- The problem of RDF graph serialisation. There is no guarantee that the syntactic elements or labels which are generated to identify blank node identifiers in one RDF graph representation format, that the same label will be generated for a given blank node each time the graph is serialised. There is no standard, reliable way to reference a blank node across graph serialisations.
3- The problem of SPARQL queries. In RDF semantics, blank nodes are considered as the variables. On the other hand, blank nodes are considered as constant symbols in SPARQL semantics. This causes an inconsistency between the RDF and SPARQL semantics when using blank nodes. Applying a SPARQL query on an RDF data with blank nodes will produce infinitely many redundant solutions for many patterns. As a result, deleting the redundant blank nodes in the original graph and restricting some entailments evaluation about blank nodes will reduce the inconsistency of SPARQL query semantics.

4- The problem of publishing Linked Data. The core principle of Semantic Web is that (http://) type URI references should be used for RDF resources. It can be used by Semantic Web agents to link between a diversity of datasets finding more information. The datasets linkages will be broken with the external data at the points of the blank nodes. This will bring many troubles in data query and mining. Eliminatuing the blank nodes in the RDF graph is an important pre-process before it can be published on the Web.

According to Hogan, et al. in (Hogan, et al. 2014), there are huge number of standards and tools and a large volume of published data available and utilise blank nodes. This will make any change to the core semantics of blank nodes incur a huge cost at this stage. Even if the core semantics could be conveniently changed, it is not clear what a better alternative would be. In fact, the RDF 1.1 Working Group (W3C 2018) has decided not to change the core semantics of blank nodes; Instead, they discourage the use of blank nodes and all resources should be named using URI references; for example, the implementers of Friend Of A Friend (FOAF) vocabulary specifications has dropped blank nodes in favour of URI references.

1.3.1.2 Resource Description Framework Schema (RDFS)

RDFS is a general-purpose language for representing simple RDF vocabularies on the Web. It facilitates the specification of application-specific ontological vocabularies in form of class and property hierarchies on top of RDF resources. RDFS Language can be used to express the class membership and subsumption between classes. For this purpose, it defines a set of reserved keywords that can be used in RDF triples to relate resources to classes. RDFS defines a system of typing for RDF resources by introducing the concept of a class. The reserved predicate (rdf:type) is used to indicate class membership or defining a resource to be of a certain type. RDFS classes are organised in a hierarchy of types for RDF resources. The reserved predicate (rdfs:subClassOf) is used to state that there is a subclass relationship between two types of classes. In RDF(S) semantics, any resource used in the predicate position of an RDF triple is a member of the class (rdfs:Property). In addition, properties can be organised in a hierarchy by means of the keyword (rdfs:subPropertyOf) (Lord 2010, Taye 2010, Cao, et al. 2012, Henson 2013, Grimm, Hitzler and Abecker 2007). The RDF(S) vocabularies for typing the resources allow the formulation of subsumption hierarchies and the distinction between instances and concepts in the ontological sense. However, in RDF(S) there is no clear separation between classes and their members. Instead, RDF(S) allows self-reference and classes being members of (meta) classes. Any resource can be tagged as a class by relating it to the predefined meta type (rdfs:Class). The domain and range of the properties can be defined with the predefined predicates (rdfs:domain) and (rdfs:range) (Lord 2010, Taye 2010, Cao, et al. 2012, Henson 2013, Grimm, Hitzler and Abecker 2007).

1.3.1.3 Web Ontology Language (OWL)

The OWL provides an expressive language for defining ontologies that capture the semantics of domain knowledge. It is built on top of RDFS and adds a logical formalism to the language. W3C standardisation efforts have produced the OWL family of languages for describing ontologies in the Semantic Web, which comes in different expressiveness capability and each emphasising on different language features: OWL-Full, OWL-DL, and OWL-Lite. OWL-Full is the most expressive language; meanwhile, emphasises on compatibility with RDFS. However, it introduces problems of computational tractability and un-decidability. OWL DL is a subset of OWL-Full and based on Description Logics (DL). To maintain its decidability, it is compatible only with a specific subset of RDFS language. OWL-Lite is subset of OWL-DL and it offers a limited feature set even though it is adequate for many applications. In fact, it is relatively efficient computationally. OWL-DL is currently the most prominent Semantic Web ontology language following the Description Logics paradigm and has desirable computational properties for reasoning systems (Hoekstra 2009, Grimm, Hitzler and Abecker 2007).

OWL facilitates the machine interpretability of Web contents more than that supported by RDF and RDFS. It is designed to be utilised by applications that require processing the content of information rather than solely presenting it to humans. OWL provides additional vocabulary along with a formal semantics. For example, it allows the expressing of individuals equality (owl:sameAs), the expressing of equivalent or disjoint classed 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, et al. 2013, Roussey, et al. 2011, Tomai and Spanaki 2005).

In addition, OWL can describe complex class by using Boolean operators and restriction constructors. Each Boolean operator takes one or more classes as operands. These classes may be named classes, or may be complex classes formed from other constructors' or operators' descriptions. The examples of these Boolean operators are owl:unionOf, owl:intersectionOf and owl:complementOf.

Restriction constructors allow describing the individuals of restricted classes in terms of constraints on relationships that those individuals participate in, using specific relation properties with individuals in specific classes. Restrictions consist of three parts:

1- Quantifier. They are value restrictions and cardinality restrictions. OWL provides three kinds of value restrictions, they are existential (owl:someValuesFrom), universal (owl:allValuesFrom) or limited existential (owl:hasValue) value restrictions and provides three kinds of cardinality restrictions, they could be maximum

(owl:maxCardinality), minimum (owl:minCardinality), or exact (owl:cardinality) cardinality restrictions.

2- Property. That specifies what property is to be used in the definition of the restriction class. It is defined by the OWL keyword (owl:onProperty).
3- Filler. That specifies the class of individuals which are used to restrict the individuals of the restricted class.

Restrictions can be defined by using the OWL class owl:Restriction. Then, the description is used to define a new anonymous restricted class and it becomes an existing class. This restricted class is a special kind of class that has individual members, which is similar to named class. Each kind of restrictions describes how the restricted class is constrained by the possible asserted values of properties. Membership in a restricted class must satisfy the conditions specified by the kind of restriction and the property specification. In general, Boolean operators and restrictions can be nested or ordered to describe a restricted class (Allemang and Hendler 2011). However, some kinds of restrictions cannot be used to define the restricted class. It is because of the Open World Assumption (OWA). In OWA, universal restrictions cannot be used for identification using equivalence axioms because there is no way to know whether the individual has additional properties of that type or not. In addition, the existential restriction does not constrain the property relationship to members of the restricted class, it just states that every individual must have at least one property relationship with a member of the named class.

The Semantic Web standards syntaxes of writing OWL are RDF/XML and Turtle; however, they are verbose and very hard to read. There is a syntax standard for writing OWL that is used in formal documents and designed to be presented for human reading, it is the Manchester syntax (W3C 2018). It is a text based and more human friendly syntax. For example, Manchester syntax allows strings, integers, decimals, and floats to be written as in most programming languages (Horridge, et al. 2006, Horrocks and Patel-Schneider 2011).

To enhance human readability of the OWL examples in this thesis, Manchester syntax will be used in the remainder of this thesis. However, Description Logics (DL) style or Turtle syntax will be used if they are needed. Table 1.1 below shows the Manchester and Description Logics (DL) syntaxes for OWL Class constructors, Boolean and restriction operators.

Table 1.1: The Manchester and DL Syntaxes for OWL Class Constructors, Boolean and Restriction Operators

OWL Constructor	DL Syntax	Manchester Syntax	Example
intersectionOf	$C \cap D$	C and D	Employee and StockHolder
unionOf	$C \cup D$	C or D	Person or Organization
complementOf	$\neg C$	not C	not StockHolder
oneOf	$\{a\} \cup \{b\}...$	{a, b, ...}	{England, Italy, Spain}

someValuesFrom	∃R.C	R some C	hasStock some Person
allValuesFrom	∀R.C	R only C	hasStock only Organization
minCardinality	≥n R	R min n	EmployerOf min 50
maxCardinality	≤n R	R max n	EmployerOf max 300
cardinality	=n R	R exactly n	EmployerOf exactly 3
hasValue	∃R {a}	R value a	hasProduct value DeskTop

1.3.2 Semantic Web Reasoning

Not only do OWL ontologies allow extensive knowledge expressivity through facts representation, but they also infer new facts from existing facts by reasoning process through the use of their meta-data. A reasoning process can be applied on a semantic knowledgebase to solve problems and make decisions that would otherwise require considerable expertise effort and time; especially, if this knowledge has to be effectively used for reasoning as a part of Decision Support Systems (DSS). Ontologies in these systems should be designed in a way that allows knowledge inference and reasoning (Corsar and Sleeman 2008, Jovic, Prcela and Gamberger 2007, Isiaq and Osman 2014).

Reasoner makes it possible to automatically compute the class hierarchy specifically when constructing very large ontologies. It is very difficult to keep large ontologies in a maintainable and logically correct state without reasoners. It is recommended to construct the class hierarchy as a simple tree when the ontology has classes that have many super-classes. In fact, computing and maintaining multiple inheritance is the reasoner responsibility. Not only does this promote the reuse of the ontology by other ontologies and applications, it also minimises human errors that are inherent in maintaining a multiple inheritance hierarchy (Goncalves, et al. 2015).

There are two categories of reasoning in Semantic Web context, Ontology OWL reasoning and user-defined rule-based reasoning. Below is an explanation for those two categories.

1.3.2.1 OWL reasoning:

In OWL reasoning, there are many tasks the correspond to standard Description Logics reasoning tasks such as checking the semantic knowledge consistency with respect to the ontology or determine whether individuals in knowledgebase do not violate descriptions and axioms described by ontology. However, there are more tasks can be achieved by reasoning process such as (Bock, et al. 2008):

1- Checking the satisfiability of a concept by determining whether a description of the concept is not contradictory or whether an individual can exist that would be an instance of the concept.
2- Checking the concepts' subsumption by determining whether concept C subsumes concept D or whether the description of a class C is more general than the description of D.

3- Checking whether the individual is an instance of a concept without violating the descriptions of the concept.
4- Individuals classification by retrieving a property fillers according to some constraints on relationships between individuals' classes.

1.3.2.2 Rule-based reasoning:

Since OWL axioms and class expressions are variable-free and modelling constructs of OWL are not always adequate, there are statements that may not be expressed simply in OWL and OWL may not suffice for all applications. Thus, rules can be an alternative paradigm for a reasoning process (Hitzler, Krotzsch and Rudolph 2009).

Rules in the Semantic Web are typically conditional statements, if-then clauses. By using these clauses, a new knowledge is added only if a particular set of statements is true. These clauses contain logical functions and operations that can be expressed in rule languages or formats. Not only do rule languages allow describing relations that cannot be described using OWL language, but also they allow sharing and reusing existing rules on the Web. Rule-based reasoning can benefit from the rule language, which permits data interoperation between different reasoners. Requirements of rule language for the Semantic Web include expressiveness, rule interchange, rule integration, rule language interoperability, and compatibility with other Semantic Web standards (Buranarach, et al. 2016).

Rule-based reasoners apply rules with data to reason and derive new facts. When the data match the rules' conditions, the reasoners can modify the knowledgebase; for example, for fact assertion or retraction, or to execute functions. It is good practice to construct rules from concepts included in the ontology. In this way ontology design is the first and necessary step in the actionable knowledge construction process. The rules can be used for reasoning as a part of a Decision Support Systems (DSS) or they may be used together with the concepts presented in the ontology (Wang, et al. 2004, Jovic, Prcela and Gamberger 2007, Hebeler, et al. 2011, Rattanasawad, et al. 2014).

There are three reasoning strategies or algorithms applied by rule-based reasoner to perform reasoning tasks, Forward chaining, Backward chaining, and a hybrid execution model (Hebeler, et al. 2011, Al-Ajlan 2015, Buranarach, et al. 2016):

1- Forward chaining is a bottom-up computational model. It starts with a set of known facts and applies rules to generate new facts whose premises match the known facts. The inference moves forward from the facts toward the goal.
2- Backward-chaining is a top-down computational model. It starts with a goal and looks for rules to support this goal. The inference moves backward from the intended goal to determine facts that would satisfy that goal.
3- A hybrid execution rule reasoning process performs reasoning by combining both forward and backward chaining. Rules present the form that can be effectively used in order to present actionable knowledge.

User-defined rule-based reasoning is a flexible reasoning mechanism through the creation of user-defined reasoning rules within the entailment of First Order logic. There are several Rule languages designed for the Semantic Web. Some of them are introduced by W3C such as Semantic Web Rule Language (SWRL) and others are introduced by different inference engines such as the Jena rule format. Each rule language usually differently supports various logic concepts, and functions (Buranarach, et al. 2016).

SWRL was introduced by W3C (W3C 2018). It is based on combination of the sublanguages of OWL, OWL-DL and OWL-Lite, with Unary/Binary Datalog RuleML, the sublanguage of RuleML. SWRL extends the set of OWL axioms to enable rules to be combined with an OWL knowledge base. The syntax of the rule language is relatively like RuleML. They can also interoperate with each other. Logical operators and quantifications supports of SWRL are the same as RuleML's. In addition, RuleML contents can be parts of SWRL content. Axioms may consist of RDF, OWL and rule axioms. A relation can be an URI, a data range, an OWL property or a built-in relation. An object can be a variable, an individual, a literal value or a blank node. Additionally, the rule language provides many sets of built-in functions such as string functions and mathematical functions (Hebeler, et al. 2011, Buranarach, et al. 2016).

The Jena rule format is used only by reasoning engines in the Jena framework (JENA Apache 2015). The rule language syntax is based on RDF(S) and uses the triple representation of RDF descriptions, which is almost like Notation3 (N3) (W3C 2018) except that a rule name can be specified in a rule, no formula notation, and built-in functions are written in function terms. The built-in functions consists of many set of functions including production functions such as instance creation and instance removing, and can also be extended by the user.

The SWRL rules are part of OWL ontology. The OWL ontology with the SWRL rules are bound to the rules reasoner engine together to execute the rules. On the other hand, the Jena rules and OWL ontology are bound to rules reasoner engine separately, then, the rules are executed.

1.3.3 Knowledge Exploration

Knowledge exploration is about how typical or regular end users can access semantic knowledge and how user interfaces hide the complexity of query languages for those end users; meanwhile, those users take advantage of using exploration and visualisation techniques. There are efforts have been made to facilitate user interaction with the semantic knowledge to assist users to learn and make sense of complex and heterogeneous data and to allow them benefit from the expressivity of Semantic Web standards and languages. Semantic Web Technologies could be utilised in exploring semantic knowledgebase by applying different methods such as keyword search, faceted browsing and auto-translation natural language queries to standard query languages. These methods should support end-

users in situations where the knowledge has complex elements that require constant user interpretation during the exploration process. For example, how to support the end users' search task when they are not familiar with the search domain or they do not have sufficient knowledge about domain to make a query. These kinds of tasks are required in the exploration of the semantic knowledgebase where end users need to identify concepts and relations from the semantic model to learn about the domain in order to understand and acquire knowledge. In addition, these methods should fulfil the tasks of exploring the semantic knowledge by end users and visualise the result of the exploration task in a human understandable format (Thakker, Yang-Turner and Despotakis 2016, Fafalios and Tzitzikas 2013).

1.4 Problem Statement

As the online documents are largely unstructured and constructed in natural human languages, they require applying of automatic techniques to extract useful information from them. This data can have more value when it is formalised in a machine understandable format. In addition, it is challenging to align the discrepancies in knowledge presentation by various contributing information sources and deliver intelligent query methods against that information and its semantic model.

However, extracting information from unstructured data and transferring it into a structured format to be processed by machines for different use-case scenarios is an important problem because it requires addressing and overcoming different challenges. These challenges cover a diversity of disciplines, approaches, tools and techniques, which include automatically extracting information from unstructured data, semantically representing domain knowledge, constructing semantic knowledge from different data sources and consuming the resultant semantic knowledgebase by intelligently exploring it and supporting the decision making process. It is challenging, also, to combine these disciplines, approaches, tools and techniques in one framework to allow the application developers emphasising their efforts on domain problems.

1.5 Proposed Solution

Knowledge-based approach is based on understanding the syntactic and semantic characteristics of domain knowledge. These characteristics can play an important role in improving Information Extraction processing tasks. Moreover, knowledge-based approaches refer to the ability to represent and process knowledge within a domain-specific problem. There is opportunity in employing a knowledge-based approach in extracting information from Web data and process it because there are increasingly online documents that describe information concerning a specific domain; for example, in politics or stock exchange news; actually, the data sources of these documents exclusively service a

particular domain. The domain specificity of these documents offers an excellent opportunity for analysing their domain knowledge to be formally characterised by capturing its attributes, linguistic features, semantic features, functions, dynamics, terminologies, concepts and relations.

I also hypothesise that utilising Semantic Web Technologies for domain knowledge representation can result 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 ontology can also facilitate the inference of new facts from the extracted information to support decision-making and knowledge exploration activities.

The financial domain will be employed as a use-case to investigate extracting information from that domain, modelling the patterns in the extracting information into a semantic model, constructing a semantic knowledgebase and exploiting the knowledgebase to support decision-making process and the intelligent exploration.

To conclude, I will approach my proposed solution of this problem through designing and implementing a framework for developing knowledge-based applications. The framework will adopt a knowledge-based approach to aid the Information Extraction process from the problem domain, its knowledge representation activities and intelligent exploration of the resulting knowledgebase. It will be established by modelling 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. The proposed framework will present a methodology for integrating several components of different techniques and approaches such as Information Extraction, Machine Learning, Evolutionary Algorithms and Knowledge representation.

1.6 Research Questions

As aforementioned, knowledge-based approaches are based on understanding the problem domain knowledge; as a result, this thesis is based on the following hypothesis:

> "Adopting a knowledge-based approach will aid the Information Extraction process from the problem domain, its knowledge representation activities and intelligent exploration of the resulting knowledgebase."

It is worth pointing out that this hypothesis is investigated in the context of developing a framework for the realisation of a domain-specific intelligent exploration. Where stock investment decision-making is selected as a use-case scenario. The investigation, which is

based on the hypothesis above, provides a set of research questions (RQ) in order to fulfil the implementation of the proposed framework. These questions are:

RQ1) Knowledge-based approach is based on analysing domain knowledge to understand its characteristics (the linguistic features and structural features). How can Information Extraction and knowledge representation benefit from this knowledge-based approach?

RQ2) Supervised Machine Learning algorithms is one of the approaches that are applied for Relation classification. The performance of supervised Machine Learning algorithms is affected by the quality of the training datasets, the quality of features vectors and the parameter values. Consequently, how can these elements be configured and optimised in a relation classification problem?

RQ3) As our intention is to exploit knowledge of the problem domain in the Information Extraction process, can the knowledge-based approach contribute to improving Machine Learning based methods for relation classification?

RQ4) A Knowledge-based approach to Information Extraction introduces a multiplicity of features that can be used to train relation classifiers such as linguistic and structural features. Some of these features could be redundant, irrelevant and noisy for robust training datasets representation. How can optimisation techniques be employed and configured to select the best feature's subsets and improve the performance of the relation classification model?

RQ5) The performance of relation classifiers in the proposed solution implies that it is affected by the quality of the feature's vector. Therefore, there is a need to investigate whether there are specific feature's type or category can be more significant in improving the relation classifiers performance. Hence, are there specific feature's type or category can be more significant in improving the relation classifiers performance?

RQ6) Semantic modelling assumes the representation of semantically tagged knowledge in binary relations; however, the characteristics of some domains, such as our use-case domain, imply more complex or non-binary relations representation. Can non-binary relations be semantically and effectively modelled by using standard Semantic Web Technologies within the knowledge-based framework?

RQ7) The semantic knowledgebase will be constructed from a heterogonous domain specific data sources, which are unstructured, semi-structured and structured in nature. Hence, can the formalism in modelling that semantic knowledgebase leverage the domain-

relevant facts aid Information Extraction and improve the intelligent exploration to support decision-making process?

1.7 Research Methodology:

The research methodology adopted in this project is based on the research activities that include a literature Review, requirement analysis and refinement, incremental and iterative development, and evaluation.

1. **Literature Review**

The research involved extensive literature review in the fields of Information Extraction techniques, Machine Learning Algorithms, Evolutionary Algorithms and Knowledge Representation approaches. The literature review was carried out to ensure the originality of the work and to avoid the repetition of existing work done in the field. The literature review of all relevant fields was an iterative process throughout the progress of the PhD research as the related works were taken to be a substantial input parameter in the requirement analysis, tuning and refinement phase, and the requirements analysis. This is quite important due to rapid progress in this area of research.

2. **Requirement Analysis and Refinement**

Similar to many other computer science research problems, identified specifications methodologies and tools considered during the course of the research were thoroughly analysed, examined and refined in order to fulfil their relevance in giving adequate answer(s) to our research motivation and questions.

The advantage of already developed tools and techniques are absorbed for a diversity of tasks in the framework's phases such as Natural Language Processing, Named Entity Recognition, knowledge representation and, sematic knowledgebase access and query.

3. **Incremental and Iterative Development**

The progress of applying the proposed solution is based on Incremental and Iterative development. Incremental development is a stage scheduling strategy in which various phases of the framework are developed incrementally and integrated in the framework as they are completed. Iterative development is a revise scheduling strategy to revise and improve the phases of the framework separately.

The framework is iteratively and incrementally developed to adapt the required tools and techniques to realise and implement the proposed framework.

4. **Evaluation**

In this research, two types of evaluation were applied. The first type is to evaluate the implementation and configuration of Machine Learning algorithms to be optimised in extracting information from unstructured online data. The second type is to evaluate the knowledge representation and knowledge accessibility according to a motivating use-case scenario.

1.8 Thesis Structure

The remaining parts of this thesis are organised as follows:

- Chapter 2 reviews the related works in the literature.
- Chapter 3 introduces the use-case motivating scenario; then, presents the proposed framework including the objectives, phases and tasks.
- Chapter 4 presents the details of the first phase of the framework, which is about domain knowledge analyses, representation and modelling.
- Chapter 5 provides the relation classification pre-processing tasks in the Information Extraction pipeline including the Natural Language Processing and Named Entities Recognition tasks.
- Chapter 6 presents the details of the implementation and evaluation of three different supervised Machine Learning relation classifiers. It includes detecting the relation instance and extracting the feature vectors for composing the training datasets and configuring the relation classifiers.
- Chapter 7 examines the problem of features selection problem by using Genetic Algorithms as a wrapper approach to optimise and reduce the dimensionality of the training datasets. Also, this chapter presents a comparison between GAs and a space search algorithm, Random Mutation Hill-Climbing (RMHC).
- Chapter 8 discusses the constructing of the semantic knowledgebase. Then, it looks at the application of Semantic Web Technologies in two aspects of accessing that knowledge: supporting the decision-making process and semantic knowledgebase exploration.
- Chapter 9 summarises our experience on addressing the challenges of framework implementation to be considered by domain experts and knowledge engineers. Also, this chapter reviews knowledge representation and knowledge accessibility.
- Chapter 10 concludes this research and summarises the main outcomes of this work and outlines suggested further work.

2 Related Works

2.1 Introduction

This chapter reviews studies in the literature that highlights the opportunity in semantically structuring the natural language texts. This opportunity comes from the capability of being exploited by a variety of applications such as advanced data exploration engines, Decision Support Systems and sentiment analysis. Because the majority of the online data is unstructured, high efforts are demanded to extract information from that data to be structured and represented in machine understandable semantic model by utilising different techniques and approaches such as the Semantic Web Technologies.

Recently, the research community has widely acknowledged the use of Semantic Web Technologies for knowledge representation when exploiting knowledge bases. Most of the researchers argue that Semantic Web Technologies are best placed to build a semantic knowledgebase because they are capable of organising and modelling the information into a highly structured knowledge. Also, these technologies are capable of reasoning structured knowledge to infer new and interesting facts to improve the decision-making and the intelligent exploration activities (Isiaq and Osman 2014, Konstantinova 2014, Kumar and Ravi 2016).

2.2 Reviewing Literature in Exploiting Semantic Knowledge Bases

In the literature, several studies have been conducted to investigate developing frameworks based on Semantic Web Technologies for different problems, proposes and domains. for example, Du and Zhou in (Du and Zhou 2012) proposed ontology-based framework interoperates financial data from various online sources to improve the performance of financial decision-making to provide a complete solution to data quality problems. This research utilises ontology mapping to improve the quality of online financial data. The Ontology-based Framework for Financial Decision-Making (OFFDM) consists of three components, which are Financial Ontology (FinO), online financial data resources, and financial decision-making. They developed FinO of income statements from exiting ontologies and other online financial resources, such as, Google Finance, Yahoo! Finance, and MSN Money Central. The FinO interoperates diverse financial data sources by using Ontology Mapping. The selected case study scenario used to evaluate the OFFDM framework is related to portfolio management. It is a typical case of intelligent financial decision-making to show how OFFDM is used to address the data quality problems. The process of portfolio management divided into three phases, which are Data Collection, Asset Valuation and Portfolio Optimization. According to their

results, the ontology-based method for addressing the missing-value problem is more effective for asset valuation than the traditional methods.

In addition, several studies have been conducted to investigate developing Semantic Web based applications for different proposes and domains. for example, Yoo, et al. in (Yoo and No 2014) argued that the Semantic Web Technologies can be utilised to share the economics knowledge. To demonstrate that this argument is correct, they implemented a system to share economic knowledge by using economic domain ontology. The system shares economics knowledge which can be generated and collected by system users. The system enabled the users to register economics knowledge through the registration interface and directly define the relationships among the economic variables. Then, the registered economics knowledge is transformed into semantic knowledgebase. They discussed the concepts of economic variables and their relationships in the ontology which represents the economic domain knowledge. They included three search functions to the system for sharing knowledge, basic search, knowledge navigation, and instrumental variable recommendation. The case study for applying the Semantic Web Technologies based approach showed the significance of the approach in recommending suitable Instrumental Economic Variables.

In other application area, which is search engines, Lupiani-Ruiz, et. al in (Lupiani-Ruiz, et al. 2011) established a research to present a domain-specific semantic search engine to overcome the practical limitations of the conventional search engines. This semantic search engine utilises Semantic Web Technologies and Natural Language Process Techniques in economic and financial domain. It was designed to deal with financial news, semi-structured and unstructured by developing a system which uses a financial ontology for semantic indexing and annotation of natural language documents. The Ontology development is based on existing Financial Domain Ontologies; for instance, BORO, TOVE, and the Ontologies from XBRL Ontology Specification Group. The new ontology covers four concepts, financial market, financial intermediary, asset, and legislation. This system consists of three modules, A Financial ontology module, The ontology population module and Ontology-based engine module. This ontology-based semantic search engine could be performed in two stages. Annotation stage which let the system obtains financial news from internet and annotates them with knowledge entities from the financial ontology. This process has been implemented using GATE (GATE 2018). Search stage which let the search engine analyses the natural language queries to extract the inner meaning to match it against the financial ontology to determine the knowledge entities in the user's query. This research has perfectly utilised the relevant Semantic Web Technologies to develop the semantic search engine such as modelling the targeted domain knowledge and transferring the annotated information into a semantic knowledgebase.

The FIRST project (FIRST 2013) provides an information extraction, information integration and decision making infrastructure for information management in the financial domain. This project addresses the financial domain challenges such as it is extremely large, dynamic, and heterogeneous sources of information to be highly trustable, easily acquirable information for decision makers in companies. The main objective of the semantic Information Extraction is extracting sentiments from texts with respect to the objects and their features. These objects and features are specific to the three use cases of FIRST, which are, market surveillance, reputational risk and retail brokerage. The secondary objective of the semantic Information Extraction is extracting entities from the financial domain texts with respect to the three use cases. The entities and sentiments serve as semantic features for decision models. These models would be utilised to either identify events such as market abuse, market events or forecast risks and returns as a basis for investment decision-making. The decision-making infrastructure includes a module responsible for the sentiment annotation from financial news and blog posts. They populated the ontology with sentiment objects of interest; for example, companies, stocks, countries. These objects and sentiment vocabularies were retrieved from a diversity of sources such as the IDMS database and SentiWordNet. For extracting the financial entities and sentiment polarities, FIRST has adopted an ontology-guided and rule-based approach. Its main aim is to classify the polarity of sentiment with respect to a sentiment object of interest. Although the project utilises the Semantic Web Technologies to model the sentiment semantic knowledgebase and the ontology contains the financial domain related relevant objects, they apply sentiment Rules and classification processes entirely by using JAPE rules which have been implemented by using GATE tool. According to FIRST project team, their system can be effectively used for financial sentiment extraction from texts with respect to the three use cases in FIRST.

Recently, the advantage of the achievements in the field of Semantic Web Technologies have been extensively used in decision-making processes to support tasks in several application domains such as financial investment recommendation, a clinical management, system audit management, network security management, justice and legal advice, waste-water management, power consumption management and electronic issue management. (Blomqvist 2014, Rospocher and Serafini 2012). As highlighted by Simeonov, et al. in (Simeonov, et al. 2016), the access to the distributed and heterogeneous information in the web can be unified after semantically aggregated by using Semantic Web Technologies. In this work, the authors provided a Decision Support System based on inference over semantically integrated data from diverse web resources and provides guidance to Small and Medium Enterprises for deciding in which country these enterprises could invest. The authors defined internationalisation indicators for these enterprises to provide a comparative view of the countries in question and shows 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 Semantic Web technologies. 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 this Semantic Web Technologies based tool in the market.

The study by Osman, et al. in (Osman, et al. 2014) investigated the challenges in developing a Dementia Care Decision Support System by utilising Semantic Web Technologies based on the independent assisted living environment of the patient's behaviour information. Semantic Web Technologies are used to model and integrate the information related to the context-aware scenario into a semantic knowledgebase. This use-case scenario is about the patient's dynamic behaviour observations such as occupants' movement and equipment use within the living environment. It requires to be analysed against the integrated semantic knowledgebase about the patient's condition such as age, illness history, medical advice and known symptoms. The proposed Semantic Web based Dementia care decision support system is intelligently interrelating the irregularities in patient behaviour captured by sensory devices to Dementia symptoms prescribed by clinical guidelines in order to assist medical advice. This system uses a rule-based reasoning to infer knowledge about the Dementia patient medical state. The authors have concluded that the incurred overhead by the semantic reasoning process is tolerable within the context of Dementia care decision support.

In another domain, Wanner, et al. in (Wanner, et al. 2015) question whether the ontologies in Semantic Web Technologies can be exploited as a core of Decision Support Systems 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 Decision Support System 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 Semantic Web Technologies, 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 Decision Support System 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 another complex domain, Thakker, et al. in (Thakker, et al. 2015) show how Semantic Web Technologies can be utilised to allow addressing the complex issues of pathology and Regions Of Interest (ROI) inferencing and matching experts expectations of decision-making support in tunnelling domain. A pathology is a problem that causes tunnel disorders; it is also the link between the disorders and its causes. A Decision Support System (DSS) in tunnelling domain deals with identifying pathologies based on disorders present in various tunnel portions and contextual factors affecting a tunnel and identifying the regions of interest (ROI). This complex diagnosis process is often subjective and poorly scales across cases and transport structures. The authors of this work introduce a working prototype of a DSS in tunnelling domain using Semantic Web Technologies, they call it Pathology Assessment and Diagnosis of Tunnels (PADTUN). They captured the domain-relevant key concepts and facts by the assist of tunnelling domain experts and developed an ontology from these key concepts and their interrelations. The ontology is utilised to take advantage of inferring capabilities offered by Semantic Web Technologies. They evaluated PADTUN in a real-world settings offered by the NeTTUN EU Project and is applied in a tunnel diagnosis use case with Société Nationale des Chemins de Fer Français (SNCF), France.

Since large amount of data is still published in unstructured format, it is crucial to transfer this data into machine understandable format by utilising Semantic Web Technologies for exploring knowledge to support decision-making processes. Next section will present some of the Natural Language Processing tools which are utilised to extract information from unstructured data.

2.3 Reviewing Natural Language pre-processing tools

The unstructured format of data is a fundamental challenge in Information Extraction because it requires to be transformed into structured knowledge that can be queried by software agents. As aforementioned, Information Extraction could be considered as a pipeline process. In each stage of that pipeline, the tasks of Natural Language Processing are applied in order to obtain valuable information from natural language texts. However, Natural Language pre-processing plays a significant role in Information Extraction pipeline process because the high-quality Natural Language Processing tasks will return high-quality Information Extraction process.

The Information Extraction process usually starts in recognising the named entities; then, identifying identity relation between named entities, which is co-references resolution.

Lastly, extracting the relation between the named entities in a certain event. However, recognising Named Entities is a fundamental task and a core process of Information Extraction because it is used directly in many application domains such as proteins and genes identification. Also, recognising Named Entities is considered as a pre-processing step by other application domains such as extracting the relationship between stock prices increase or decrease and companies in the news. Named Entity Recognition can be considered as prerequisite task that can be met by standard techniques. As a result, research and commercial communities have spent efforts to publish Natural Language Processing tools to perform Named Entity Recognition and other tasks in the Information Extraction process pipeline (Atdağ and Labatut 2013, Rizzo and Troncy 2012). Below is a list of some of these tools.

1. **AlchemyAPI:**

It uses Machine Learning and Natural Language parsing technology for analysing web or text-based content for Named Entity Recognition, sense tagging, as well as for relationships and topics. It is available as a demo web application or as a REST service, also for mobile SDKs. However, It is acquired by IBM in 2015 and its technology is now a core component of the cognitive APIs offered on IBM's Watson Developer Cloud. This tool has been adopted by several works to recognise the Named Entities such as the work of Saif, et al. in (Saif, et al. 2014).

2. **Apache OpenNLP:**

It is a Machine Learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, Part Of Speech tagging, Named Entity extraction, chunking, parsing, and co-reference resolution. These tasks are usually required to build more advanced text processing services. OpenNLP, also, includes Maximum Entropy and Perceptron based Machine Learning. The goal of the OpenNLP project is to create a mature toolkit for the abovementioned tasks. An additional goal is to provide a large number of pre-built models for a variety of languages, as well as the annotated text resources that those models are derived from. Previous studies have based their Named Entity Recognition on this tool; for example, the study of Kovačević, et al. in (Kovačević, et al. 2013). They collected the candidate clinical department names from their datasets.

3. **Stanford CoreNLP:**

An integrated suite of Natural Language Processing tasks for several human languages, which are English, Spanish, and (mainland) Chinese. The Natural Language Processing tasks include tokenisation, POS tagging, Named Entity Recognition, parsing, and co-reference. It has been developed by the Stanford NLP Group, which can be incorporated

into applications with human language technology needs. These packages are widely used in industry, academia, and government. This tool has been used in many investigational studies such as the work of Liu, et al. in (Liu, et al. 2017). They adopted it to generate entity mentions and get POS tags features for their datasets.

4. **Thomson Reuters Open Calais:**

It offers an accurate way to tag Named Entities, facts and events in unstructured data to increase its value, accessibility and interoperability. It uses Natural Language Processing and Machine Learning algorithms trained by hundreds of Thomson Reuters' Editorial teams for several years to offer the industry's best combination of company extraction and relevance. A variety of works have adopted this tool to perform Named Entity Recognition. For instance, the authors in (Saidi, Amer-Yahia and Bahloul 2014) used this tool annotate the documents in their corpus to extract entities, types and categories.

5. **GATE, General Architecture for Text Engineering:**

GATE tool can be utilised to develop language process applications by using GATE Developer and GATE Embedded. GATE Developer or IDE is used for visualisation of the data structures produced and consumed during processing, and for debugging and performance measurement. GATE Embedded is an API Java libraries that are used to embed GATE-based language processing facilities in an application framework. The functionality of GATE to process natural language is constructed in various types of component, which are Language, Processing and Visual Resources. Language Resources represent data components such as the corpora of documents. Processing Resources represent the primary natural language algorithms to automatically create and manipulate annotations on documents such as POS taggers. Lastly, Visual Resources that represent visualisation and editing components that participate in GATE Graphic User Interface. These process resources in GATE are used to implement various Natural Process tasks such as tokenisation, parsers, morphology, tagging for various languages. These tasks are performed by GATE Developer and GATE Embedded to develop automatic linguistic processing applications. While GATE is distributed with a number of core Process Resources, there are more process resources which are developed and made available by other GATE developers. They are included to GATE as plugins; for example, Stanford parser process resource. In addition, there are built in applications for automatic linguistic analysing the natural language texts. For example, ANNIE (A Nearly-New Information Extraction system). ANNIE pipeline is for Natural Language Process and Named Entity Recognition tasks (Cunningham, Maynard and Bontcheva 2014). ANNIE pipeline has been adapted to many different application domains with acceptable results. For example, it was adapted to recognise named entities in the work of Ruiz-Martínez, et al. in (Ruiz-Martínez,

Valencia-García and García-Sánchez 2012) to analysis sentiment polarity in financial news domain.

2.4 Conclusion

Semantic Web Technologies are best placed to build such domain-specific knowledge as they are capable of organising and modelling the information into a highly structured knowledge in order to assist machines to understand information published on the Web. They describe and combine the corresponding relation between the concepts' instances from different sources and infer more information about these concepts in different contexts. They argue that Semantic Web Technologies able to infer new information in order to deliver relevant, reliable and accurate information to a user when and where it is needed towards making a particular decision. In this context, Semantic Web is an extension of the World Wide Web, whose contents can be accessed, shared and explored without human intervention.

A considerable amount of literature has been published on exploiting semantic knowledge bases by utilising Semantic Web Technologies. In these studies, Semantic Web Technologies have been extensively used in different application types such as developing Decision Support Systems and exploring semantic knowledge bases in different problem domains; for example, a clinical management, environmental conditions, economic and finance, justice and legal advice and tunnelling issues domain.

As the result of this literature survey, we concluded that despite the enthusiasm of the research community about the Semantic Web, more efforts are required to contribute towards creating a unifying framework that facilitates the interoperation of intelligent agents or reasoning engines. In this research, we will investigate developing knowledge-based framework to integrate exploiting semantic knowledge bases and supporting decision-making activities in specific domains.

3 The Proposed Framework for Domain-Specific Information Exploration and Decision-Making

3.1 Introduction

According to Buranarach, et al. in (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 process. The issue of semantic data publishing is about how to transform unstructured data into structured data and mapping them to existing semantic dataset. The issue of semantic data consumption process is about how to discover, access and process 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. In fact, it is challenging to extracting information from unstructured data to be semantically constructed in a knowledge bases. In this research, we intend to investigate these issues by proposing a comprehensive framework 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 exploiting the resultant semantic knowledgebase to support knowledge exploration in the context of decision-making activities.

This research proposes a knowledge-based framework that has a roadmap for linking several components of different techniques and tools. The framework focuses on providing reusable and configurable data and application templates, which allow the users to apply it in diversity of domains. The framework allows the application developers to focus on domain problems rather than the tools, techniques and approaches of the application. The framework covers a diversity of disciplines and techniques, which are knowledge representation, automatic information extraction from unstructured data, constructing a semantic knowledgebase from different sources and consuming semantic knowledgebase by intelligent exploration and support decision-making.

In this chapter, we describe this research use-case motivating scenario for knowledge-based application, Semantic-based Decision Support Systems for stock investments, and, the objectives and phases of the proposed framework.

3.2 The Motivating Scenario

Intelligent techniques can be employed on the semantic knowledgebase to generate new information that can support users in making the correct decisions or intelligently explore the semantic knowledgebase to query about a company or country situation; for example, information relating to the stock investment decision-making process.

The online data, unstructured, semi-structured and structured, in financial and economic domain is considered an important domain because there is an extensive information on different topics including information related to stock market and shares. Hence, the motivating use-case scenario is about supporting users in stock investment decision-making process.

Generally, the prediction of stock prices is a very difficult task as it behaves like 'random walk' process and the prediction might be out of control due to some unexpected concerns that have direct impact on the targeted company performance. In fact, there are many factors impact the performance of any company who is under observation by investor; accordingly, its shares price. These factors should be considered when making a decision of buying, holding or selling those shares. The factors could be classified into macroeconomic or microeconomic. Macroeconomic factors are the external factors that are affected by a national economy as a whole. Microeconomic factors are the internal company specific factors (Hunjra, et al. 2014).

Since years ago, there are different analysis methods have been arisen to obtain answers on what share to buy and when to buy and sell the share of the targeted company. Some of these methods analyses the basic financial factors such as sales, profit margin and other factors. The researchers who perform these financial analysis methods heavily rely on statistics and they will be looking through the auditor's reports, the profit-and-loss statement, balance sheet, dividend records, and internal developments of the companies whose shares is under their observation. The investors who rely on the results of these methods will purchase stocks that are viewed as under-priced. Other methods are looking at how shares' prices perform. They focus on the shares prices to assess and evaluate the demand and the supply for the shares based on the market prices without considering the news about shares. The developers of these methods believe that the market itself is the best source of data. They believe that all the investors' reactions towards all the information regarding the security already embedded in the share price. However, many analysts have argued that public financial news impacts the stock price. They believe that investors are motivated by the public news that are related to microeconomic and microeconomic factors. They will also analyse business activities, accounting errors, scandals and regulatory information to estimate the company's future business condition (Yong and Taib 2009).

Investors could face some difficulties in processing the available information by him/herself when making stock investment decisions because using the right formulas which are

suitable for decision-making process problem requires specialist expertise. As result, the process of financial decision-making in general and stock investment decision-making in specific are supported by decision-making systems. These systems could be utilised to access a vast amount of data over different sources, private and public. Decision Support Systems (DSS) are designed to assist decision makers to improve the decision making process. They can be defined as machine-based applications that support people and organisations in their decision-making processes from problem formulation to decision recommendation (Songsangyos and Iamamporn 2014).

Decision Support Systems can be divided into the following main categories: Model-driven DSS, Data-driven DSS, Document-driven DSS, Communication-driven DSS and Knowledge-driven DSS. In Knowledge-driven category; for example, DSS recommends or suggests actions to the users, rather than just retrieve information relevant to a certain decision; in other words, these systems try to perform some part of the actual decision-making for the user through special-purpose problem-solving capabilities (Yong and Taib 2009, Simeonov, et al. 2016).

3.3 Semantic-based Decision Support Systems for Stock Investment

Semantic Web Technologies are best placed to build such domain-specific knowledge as they are capable of organising and modelling the information into a highly structured knowledge in order to assist machines to understand information published on the Web. They describe and combine the corresponding relation between the concepts' instances from different sources and infer more information about these concepts in different contexts (Aljamel, Osman and Acampora 2015). We argue that Semantic Web Technologies share many goals with DSS; for example, being able to infer new information in order to deliver relevant, reliable and accurate information to a user when and where it is needed towards making a particular decision. DSS field has taken advantage in the last decade's achievements and results of Semantic Web Technologies. For example, rule reasoners and ontology reasoners of the Semantic Web Technologies have been recently adopted in DSS in various purposes such as reasoning some of the decision support phases, to characterise the data manipulated by the DSS and to define the tasks and parameters of the various modules of the system (Blomqvist 2014, Rospocher and Serafini 2012).

Semantic Web Technologies have been extensively employed in decision-making processes in several application domains. In this research, we would like to investigate the usability of Semantic Web technologies to develop a semantic-based stock investment

Decision Support System. We broadly follow the approach advocated by Rospocher and Serafini in (Rospocher and Serafini 2012) that defines three phases in a decision-making process, which are the formulation of the decision-making problem, the integration of the relevant data for the given problem and the reasoning on the semantic knowledgebase to make a decision.

3.3.1 The Formulation of the Decision-making Problem

This phase is about formulation or modelling of the decision-making problem, i.e. the stock investment decision-making problem. Before explaining the modelling in detail, we should introduce background information about stocks.

Typically, each stock makes the investor who owns that stock also owns a share of the corporation. Investor receives benefits in the form of dividends, capital gains or both. There are several types of stock; however, we will limit our details to Common stocks. Holders of common stock exercise control by electing a board of directors and voting on corporate policy; nevertheless, they are on the bottom of the priority ladder for ownership structure. In the event of liquidation, common shareholders have rights to a company's assets only after bondholders, preferred shareholders and other debtholders are paid in full. Historically, common stocks have provided a higher return though they have a higher risk. An investor earns capital gains (the difference between the purchase price and selling price) when he/she sell at a higher price than the purchase price (Levišauskait 2010).

Investors should perform a decision-making analysis for stock investment including macroeconomic or economic analysis to describes the economic situation in a 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 releases on earnings and profits, and future estimated earnings 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. Next subsections will present the details of these analyses (Levišauskait 2010, Mian and Sankaraguruswamy 2012, Li, et al. 2014a).

3.3.1.1 Country Economic Analysis (Macroeconomic Analysis)

The macroeconomic analysis is about analysing the behaviour of economics in the context of economic cycle. The economic cycle is the natural rise and fall of economic growth that occurs over time. Each economic cycle has four phases, expansion, peak, contraction and trough. These economic cycle phases do not occur at regular intervals. A well-managed

and stable economy can remain in the expansion phase for long time (Kim and Burnie 2002). There several recognisable economy indicators to measure the economies' stability. According to Cashell in (Cashell 2006) and Yelwa, et al. in (Yelwa, David and Awe 2015), the most important economy indicators are Gross Domestic Product (GDP), inflation and unemployment rates. GDP rate measures the growth economic output value of all products and services produced inside the boundaries of the measured economy, country. Inflation rate measures the cost of living by using the consumer price index. The unemployment rate measures the number of unemployed individuals by all individuals currently in the labour force.

The stability of macroeconomic cannot be measured by just one of these indicators apart from the others because they are interdependent. Generally, the economists reveal that making the economy stable means encouraging the increase of GDP rate while lowering unemployment rate, in the meantime, this should be balanced against inflation rate, which might occur if the GDP rate is increased rapidly. If the inflation rate is slightly high and under control, it could encourage companies to increase production because of the high demand on products and services. This will improve the overall GDP. As a result, the stock market will be strengthened because investors are always preferring companies' profitability. On the other hand, if the GDP rate is very high, inflation rate will increase also. This will consume stock market gains and make them less valuable. Thus, a careful balance should be maintained between these indicators in order to keep the economy stable (Cashell 2006, Yelwa, David and Awe 2015, Levišauskait 2010, Gokal and Hanif 2004).

There is a widespread support between the researchers that there are complicated relationships between these indicators. However, there is no precise guidelines for a known critical or threshold values for them because there can be other events can temporarily affect the relationships between the economy indicators. For example, unexpected change in oil prices can cause a temporary rise in the rate of inflation even with relatively high unemployment rates. However, we adopted the threshold values of GDP, unemployment and inflation rates that is provided by Cashell in (Cashell 2006) and Pollin, et al. in (Pollin and Zhu 2006). These values are presented in Table 3.1 below.

Table 3.1: Economy Indicators Vlaues Thresholds for Stable Economy

Indicators Name	Minimum Rate	Maximum Rate
GDP Rate	2.5	3.5
Unemployment Rate	5	6
Inflation Rate	3	5

Table 3.2 below shows an example of the relation between economy indicators and the stability of the economy. In this table, stable means that the indicator rate value is between minimum and maximum values thresholds. High means that the indicator rate value is greater than maximum value threshold and Low means that the indicator rate value is less than minimum value threshold.

Table 3.2: The relation between Economy Indicators and Economy Stability

No	GDP	Unemployment	Inflation	Economy Situation
1	Stable	Stable	Stable	Safe to Invest
2	Stable	Stable	High	Safe to Invest
3	Low	High	High	Risk to Invest
4	Low	High	Low	Risk to Invest
5	Stable	Low	High	Risk to Invest

These relations are provided by Kolovson in (Kolovson 2014). They are general relationships between the economy indicators will be used in this research as an example for the use-case scenario; however, there can be more relations and factors occur independently.

Economic situation in a specific country influences the profitability of company's stocks in the that country. The confidence in stock markets increases when the prices of the stocks continue to grow. Usually, this occurs in the expansion phase (Levine 2012, Reilly and Brown 2011).

Usually, individual investors are affected by the positive or negative public announcements of the economic indicators because the process of making decisions and executing trades could take different amount of time for buy or sell stocks. In general, their behaviour is central to the stability of country's economy (Nofsinger 2001).

3.3.1.2 Company Analysis (Microeconomic Analysis)

There are two most frequently forms of analysis are used, technical analysis and fundamental analysis. Technical analysis involves the analysis of market prices in an attempt to predict future price movements for the particular financial asset traded on the market. This analysis examines the trends of historical prices and is based on the assumption that these trends or patterns repeat themselves in the future. Fundamental analysis is focused on the evaluation of intrinsic value of the stock price. From an investing prospective, the best evidence to consider the stock as growth investment is an increasing price over time. This analysis is performed on the current and historical data to predict to

find out the intrinsic value of the stock price (Levišauskait 2010, Agrawal, Chourasia and Mittra 2013, Schumaker and Chen 2009).

This analysis includes the examination of the market value ratios. These ratios provide investors with a shortest way to understand how much attractive is the stock in the market. To look for long-term investment decisions, investor must analyse not only the current market results, but also assess the potential of the company to generate earnings in the future to receive the whole picture of the financial condition of the company. The valuation theory to model the stock investment problem is grounded on the assumption that stock market prices reflect the fundamental value. Fundamental value is that the value of a stock investment that can be held over a long term. According to Levišauskait in (Levišauskait 2010), the stock valuation process include:

1- Forecasting of future cash flows for the stock.
2- Forecasting of the stock price.
3- Calculation of Present value of these cash flows. This result is called intrinsic (investment) value of stock.
4- Comparison of intrinsic value of stock and current market price of the stock and decision making: to buy or to sell or to keep the stock.

There are three methods of stock price valuation, income capitalization method, Discounted Dividend Models (DDM) and valuation using multiples. Discounted Dividend Models is based on the method of income capitalisation and considers the stock price as the discounted value of future dividends, at the risk adjusted required return of equity, for dividend paying firms, where the dividend is a share of earnings a company pays to stockholders. An important assumption behind the DDM is that the only way a corporation can transfer wealth to its stockholders is through the payment of dividend. It is because dividends are the only source of cash payment to a common stock investor. There are various types of DDM, depending upon the assumptions about the expected growth rate in dividends. They are “Zero” growth DDM, constant growth DDM and Multistage growth DDM. In fact, the selection of the appropriate benchmark to evaluate the stock price is a difficult decision. In the use-case scenario of this research, we will apply the constant growth DDM, it is also called Gordon Growth Model, because it is simple, powerful and convenient method of valuing stocks prices (Levišauskait 2010, Amiri, Ravanpaknodezh and Jelodari 2016).

The constant growth DDM (Gordon growth model) relates the value of a stock to its expected dividends in the next year time period, the required rate of return by investor and the expected growth rate in dividends. The intrinsic value of the stock can be calculated as in equation (3.1) below (Damodaran 2012, Amiri, Ravanpaknodezh and Jelodari 2016,

Reilly and Brown 2011, Levišauskait 2010):

$$V = \frac{D_1}{k - g} \quad (3.1)$$

Where,

V = Intrinsic value of the stock

D_1 = Next year's dividend value

k = Required rate of return for stock investors. Also known as discount rate or capitalisation rate.

g = Dividend growth rate.

In constant dividends growth rate (g), if in the last year a company payed (D_0) dividend, then in the next year period its dividends (D_1) will grow at growth rate (g) and it can be found as in equation (3.2) below:

$$D_1 = D_0(1 + g) \quad (3.2)$$

The valuation equation will be as in equation (3.3) below:

$$V = \frac{D_0(1 + g)}{k - g} \quad (3.3)$$

Because dividends growth rate (g) is constant, it can be calculated by using the historical dividends paying of two sequenced years; for example, if the dividend payed in previous year (i) is (D_i) and the dividend payed in previous year (i-1) is (D_{i-1}), then the dividend growth rate two years ago as in equation (3.4) below:

$$g = \frac{D_i - D_{i-1}}{D_{i-1}} \quad (3.4)$$

Where,

D_i = The dividend value in (i) year ago, i >= 1

From the constant-growth DDM, we can infer the market capitalisation rate or expected return rate or the rate of return demanded by investors (k). This rate can be calculated by applying formula in equation (3.5) below:

$$k = \mathrm{y} + g \quad (3.5)$$

Where,

k = Expected Return Rate

y = Dividend Yield Rate

g = Dividend Growth Rate

For more details about equation (3.5), return to Spaulding in (Spaulding 2017) and for the other equations above return to Levišauskait in (Levišauskait 2010).

In the constant-growth DDM that presented in equation (3.3), we applied the following assumptions:

1- Dividends grow at a constant rate.
2- The constant growth rate will continue for an infinite period.
3- The required rate of return (k) is greater than the infinite growth rate (g). If it is not, the model gives meaningless results because the denominator becomes negative.

The view of stock investment analysts is that the intrinsic value can divide a company's estimated future earnings by the number of its existing shares to determine whether a stock's current price is a bargain. This measure allows investors to make decisions based on a company's future potential independent of short-term enthusiasm or market hype. After complete the valuation by calculating the intrinsic value of the stock, the decision-making for investment in stocks will as follows (Levišauskait 2010),

- If the current stock price (P_0) is less than the intrinsic value of the stock, the decision should be to buy or keep the stock because it is under valuated.
- If the current stock price (P_0) is greater than the intrinsic value of the stock, the decision should be to sell the stock because it is over valuated.
- If the current stock price (P_0) equals the intrinsic value of the stock, the decision depends on the additional observations of investor. It could be keep or buy the stock.

3.3.1.3 Online News Analysis

Besides economy and business indicators, it is important for the investors to review the news articles' contents periodically because the prices of stocks are sensitive to relevant events published in news articles' contents. In fact, there are controversial opinions of how online news can impact stock prices. According to Mian, et al. in (Mian and Sankaraguruswamy 2012), the companies' news releases on earnings and profits, and future estimated earnings is one of the factors that affect their shares prices. In addition, the opinion of Li, et al. in (Li, et al. 2014a) that in spite the fact that there is a delay between data published by indices themselves and the online news articles about the these indices and their constituent stocks, it is important for the investors to review the news articles'

contents periodically. Li, et al. in (Li, et al. 2014b) mentions the significant relationship between news sentiment and stock prices. They argued that when the news sentiment can be identified as positive or negative, the stock prices are affected by this news. Negative sentiment news will make the investors sell stocks and positive sentiment news will encourage the investors to buy or hold stocks.

Companies are required by law to keep shareholders up to date on how they are performing. Some of that information is published in the news which summarises the company's operations for individual investors. The news could contain a critical information about illegal activities such as accounting errors, scandals and uncertainty about firms' earing issues. Usually, investors base their expectations on a company's sales and earnings as evidence of its current strength and future potential. When a company's earnings are up, investor confidence increases and the price of the stock usually rises. Stock markets are sensitive to public information and with the growing popularity of that information in the Web, the reaction of the investors should be very fast to succeed the investment. It is because waiting for more information could trap their investment in the downdraft, then they could lose worthy investment opportunities (Mian and Sankaraguruswamy 2012, Li, et al. 2014a)

In our approach, we extract information from unstructured online news and present it to investors as structured information to be easily explored and understood by machines. The investor can use this information as a negative or positive indicators to proceed in stock investment process.

3.3.2 Retrieving the Relevant Data for the Given Problem

This phase can be represented by information module, which allows to store the information into a semantic knowledgebase to be processed by the DSS. As we indicated in the formulation phase section above, 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 types of information that could be retrieved from other sources, such as the details of companies' products and employees.

The system retrieves the required information for stock investment decision-making process to be provided to the reasoning phase to produce a decision. This information could be collected from several sources to be integrated and modelled in a semantic knowledgebase by using Semantic Web Technologies or ontology. Examples of these data sources are unstructured online news, semi-structured data sources and structured online datasets. The

relevant information in semantic knowledgebase is selected to be processed to produce a corresponding decision advice.

3.3.3 Reasoning on the Semantic Knowledgebase

This phase can be represented by reasoning module, which implements the decision support strategy. It is triggered by the information in the semantic knowledgebase according to the user decision support request. Producing and delivering the decision depends on the applied techniques to implement the DSS. We implemented a module responsible for decision making which based on compiling both classification rules that are hard wired into the knowledge-base's semantic model such as first predicate logic's Necessary & Sufficient conditions, and also explicit the rule-based reasoning computation to classify events and make decisions that might be of importance to end users.

The overview of the framework implementation scenario is depicted in Figure 3.1 below.

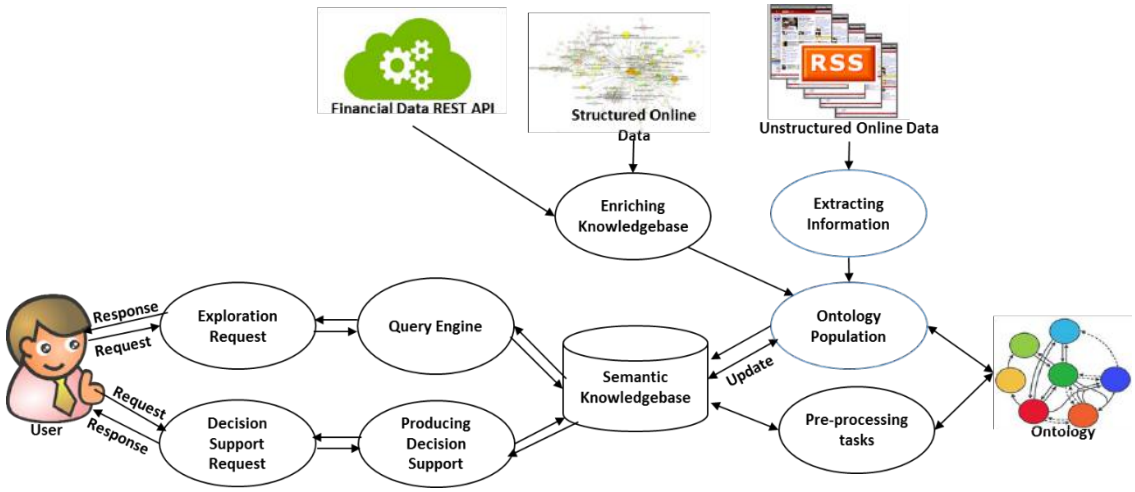

Figure 3.1: The overview of the framework implementation scenario

The starting point in this diagram above is constructing and enriching a semantic knowledgebase from the data of targeted problem domain by sourcing unstructured, semi-structured and structured data. The system describes the user request to select the background knowledge to gather, store, and integrate the information relevant for a requested decision-making problem. The system checks whether the requested information is available in the semantic knowledgebase. If it is not available, the system attempts to extract that information from the relevant data sources. The system applies Rule-based reasoning over the relevant information to produce recommended decision which is based

on the available knowledge and the details of the users' request. The system will deliver that recommended decision to the user and present the information that is used to make the decision to the user.

Providing a semantic solution to such scenario requires handling a semantic knowledgebase from diversity data sources. It requires, also, a framework to provide a road map to an knowledge-based application developers and recommends approaches, techniques or tools for every stage of the use-case scenario; for example, how to extract information and how to model the information in the knowledgebase.

3.4 The Framework's Objectives

As aforementioned, the framework adopted a knowledge-based approach to extract information, construct a semantic knowledgebase, handle the resultant semantic knowledgebase and deliver inferred facts to end users. The adopted approach exploits the domain knowledge to improve the fundamental information retrieval tasks of Named Entity Recognition by enriching the gazetteer listing of some entities; for instance, persons, locations and organizations and improving the Relation Classification by investigate improving the supervised Machine Learning technique factors, which are compiling the training datasets and selecting the best features.

The implementation of the knowledge-based framework relies on constructing and exploiting a semantic knowledgebase. This knowledgebase can be further processed and arranged with advanced reasoning techniques to infer new and interesting facts from the sourced domain data to be intelligently explored by end users (Aljamel, Osman and Acampora 2015). However, we require clear objectives to fulfil the answer of the research questions. These objectives can be summarised into the following:

1- Analysing the domain knowledge to understand its semantic and syntactic characteristics to be utilised in building domain's knowledge map. This knowledge map is used to describe the prearranged vocabulary and semantic structure for exchanging information about that domain.
2- Translating the knowledge map into a formal semantic model, ontology, that defines concepts and their relations in Semantic Web Technologies' standard languages.
3- Investigating the representation of domain's non-binary relations by using standard Semantic Web Technologies.
4- Retrieving a domain-specific unstructured online data and applying a boilerplate removal to extract full clean text from HTML pages.
5- Adopting Semantic Web based approach to utilise available semantically tagged online datasets to inform Information Extraction process in collecting gazetteer lists for recognising named entities.

6- Conducting the linguistic pre-processing tasks or Natural Language Processing (NLP) tasks to be used for Named Entity Recognition, relation detection, features extraction.
7- Adopting a supervised Machine Learning based relation classification. The supervised Machine Learning algorithms are configured and evaluated to improve their performance in relation classification problem.
8- Investigating the approaches of generating labelled instances for training the relation classifiers, manually by domain experts and automatically by adopting Semantic Web based approach to utilise available semantically tagged online datasets.
9- Exploiting the domain knowledge and rule-based approach to create a new set of features for supervised Machine Learning relation classification and investigating the application of Genetic Algorithms for features selection.
10- Further exploring the impact of the features combinations on the relation classification models accuracy in extracting relations from unstructured data.
11- Populating the Knowledgebase to transform unstructured data into instances of the concepts and relationships defined in the ontology to relate text to ontology.
12- Enriching the resulting knowledgebase by utilising publicly available datasets that apply the same standardised metadata. These datasets could be used to publish ground facts that are relevant to our problem domains.
13- Investigating the application of advanced reasoning techniques on the resulting knowledgebase in order to extract new and interesting facts to improve Intelligent Exploration of the semantic knowledgebase and assist the implementation of Decision Support Systems.

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

3.5 The Framework's Phases and Tasks

As we explained early, this framework is based on domain-specific knowledge-based approach. The objectives of the framework should be transferred into tasks to be performed. These tasks are implemented by using a diversity of algorithms, methods, approaches, techniques and tools, which are mainly related to these four disciplines, Natural Language Processing (NLP), Semantic Web (SW) Technologies, Machine Learning techniques (ML) and Evolutionary Algorithms (EA). These tasks can be categorised into four main phases. Moreover, the phases and their tasks 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. Below, is the description of the four framework's phases.

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.

Phase two (Natural Language Pre-processing, Named Entity Recognition and Relation Classification):

Applying the Natural Language pre-Processing and Named Entity Recognition 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.

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.

Phase four (Applying Reasoning Techniques and Exploiting the Semantic Knowledgebase):

Investigating 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 3.2 below.

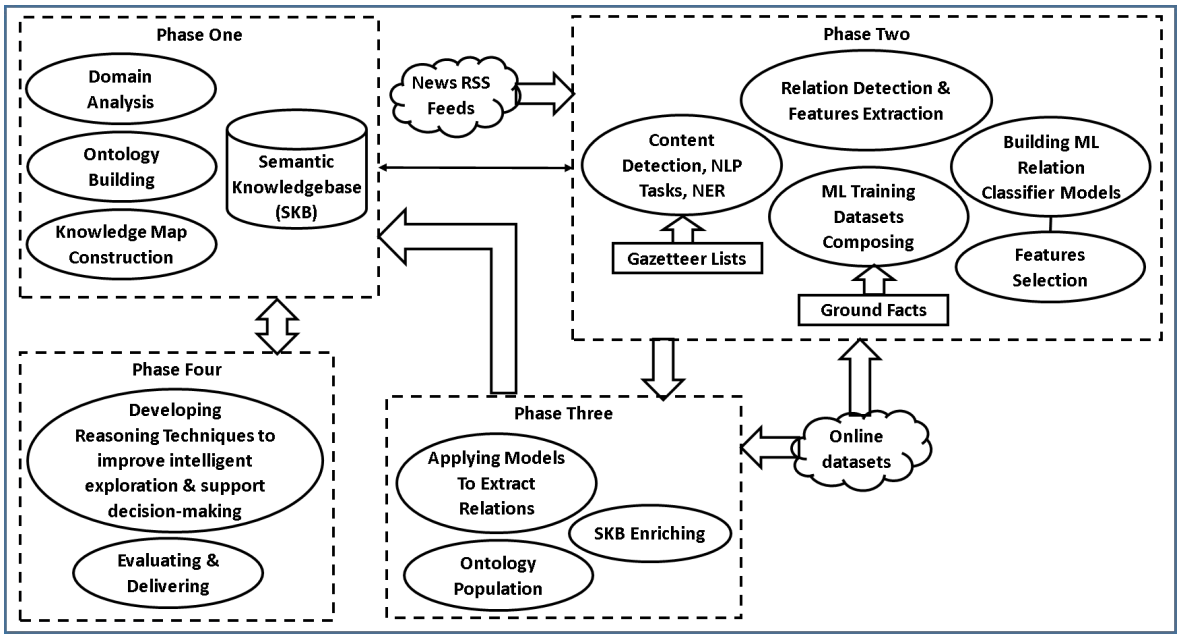

Figure 3.2: The Four phases of The General Framework

The tasks of the proposed knowledge-based framework' phases will be presented in detail in the next chapters. The tasks of phase one, analysing and modelling the problem domain knowledge will be described in chapter 4. The tasks of phase two, extracting information from online unstructured data will be presented in chapters 5, 6, and 7. The tasks of phase three, populating the extracted information into a semantic knowledgebase and enriching it by sourcing semi-structured and structured online datasets, and the tasks of phase four, applying reasoning and exploring techniques on the resulting semantic knowledgebase to support the process of decision-making process will be presented in chapter 8.

3.6 Summary

This research proposes a domain-specific knowledge-based framework that has a roadmap of integrating several components of different techniques and tools. The framework focuses on providing reusable and configurable data and application templates, which allow the users to apply it in diversity of domains. The framework allows the application developers to focus on domain problems rather than the tools, techniques and approaches of the application. It covers a diversity of disciplines and techniques, which are knowledge representation, automatic information extracting from unstructured data, constructing a semantic knowledgebase from different sources and consuming semantic knowledgebase by intelligent exploration and support decision-making process. The decision-making process phases can be categorised into formulation of the decision-making problem,

integration of the relevant information for the given problem and reasoning on the relevant information to make a decision.

In this research, we adopted stock investment decision-making use-case scenario. In this use-case scenario, we employed three types of information. They are company's information such as the current stock price and the historic dividend values, country's economic indicators such as GDP, inflation and unemployment rates and company's events online news such as profit margin increase or decrease or share prices increase or decrease. There are other types of information that could be retrieved from other sources such as the details of companies' products and employees. That information is reasoned to deliver inferred facts; then, produce recommended decisions for end-users. This information could be collected from several sources to be integrated and modelled in a semantic knowledgebase by using Semantic Web Technologies or ontology. Examples of these data sources are unstructured online news, structured online datasets and semi-structured data sources.

The proposed knowledge-based framework has four phases, phase one is about analysing and modelling the domain knowledge, phase two is about Natural Language Pre-processing, Named Entity Recognition and Relation Classification, phase three is about constructing and enriching the semantic knowledgebase and phase four is about applying reasoning techniques and exploiting the semantic knowledgebase. The main objectives of this framework are, extracting information from unstructured data sources, constructing a semantic knowledgebase, reasoning the resultant semantic knowledgebase and delivering the inferred facts to end users with reference to the use-case scenario.

The following chapters describe in details the tasks of our proposed knowledge-based framework. Chapter 4 will describe phase one, chapters 5, 6, and 7 will describe phase two and chapter 8 will describe phases three and four.

4 Domain Data Modelling for Bridging the Gap between Data and Knowledge

4.1 Introduction

This chapter presents the details of the first phase of the framework, problem domain knowledge representation or modelling. It is a crucial issue; especially, if this knowledge requires to be effectively processed and reasoned as a part of Decision Support Systems (DSS) (Jovic, Prcela and Gamberger 2007, Castells, et al. 2004). The process of delivering effective knowledge representation techniques and inference mechanisms are an important task. In addition, constructing a domain knowledge in a semantic model is an important step in developing semantic knowledge-based applications.

There is an opportunity in the increasing availability of domain-specific knowledge in the Web because understanding the syntactic and semantic characteristics of the domain knowledge is a key to the success of Information Extraction and then semantic modelling the extracted information. A domain-specific knowledge's entities, concepts, relationships and events play a central role in realising the full potential of domain's semantic modelling because they are fundamental to semantics and associate meanings to words, terms and entities and also to infer new information insights (Perera, et al. 2012).

We intend to utilise Semantic Web Technologies as the modelling tool for our targeted domain knowledge as they facilitate the organisation of information into a highly-structured knowledgebase that can be comprehended and processed by software agents. Semantic model describes and combines the corresponding relation between the concepts' instances from different sources and infer new information about these concepts in different contexts and enables the sharing and reusing of domain knowledge. These advantages of semantic domain models have been widely investigated and confirmed by several works such as the works of Du, et al. in (Du and Zhou 2012), Lupiani-Ruiz, et al. in (Lupiani-Ruiz, et al. 2011), Yoo, et al. in (Yoo and No 2014) and Wang, et al. in (Wang, et al. 2004).

A series of ontology building methodologies have been reported on the literatures. They describe various steps and tasks to be followed when building ontology. However, each of these methodologies following different approaches. Examples of these approaches are the Cys, Uschold and King, Gruninger and Fox, Sensus, the METHONTOLOGY and On-To-Knowledge methodologies. Since these methodologies are not widely accepted and there is no technological support for most of them, they cannot be easily applied in the ontology construction task. In addition, there is no correspondence between some of these methodologies and ontology building tools. In fact, most of the tools just focus on few tasks,

which are described by these methodologies (Dombeu and Huisman 2011, Rekha and Syamili 2017, Beck and Pinto 2002). In this research, we followed the methodology which is described by Beck, et al. in (Beck and Pinto 2002) and Dombeu, et al. in (Dombeu and Huisman 2011). They reveal that ontology building is a process that composed of a serious of stages which are, specification, conceptualisation, formalisation and implementation; nevertheless, there are more activities that should to be performed to achieve each of these stages. Specification task is about identifying the purpose and scope of the ontology. Conceptualisation is about describing the conceptual model for the ontology to meet its specification. Formalisation is about transforming the conceptual description into a formal model. At last, implementation is about implementing the formalised model in a formal knowledge representation language, ontology.

However, we would like to emphasise on some fundamental rules in ontology design which are mentioned by Noy, et al. in (Noy and McGuinness 2001). These rules can assist in making design decisions. The first rule is that domain modelling depends on use-case scenario requirements; hence, there is no one correct way to model a domain. The second rule is that the ontology development is necessarily an iterative process. Lastly, concepts in the ontology should be close to objects and relationships in our domain of interest.

In the next subsections, we will explain the stages of developing the semantic model of our domain knowledge, ontology.

4.2 Identifying the Purpose of the Semantic Model, Ontology Specification Task

4.2.1 Overview

The specifications in an ontology should be limited to knowledge about a particular domain of interest rather than covering a broad range of related topics. The narrower the scope of the ontology model for the domain, the more an ontology engineering can focus on logic based constraints to describe the details in that domain (Grimm, Hitzler and Abecker 2007).

A number of studies such as Davies, et al. in (Davies, Studer and Warren 2006) and Slimani in (Slimani 2015), have found that ontologies could be categorised based on their scope as, upper-level ontologies, domain ontologies, task ontologies and application ontologies. As shown in Figure 4.1 below, generic ontologies in the upper-level could be imported by ontologies at lower levels to add a specific knowledge. Application ontologies employ both domain and task ontologies to describe the role of domain-specific concepts in specific tasks. In conclusion, the development of application ontologies benefits specific tasks to be performed within specific domains. In this research, we attend to develop a domain-specific ontology for a specific application and a range of tasks where the domain knowledge and the application tasks are represented and described in the ontology by vocabularies about

domain concepts and their relationships. Consequently, we should identify the purpose of the semantic model or ontology.

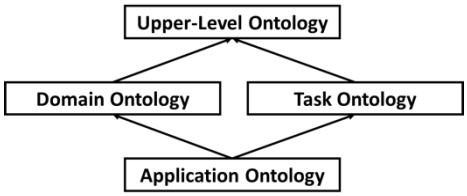

Figure 4.1: Ontologies Categorisation

Determining the domain scope of the semantic model needs to answer several competence questions such as: What is the use-case scenario of this ontology? Who will use that application? What types of questions should be asked to the modelled knowledge and what answers should be provided? Answering these questions will support understanding the scope of the ontology to capture the relevant concepts and their interrelations in the problem domain. Moreover, domain analysis and knowledge acquisition should be established to identify, capture and organise the information used in a particular domain for the purpose of making it available in an ontology, which should be based on comprehensive sources of knowledge. Knowledge acquisition activity is about acquiring the domain knowledge that will be modelled in terms of the intended motivation use-case scenario and the specified scope of the ontology. (Castro, et al. 2006).

4.2.2 The Scope Of Our Ontology

There is an increase in the availability of domain-specific knowledge in the Web such as financial news. Consequently, we will employ the financial information exploration and financial decision-making activities as use-case scenarios for the proposed semantic knowledge-based framework implementation. This scenario is about assisting the individual investors who would like to decide whether they invest in individual stocks or selling and reinvesting in other individual stocks. The application is for both investors case scenarios, investors who would like explore the sematic knowledgebase and then make the decision by themselves and investors who would like to acquire a decision-making assistance from the application. The semantic knowledgebase of this application is based on extracting information from unstructured data such as online news and then this semantic knowledgebase is 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. The concept map of our use-case scenarios activities is illustrated in Figure 4.2 below.

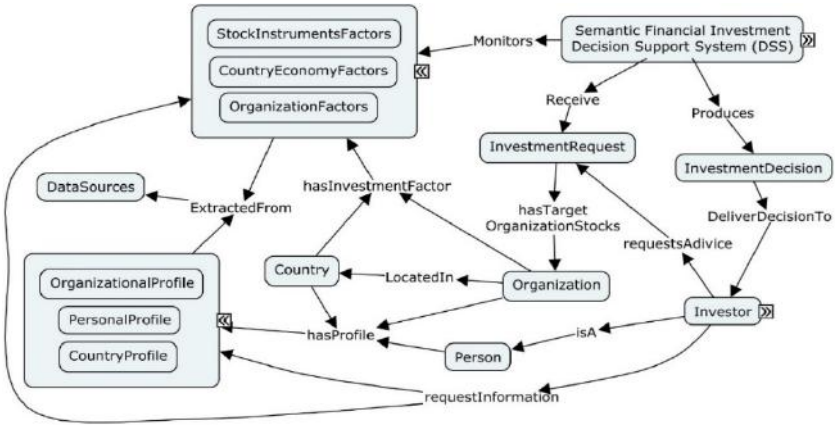

Figure 4.2: The Concept Map of the Activities of the motivating Scenario

Finance communities have access to massive volumes of unstructured, semi-structured and structured data from various sources. The unstructured data source of this research is online financial news articles. They are retrieved by using the Rich Site Summary (RSS) feeds including BBC, Reuters and Yahoo Finance. These online sources have their own content creators or they are authorised to source and redistribute news by partnering with other stock market news sources. We specifically retrieved documents from these online sources that are about stock market news. Table 4.1 presents some examples of those news RSS Feeds links.

Table 4.1: Examples of Online news RSS feeds of Unstructured data

RSS Link	News Source
http://rss.cnn.com/rss/money_markets.rss	CNN Money (CNNMoney 2018)
http://feeds.bbci.co.uk/news/business/rss.xml	BBC (BBC 2018)
http://feeds.reuters.com/reuters/UKPersonalFinanceNews	Reuters (Reuters 2018)
https://uk.finance.yahoo.com/news/provider-yahoofinance	Yahoo Finance (Yahoo 2018)

We applied information extraction techniques to retrieve information from these unstructured data sources to be constructed in a semantic knowledgebase.

The semi-structured and structured data sources, which are used to enrich the resultant semantic knowledgebase, are from diversity of sources such as the Linked Open Data cloud and semi-structured data sources such as API endpoints that provide access to different economic datasets. Table 4.2 presents the sources of the structured and semi-structured data that are used to enrich our semantic knowledgebase.

Table 4.2: The sources of the structured and semi-structured data

Dataset Access Source	Dataset Access Type
http://dbpedia.org/sparql (DBpedia Team 2015)	Linked Data endpoint
https://developers.google.com/freebase/ (Freebase Metaweb 2014)	Linked Data endpoint
http://worldbank.270a.info/sparql (Capadisli 2014)	Linked Data endpoint. Also, available as web service API (WorldBankData 2018)
https://www.crunchbase.com/#/home/index (Crunchbase 2018)	Web service API. This API is not free; as a result, I down loaded RDF data dump available in this link: https://datahub.io/dataset/linked-crunchbase. This linked data dump has been constructed by Färber, et al. in (Färber, Menne and Harth 2017).
http://finance.yahoo.com/lookup?s=API (YahooFinance 2018)	Web Service API

The semantic knowledgebase is constructed from the data sources mentioned above. The scope of ontology-based semantic knowledgebase should fulfil the activities of use-case motivating scenario. This ontology is used in semantic knowledge-based application to base decisions on reasoning about domain knowledge. Furthermore, we should make sure that the ontology-based semantic knowledgebase has enough information to deliver a required specific stock investment decision support. As a result, the ontology should describe the concepts and their interrelations (Noy and McGuinness 2001, Grimm, Hitzler and Abecker 2007).

The economy and finance domain is a conceptually rich domain in terms in varsity, volume and value. A massive amount of information is produced world-wide every day; however, its processing is very difficult and time consuming task. Thus, this domain knowledge should be represented in a form that can be processed by machines. Moreover, this domain contains enormous set of concepts and definitions to describe in an ontology thus limiting the scope the ontology is crucial. However, it is difficult to decide where the generality level of the ontology should stop and it is impossible to include all concepts and their interrelations in the target domain. As a result, we have proposed our targeted domain ontology for stock market investment information exploration and decision-making support. Considering the scope of our ontology and the motivating scenarios, the acquiring and analysing the domain knowledge has been done by the assessment of domain experts. The goal of this analysis is to capture concepts and their interrelations from unstructured, semi-structured and structured data sources such as people, organisations, locations, numbers, dates, addresses, stock symbols and stock indices. Table 4.3 presents examples of concepts and relations that captured from our target domain knowledge.

Table 4.3: Examples of concepts and relations that captured from our target domain knowledge

Entity	Concept Type	Relations for concept
Jeff Jacobson	Person, Investor	has Employer, has Location, has Stock, has Request
Xerox Technology	Organization	employer of, has Stock Symbol, share Increased By
United States	Location, Country	has City, has Economy Indicator

XRX	Stock Symbol	Issued By
NASDAQ	Stock Index	Index Increased By
December 2014	Date	profit Decrease Date

Table 4.4 below shows examples of concepts and their interrelation in the context of our domain-specific knowledge.

Table 4.4: Examples of Concepts and Their Interrelations in Our Domain-specific Knowledge

Data	Instances and Concept	Description
Douglas Flint has been the chairman of HSBC since the end of 2010	HSBC (Organization) is the employer of (Relation) Douglas Flint (Person)	A relation between two entities
Shares of Wells Fargo were down 2.3 percent Friday morning	Wells Fargo (Organization) Share is increased (Relation) By 2.3% (Percentage) On Friday (Date)	An event of an entity in a specific date
France has GDP rate of 2.5% in 2016	France (Location) Has GDP rate of (Relation) Of 2.5% (Percentage) In 2016 (Date)	A relation between two entities with a timestamp

The results of the domain-specific knowledge analysis in this task will be used in the next task, descripting the concept map of the domain-specific knowledge.

4.3 Descripting the Concept Map, Ontology Conceptualisation Task

4.3.1 Overview

Domain conceptualisation or building the domain's knowledge map aims to create a prearranged vocabulary and semantic structure for exchanging information about that domain. This task consists of building an intermediate conceptual model that can be in any form which is understood and accepted by domain experts such as concept map. Concept map can be viewed as an intermediary conceptual level representation above the implementation level of the ontology (Nagypál, Deswarte and Oosthoek 2005).

We utilised concept map technique because it has simple semantics that can act as concept captured mechanism from the knowledge to domain experts. Domain experts can convey their understanding of a domain in order to fulfil the criteria of knowledge identification, interaction, representation and sharing. The concept map is a simple graphical representation in which instances and classes are presented as nodes, and relationships between them are shown as arcs. We exploit this feature in order to perform the informal modelling stage of building an ontology (Castro, et al. 2006).

There are efforts to build a comprehensive taxonomy for financial domain; therefore, we considered the reuse of publically available ontologies when we built our ontology. One of the ontologies which are used as a start point for our semantic model is a finance ontology from Fadyart (Fadyart 2013). To construct the concept map and then the ontology, we

followed the approach that is presented in the work of Hegazy, et al., in (Hegazy, Sakre and Khater 2015). The main steps of this approach are:

1. Identify the concepts and their hierarchy.
2. Identify the concept disjoint.
3. Add the relationships between the concepts
4. Refine the concepts based on relationships they participate in.
5. Identify the definitions of concepts and relations.
6. Refine the ontology through various iterations of the above steps.

However, concepts in ontology can be identified by using three methods, top-down, bottom-up, and hybrid or middle-out. In the top-down method, the most abstract concepts are identified first, and then specialised into more specific concepts. In the bottom-up method, the most specific concepts are identified first and then generalised into more abstract concepts. On the other hand, the hybrid method is a combination of top-down and bottom-up methods. It starts with identifying the most important concepts which are more highly connected to other concepts; then, generalised and specialised into other concepts (Corcho, Fernández-López and Gómez-Pérez 2003, Du and Zhou 2012, Beck and Pinto 2002). We believe that the hybrid method is appropriate for developing our ontology because it is better to start by the most important concepts to be correctly and accurately defined. Furthermore, the concept map should be easily revised and improved by both the domain experts and knowledge engineers. It should have the ability to accommodate the unforeseen circumstance, where the process of accomplishing solution to a certain activity in the use-case motivated scenario is subject to change and propose plan and the concept map might change; thus, the ontology development should be able to accommodate the model reengineering.

4.3.2 Our Concept Map Implementation

We intend to model the domain knowledge of the motivation use-case scenario in terms of key concepts, their interrelations and the characteristics of the data as well as the interaction with the target beneficiary groups to be structured as a map of interrelated concepts. The process of modelling this domain knowledge should capture all entities that are related to the problem domain and the scope of the ontology. In the implementation of the concept map, we attempted to model the interaction of all beneficiary groups involved in our domain knowledge. The target concepts, including the knowledge users, were organised, explored and verified within the targeted problem domain to identify and understand how they have been used in the domain knowledge.

Practically, we started with identifying the main concepts, which are Company, Country and Person. Then, we identified the general and specific super- and sub- concepts accordingly to developing the concept hierarchy. The next step is defining the properties of concepts and relations between the concepts. For example, we identified the super-concepts

Organization for Company and Location for Country, and we identified the sub-concept Employee for Person concept. The relations between Organization members and Location members could be (Organization is located in a Location) and the relation between Organization members and Person members could be (Organization is an employer of Person). Then, defining the other concepts and linking them with appropriate properties to appropriate classes such as Stock and StockIndex concepts. After several iterations analysing and discussing the previous issues, we obtained the first version of the concept map design, which comprises the concepts and relations shown in Figure 4.3 below.

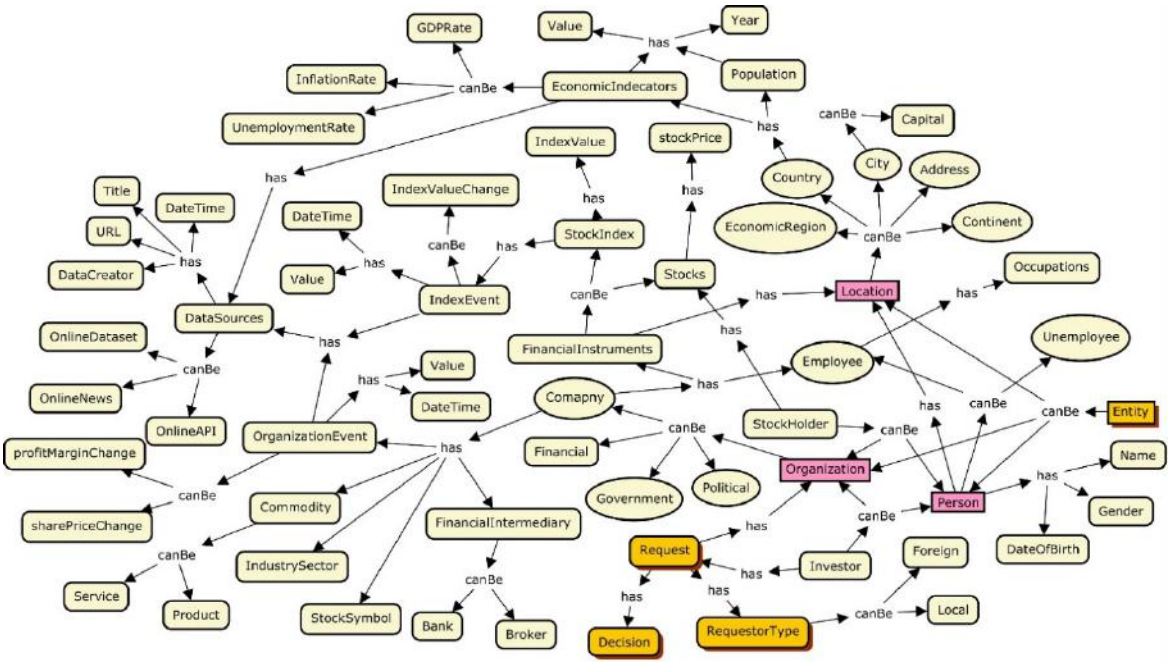

Figure 4.3: The Concept Map of Our targeted Domain-Specific Knowledge

After determining the knowledge sources and conceptualising the domain knowledge, the next steps are about deciding which technologies, languages and tools will be used in order to design the ontology.

4.4 Transforming the Conceptual Description into a Formal Model (Ontology), Formalisation Task

4.4.1 Overview

The Formalisation task is about knowledge representation by transforming the informal model, the Concept Map, into a formal model, ontology. This ontology represents the

domain knowledge in a manner that can be reasoned and interpreted by machines. To perform this task, we have utilised the Semantic Web Technologies.

Semantic Web offers a powerful logical and standardised technologies to represent, share and process knowledge such as inference and validation. One of the core components of Semantic Web is ontology. Ontology is a formal explicit description of the targeted domain knowledge and it plays a key role in Semantic Web knowledge representation. The formalisation of semantic knowledgebase by using ontology could include multiple axioms, definitions, rules, facts, statements, and any other primitives. The main components of the ontology are concepts, relations, instances and axioms. 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 languages include Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL). The specifications of these languages are standardised and recommended by The World Wide Web Consortium (W3C) (Ameen, Khan and Rani 2014a).

4.4.2 Formalising Our Problem Domain Knowledge

RDF triples model provides more suitable mechanisms for applying Semantic Web knowledge representation languages. In fact, we believe that there is a significant advantage in RDF model's semantic interoperability because of its triple structure below:

(Subject → Predicate → Object)

This structure provides natural semantic units representation and it can be mapped to extracted relations from our problem domain knowledge.

When several triples are connected together, they form an RDF graph. The nodes of this graph are resource with URIs or literal and its arcs are properties. Furthermore, RDF graphs support blank nodes, which represent anonymous resources. Though these blank nodes cause some issues in processing the knowledgebase such as the problems of merging different RDF graphs and linking a diversity of datasets (see subsection 1.3.1 in chapter 1 above). In fact, RDF is designed to capture knowledge and meta data that is spread over the web (Lord 2010, Taye 2010, Cao, et al. 2012, Henson 2013, Grimm, Hitzler and Abecker 2007).

For example, the ground fact (Apple shares is decreased by 5.86%) can be modelled by using the RDF triple model as presented in the triple below:

kbfwo:apple → kbfwo:shareDecreasedBy → "5.86%" ^^xsd:string

Where:

(kbfwo:) is the prefix of the namespace of our ontology, Knowledge-Based FrameWork Ontology.
In our ontology, the namespace is:
"http://localhost:8085/myOntology/myLastOntology.owl#"
(kbfwo:apple) is the URI identity name of subject resource "apple"
(kbfwo:shareDecreasedBy) is the URI identity name of the predicate resource "shareDecreasedBy"
("5.86%" ^^xsd:string) is the typed literal object where (xsd:string) is the String XML Schema Data type namespace.

RDFS is a general-purpose language for representing simple RDF vocabularies on the Web. It facilitates the specification of application-specific ontological vocabularies in form of class and property hierarchies on top of RDF resources. The resources that are represented in the RDF triples example above are illustrated by using RDFS vocabularies in the RDF graph below,

kbfwo:Organization → rdf:type → rdfs:Class
kbfwo:Company → rdf:type → rdfs:Class
kbfwo:Company → rdfs:subClassOf → kbfwo:Organization
kbfwo:apple → rdf:type → kbfwo:Company
kbfwo:shareDecreasedBy → rdf:type → rdfs:Property

The above graph, uses the RDFS vocabularies for describing the resources to allow the formulation of subsumption hierarchies and the distinction between instances and concepts in the ontological sense. However, there is no clear separation between classes and their instances in RDFS. Instead, it allows self-reference and classes being members of other (meta) classes. Any resource can be tagged as a class by relating it to the predefined meta type (rdfs:Class). The domain and range of the properties can be defined with the predefined predicates (rdfs:domain) and (rdfs:range) (Lord 2010, Taye 2010, Cao, et al. 2012, Henson 2013, Grimm, Hitzler and Abecker 2007). In the above example, we can set the domain range of the property (kbfwo:shareDecreasedBy) as follows:

kbfwo:shareDecreasedBy → rdfs:domain → kbfwo:Company
kbfwo:shareDecreasedBy → rdfs:range → xsd:^^string

That means, any resource that fills the subject position of an RDF triple with (kbfwo:shareDecreasedBy) property as predicate should be of type (kbfwo:Company) and any resource that fills the object position should be literal of type (xsd:string).

OWL provides an expressive language for defining ontologies that capture the semantics of domain knowledge. It is built on top of RDFS and adds a logical formalism to the language. Also, it provides additional vocabularies along with a formal semantics. For example, it allows the expressing of individuals equality (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). Also, expressing the inverse of object proberties (owl:inverseOf) (Polleres, et al. 2013, Roussey, et al. 2011, Tomai and Spanaki 2005).

For example, the ground fact (Microsoft is an employer of Bill Gates) can be modelled by using the RDF triple model as presented in the triple below:

kbfwo:Organization → rdf:type → owl:Class
kbfwo:Company → rdf:type → owl:Class
kbfwo:Company → rdfs:subClassOf → kbfwo:Organization
kbfwo:microsoft → rdf:type → kbfwo:Company
kbfwo:Person → rdf:type → owl:Class
kbfwo:Employee → rdf:type → owl:Class
kbfwo:Employee → rdfs:subClassOf → kbfwo:Person
kbfwo:billgates → rdf:type → kbfwo:Employee
kbfwo:employerOf → rdf:type → owl:ObjectPropery
kbfwo:employerOf → rdfs:domain → kbfwo:Company
kbfwo:employerOf → rdfs:range → kbfwo:Employee

kbfwo:microsoft → kbfwo:employerOf → kbfwo:billgates

If we add an inverse property (kbfwo:hasEmployee) to the property (kbfwo:employerOf),

kbfwo:hasEmployer → rdf:type → owl:ObjectPropery
kbfwo:employerOf → owl:inverseOf → kbfwo:hasEmployer

The following triples can be inferred,

kbfwo:hasEmployer → rdfs:domain → kbfwo:Employee
kbfwo:hasEmployer → rdfs:range → kbfwo:Company

kbfwo:billgates → kbfwo:hasEmployer → kbfwo:microsoft

Where (owl:inverseOf) and (owl:ObjectPropery) are OWL vocabularies.
OWL allows to describe complex classes. They can be described by using Boolean operators and restriction constructors. Each Boolean operator takes one or more classes as operands. These classes may be named classes, or may be complex classes formed from other constructors' or operators' descriptions. The examples of these Boolean operators are owl:unionOf, owl:intersectionOf and owl:complementOf. In addition, restriction constructors in OWL allow describing the individuals of restricted classes in terms of constraints on relationships that those individuals participate in using a specific relation properties with individuals in specific classes. They are value restrictions and cardinality restrictions. The value restrictions are existential (owl:someValuesFrom), universal (owl:allValuesFrom) and limited existential (owl:hasValue). The cardinality restrictions are maximum (owl:maxCardinality), minimum (owl:minCardinality) and exact (owl:cardinality)

cardinality restrictions (Polleres, et al. 2013, Roussey, et al. 2011, Tomai and Spanaki 2005).

For example, if we would like to classify individuals who are not employees; in other words, the individuals who are members of class (kbfwo:Person) but are not members of class (kbfwo:Employee) will be members of class (kbfwo:Unemployee). This classification axiom can be expressed by using the OWL Boolean operator (owl:complementOf) and written as follows:

kbfwo:Person → rdf:type → owl:Class
kbfwo:Employee → rdf:type → owl:Class
kbfwo:Unemployee → rdf:type → owl:Class
kbfwo:Employee → rdfs:subClassOf → kbfwo:Person
kbfwo:Unemployee → rdfs:subClassOf → kbfwo:Person
kbfwo:Employee → owl:disjointWith → kbfwo:Unemployee
kbfwo:Unemployee → owl:complementOf → kbfwo:Employee

An (owl:disiointWith) property between two classes states that any individual cannot be a member of these both classes in the same time. An (owl:complementOf) statement describes a class for which the class extension contains exactly those individuals that do not belong to the class extension of the class description that is the object of the statement.

For reasoning tasks, we applied and utilised the two types of Semantic Web reasoning, Ontology OWL reasoning and user-defined rule-based reasoning. Below is an examples of these types.

For OWL reasoning example, the relevant classes representing (kbfwo:Person), (kbfwo:Company), (kbfwo:Stock) and (kbfwo:StockHolder) and relevant object property representing (kbfwo:hasStock). Also, assuming the following triples are exist in the semantic knowledgebase,

kbfwo:kwakeb → rdf:type → kbfwo:Company
kbfwo:hadi → rdf:type → kbfwo:Person
kbfwo:shares001→ rdf:type → kbfwo:Stock
kbfwo:hadi → kbfwo:hasStock → kbfwo:shares001

If we would like to classify the stock holders in a specific class (kbfwo:StockHolder), we can apply the OWL existential restrictions. They 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 those are members of a specific class. If we would like to use the Description Logics to represent this restriction, it will be as in the formula that is represented by using Manchester syntax as below:

Class: D **EquivalentTo:** P **some** C

Where:

D is an named class to be equivalent to the unnamed restriction class

C is the named class that has the individuals which they will be used to define the individuals of the unnamed restriction class.

P is the property that is used to link between its right hand side individuals of C class and the individuals of the unnamed restriction class.

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

Class: kbfwo:StockHolder **EquivalentTo:** kbfwo:hasStock **some** kbfwo:Stock

The meaning of this restriction is that exactly those individuals will belong to the anonymous restricted class which have at least one (kbfwo:hasStock) property that is linked to an individual belonging to a given class description (kbfwo:Stock) on its right-hand side.

It is worth noting, also, that we use (owl:equivalentClass) to relate the restriction to the class being described, (kbfwo:StockHolder). The equivalent classes are sometimes referred to as a Necessary & Sufficient criteria. This is because the restriction specifies necessary and sufficient conditions for being a stock holder. Anyone who is a stock holder must own at least one company share, and anyone who has at least one company share is a Stock holder. In other words, not only are the conditions necessary for membership of the class (kbfwo:StockHolder), they are also sufficient to determine that any individual that satisfies them must be a member of the class (kbfwo:StockHolder).

Since (kbfwo:hadi) has as a (kbfwo:hasStock) relation with (kbfwo:shares001). By iterating over all the individuals in an OWL ontology, querying for subsets of named individuals with certain properties can be achieved. A reasoner would derive the following statement:

kbfwo:hadi → rdf:type → kbfwo:StockHolder

For User-defined rule-based reasoning, we adopted Jena rule format that is used only by reasoning engine in the Jena framework. The syntax of this rule language is based on RDF(S) and it uses the triple representation of RDF descriptions, which is almost like Notation3 (N3) except that a rule name can be specified in a rule. The built-in functions consist of many set of functions including production functions such as instance creation and instance removing, and can also be extended by the user. The Jena rules and OWL ontology are bound to rules reasoner engine separately, then, the rules are executed.

To make rule examples by using Jena rules, we create this scenario. Suppose we want to assist an stock investor for buying or holding stocks according to some information related to the targeted company exist in the semantic knowledgebase. These information includes the current stock price (Price) and the intrinsic value of the stock (Valuation). The investment

decision will be taken according to the fact that whether the stock is under valuated price 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 as below by using Jena rules syntax, the rules will be as follows:

```
[ruleName: (kbfwo:investorRequestID kbfwo:hasTargetedCompany ?TargetedCompany),
        (?TargetedCompany kbfwo:hasSharePrice ?Price),
        (?TargetedCompany  kbfwo:hasStockPriceValuationValue ?Valuation),
        lessThan(?Price, ?Valuation)
->
        (kbfwo:investorRequestID kbfwo:hasDecisionConclusion "Buy or hold the stock")]
```

In the previous subsections, we have presented details about specification, conceptualisation and formalisation tasks for modelling our targeted domain knowledge. Nevertheless, we believe that this domain knowledge is heavily represented by non-binary relations. We believe, also, that direct binary relations are not sufficient to represent and model our problem domain knowledge. Consequently, we have attended to adopt N-ary relation pattern to represent domain-specific non-binary relations in the resultant semantic knowledgebase. Representing N-ary relations in our research's ontology necessitates investigating the modelling requirements such as reasoning requirements (OWL constructors, Necessary and Sufficient Conditions Classifications, Property restrictions, Property Characterises, Rules, Query-specific reasoning using SPARQL) and how they will serve the end user requirements. In next section, we will present Non-binary relation investigation and modelling Implementation.

4.5 Non-Binary Relations Problem

4.5.1 Problem Overview

Semantic ontologies are constructed by using OWL language. The designers of OWL decided to be compatible with already existing standards, RDF and RDFS. These standers obey the universal RDF data object, the triple, as in the form of (4.1) below which is presented in Description Logics style.

$$Predicate(Subject, Object) \qquad (4.1)$$

The RDF triple model, in fact, 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. The ground facts in the triples below are examples which are extracted from our problem domain.

```
kbfwo:shareDecreasedBy(kbfwo:apple, "5.86"^^xsd:string)
kbfwo:shareDecreaseDate(kbfwo:apple, "Friday, 16/12/2016"^^xsd:datetime)
```

The two triples above are stating that stock prices of Apple company is increased by 5.86 percent on Friday, 12/12/2016. Assume that Apple has a share decrease by another value and in another date such as:

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 decreases. Also, it is hard to add more details about these facts such as the source of these facts or add details related to the information extraction technique used to extract them.

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

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

This Non-binary relation general form can be simplified by including only one subject, which is in most cases. That form will be as in the form (4.2) below:

$$predicate(subject, object_1, object_2, object_3, ... object_n) \quad (4.2)$$

The representation of N-ary relations in OWL ontologies is one of the design pattern issues that are investigated by researchers in the Semantic Web communities. Some of those researchers have investigated the extension of the OWL language features such as Krieger, et al. in (Krieger and Willms 2015) and Salguero, et al. in (Salguero, Delgado and Araque 2009). Other researchers have investigated the use of existing features of OWL language such as Sinha, et al. in (Sinha and Couderc 2012), Vinu, et al. in (Vinu, et al. 2014) and Hoekstra in (Hoekstra 2009). In addition, this issue is discussed by the Semantic Web Best Practices and Deployment (SWBP) working group in World Wide Web Consortium (W3C) (W3C 2018) and other research groups such as the Ontology Design Patterns (ODP) (ODP 2018), which deals with N-ary relations as one of ontology design patterns. In this research, we adopted the approach of using existing features of OWL language because there is no standard yet for extending the OWL language even though this standard should be supported by Ontology editing and reasoning tools.

As an ontology design pattern, there are two main solutions have been proposed to represent N-ary relation, statement centred (statement-as-class) or relation centred (relation-as-class) (Noy, et al. 2006, Aranguren, et al. 2008).

The first approach is called reification. It is the process of representing (subject-predicate-object) statement as a subject in other statements. Although 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. As a result, W3C is not supporting reification anymore (Noy, et al. 2006, Vinu, et al. 2014). According to Dodds, et al. in (Dodds and Davis 2012) suggestion, the reification technique is beneficial in the description of the changes in the structure of the RDF graphs; for example, the added or removed statements. It is because, as aforementioned, reification is about adding more information about statements rather than relations.

The second approach for representing non-binary relations is called relation-as-class. This 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 correspond 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 (Noy, et al. 2006, Hoekstra 2009). To illustrate this pattern, we return back to the pattern that is introduced in the form number (4.2) above. It can be represented in terms of the intermediate resources and the arguments of the relation as in the form number (4.3) below:

$$P(s, o_1, o_2, \ldots., o_n) \Rightarrow P(s,t) \times P_1(t, o_1) \times P_2(t, o_2) \times P_3(t, o_3) \times \ldots\ldots\ldots \times P_n(t, o_n) \quad (4.3)$$

Where:

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.

o_1, o_2,….. o_n : 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.

P_1, P_2, ….. P_n : the proprieties 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 4.4 below:

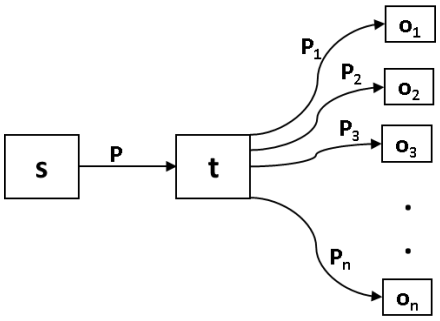

Figure 4.4: N-ary relation pattern

Also, we can express the form number (4.3) in terms of domain and range classes as in equation form number (4.4) below.

$$P(C, D_1, D_2, \dots., D_n) \Rightarrow P(C, RC) \times P_1(RC, D_1) \times P_2(RC, D_2) \times P_3(RC, D_3) \times \dots\dots. \times P_n(RC, D_n) \quad (4.4)$$

Where:

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

C : a domain class for P predicate.

P_1, P_2, ….. P_n : 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, P_1, P_2, ….. P_n .

D_1, D_2,….. D_n : the range classes for all properties of the binary relations, P_1, P_2, ….. P_n .

As Szeredi, et al. in (Szeredi, Lukácsy and Benkő 2014) argue, the limitation of that RDF can only define and represent binary relations does not pose a barrier to represent N-ary relations. In some special cases, RDF can provide a direct mean of assisting represent N-ary relations. For example, we can say a person has an address and the address has properties such as house number, street name, post code and city name. In this case, the address concept can be represented by a class and considered as an intermediate resource. In other words, the N-ary relations model, which are presented in form number (4.3) and Figure 4.4, is not new for RDF modelling, it is just about making it more general.

4.5.2 Our Approach to Implement an N-ary relation pattern

The above ground facts of Apple's shares increase could be formulated by using the simplified form number (4.2) above to be as in the 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, we could add the online news source document details of the ground fact. 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 and its licensing terms could encourage reusing the datasets. This metadata provides consumers of the data clarity about the provenance and relevance of a datasets (Heath and Bizer 2011). The details about the data sources could support the information in the triples; for example, the date which is stated in triple will be more clear 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.

We could express this N-ary relation ground fact 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.

We can model the N-ary relation ground fact above by using the pattern presented and explained in the form number (4.3) and

Figure 4.4 by transferring the N-ary relation ground fact into multi-binary ground facts. Firstly, we create a new Class (kbfwo:SharePriceChange) to represent N-ary relation's main predicate (kbfwo:sharePriceChange). Secondly, we create an individual of this association class, (kbfwo:sharedecrease_1). Then, we link N-ary relation subject with this individual. Lastly, we link the individual (kbfwo:sharechange_1) with the other properties' values that describe the N-ary relation. For example, the percentage value of share decrease, the date of decrease, the Machine Learning confidence value of the main relation 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:hasConfidenceValue (sharechange_1, "0.85745"^^xsd:float)
kbfwo:hasDataSourc (sharechange_1, kbfwo:158b_gone_the_apple)

Also, we could add information about the data source document 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 4.5 below.

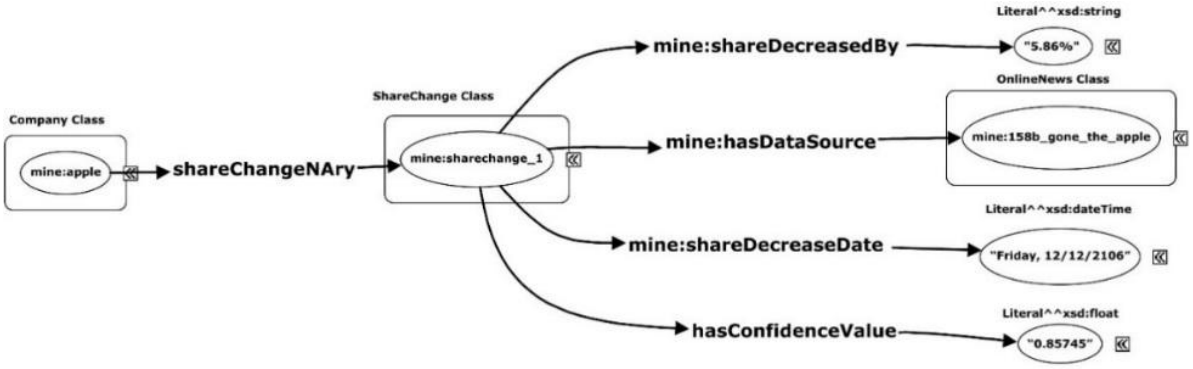

Figure 4.5: N-ary Relation Example

However, there are some considerations when introducing a new intermediate class for an N-ary relation. Firstly, we should give meaningful names to instances of properties 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 details.

4.5.2.1 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 that provides meaningful name for it and for its individual instances; also, to the main predicate of the N-ary relation. The individual members of the intermediate classes are required to serve as intermediate resources that link the subject to the objects of the N-ary relation (see form number (4.3) and Figure 4.4). In fact, it is common to use of blank nodes to represent these intermediate resources in the most of N-ary relations patterns representations.

However, we 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 (see subsection 1.3.1.1 in chapter 1 above). Secondly, all parts of the N-ary relations should be considered as one component and all the 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 4.5 above. This claim is in agreement with Krieger, et al. in (Krieger and Declerck 2015) opinion which 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 our implementation of N-ary relation pattern, we generated unique URI reference names for the individuals instances of the intermediate classes. These individuals instances are accessible through the other individuals properties of the N-ary relation.

There are two types of these intermediate classes, relation classes and event classes. In relation classes, the classes represent every relation property in the ontology of the domain knowledge. The members of these classes are URI individuals that represent every extracted relation instances in the domain knowledge. For example, the property (kbfwo:employerOf) has intermediate class (kbfwo:EmployerOfRelation) and has a main predicate or property for the N-ary relation (kbfwo:employerOfNAry).

To clarify this example more, we present the sentence example below:

“Jeff Jacobson, Xerox (XRX) corporate executive vice president and president of Xerox Technology, will speak at the conference next week.”

This sentence is retrieved from the online news document of title “Xerox’s President of Technology to Speak at Morgan Stanley Technology, Media & Telecom Conference.”

From this sentence, we can extract the following individuals:

kbfwo:jeff_jacobson → rdf:type → kbfwo:Employee
kbfwo:xerox_technology → rdf:type → kbfwo:Company
kbfwo:xeroxs_president_of_technology_to_speak_at_Conference → rdf:type →kbfwo:OnlineNews

However, we represent the fact that (Xerox Technology) is the employer of (Jeff Jacobson) in N-ary relation according to its data source documents as follows:

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

kbfwo:employerofnary_1234567 → rdf:type → kbfwo:EmployerOfRelation

Then, the binary triples that represent N-ary relation are linked to the intermediate individual member of the intermediate class as in the following 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 second type of intermediate classes is event classes. In event classes, the classes are used in N-ary relations representation as intermediate classes for event relations. The approach for this kind of N-ary relations is similar to the approach of relation classes except for the naming of the intermediate class and the main N-ary predicate. 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, we present the sentence example 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, we can extract the following individuals:

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

First, we create an intermediate individual member of the intermediate class (kbfwo:SharePriceChange). 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

4.5.2.2 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 inverse N-ary relation of the above N-ary relation by using the inverse properties is shown in the triples and Figure 4.6 below:

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

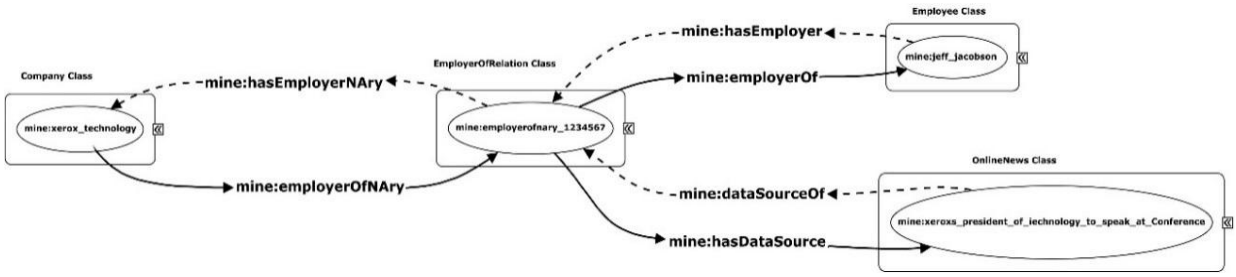

Figure 4.6: Inverse N-ary Relation Example

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

4.5.2.3 OWL axioms and Reasoning for N-ary Relations

Creating a class to represent an N-ary relation requires having local ranges or cardinality restrictions on some properties in the N-ary relation that depend on the class of some other properties. For instance, in the N-ary relation example presented in following 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 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 formula number (4.4) above. It is as shown in the relation from below.

$$\text{kbfwo:employerOfNAry}(\text{kbfwo:Company}, \text{kbfwo:Employee}, \text{kbfwo:OnlineNews}) \Rightarrow \text{kbfwo:employerOfNAry}(\text{kbfwo:Company}, \text{kbfwo:EmployerOfRelation}) \times \text{kbfwo:employerOf}(\text{kbfwo:EmployerOfRelation}, \text{kbfwo:Employee}) \times \text{kbfwo:hasDataSource}(\text{kbfwo:EmployerOfRelation}, \text{kbfwo:OnlinNews})$$

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, we specify a property (kbfwo:hasEmployerNAry) 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, we have defined both (kbfwo:employerOf) and (kbfwo:hasDatasource) 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 4.7 below.

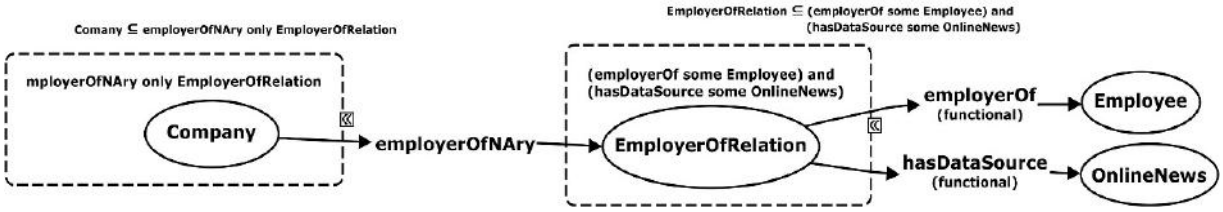

Figure 4.7: OWL axioms for N-ary Relations Example

When applying reasoning tasks, the intermediate classes and their individual members should be considered; for example, If we would like to classify the stock holders in a specific class, (kbfwo:StockHolder), we can apply the OWL existential restrictions. 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 we are using N-ary representation, represent this restriction by using Manchester syntax, it 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 P_1 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)

It is worth noting that there are two restriction classes, external and internal, and the external restriction class is the target to be equivalent to (kbfwo:StockHolder).

The meaning of this restriction is that exactly those individuals will belong to the anonymous internal restricted class which have at least one (kbfwo:hasStock) property that has an individual belonging to a given class description (kbfwo:Stock) on its right-hand side. In the meanwhile, exactly those individuals will belong to the anonymous external restricted class which have at least one (kbfwo:hasStockNAry) property that has an individual belonging to the anonymous internal restricted class on its right-hand side.

Since (kbfwo:hadi) has as a (kbfwo:hasStockNAry) with (kbfwo:hasstockrelation001) and (kbfwo:hasstockrelation001) has a (kbfwo:hasStock) relation with (kbfwo:shares001). By iterating over all the individuals in an OWL ontology, querying for subsets of named individuals with certain properties can be achieved. A reasoner would derive the following statement:

kbfwo:hadi → rdf:type → kbfwo:StockHolder

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, we will use the same example scenario in the previous subsection 4.4.2. This example is about supporting a stock investor for buying or holding stocks 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 our 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 is represented in our 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.

4.5.3 Discussion

We have adopted N-ary relation as relation centred or relation-as-class pattern as a N-ary relation pattern to represent domain-specific non-binary relations in our problem domain because the direct binary relations are not sufficient to represent them. 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.

We have investigated the N-ary relation patterns considerations when introducing a new intermediate class for a relation. Firstly, we should give meaningful names 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, we should define inverse properties for all properties involved in the N-art relation. Lastly, we consider the intermediate resources 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, our 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, we generated unique URI reference names for the individuals 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.

Our finding revealed that the N-ary relation pattern is a very important for the non-binary relations in a variety of domains. Whilst much can be modelled with binary relationships, there is a wide need for relationships of higher arity. Moreover, the existing Semantic Web Technologies and languages can be employed to represent the suggested N-ary relation pattern after taking the above considerations into account. We can argue that in some special cases, RDF can provide a direct mean of assisting represent N-ary relations. For example, we can say a person has an address and the address has properties such as house number, street name, post code and city name. In this case, the address concept can be represented by a class and considered as an intermediate resource. We can conclude though that the limitation of that RDF can only define and represent binary relations does not pose a barrier to represent N-ary relations. We believe that the N-ary relations pattern is not new for RDF modelling, it is just about making it more general.

In fact, 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, we can easily have relations crossing items, customers, dates, and regions. We may want to extract maximal associations between such attributes for business decision-making. Another typical application domain concerns the numerous situations where object properties can be recorded as features for a collection of objects over time. This typically provides kind of N-ary relations.

4.6 Implementing the Formalised Model, Implementation Task

4.6.1 Overview

For the process of designing, developing, editing and modifying the ontology, several implementation tools have been developed. Not only they are used to provide support to the development process of the ontology, but also they provide support to ontology conceptualisation. They are utilised to transform the concept map into a formal semantic model by supporting the Semantic Web ontology languages, RDF, RDFS, OWL (Isiaq and Osman 2012, Lloret, Gutiérrez and Gómez 2015).

According to Isiaq, et al., in (Isiaq and Osman 2012), most of these tools are beneficial and suitable development tools. However, their environments absolutely depend on the criteria for fulfilling the proposed applications objectives. Some identified criteria are crucial to semantic application development include interoperability and adaptability by the ontology standard languages, inference mechanism, tools architecture enhancement, developmental methodology support and usability.

The ontology development tools should offer a multiple inference mechanism with varying level of reasoning and features such as automatic classification, constraint, consistency check and exception handling. Because Semantic Web Ontologies adopt OWL as an implementation Language and it is an expressive knowledge representation language, the ontology implementation tools should support the OWL language. These tools should be used for querying OWL ontologies with respect to inferred knowledge or for verifying their consistency (Isiaq and Osman 2012, Grimm, Hitzler and Abecker 2007).

Several studies compared between existing ontology implementation tools; for example, the work of Kapoor, et al. in (Kapoor and Sharma 2010), the work of Alatrish in (Alatrish 2013), and work of Khondoker, et al. in (Khondoker and Mueller 2010). After careful consideration of major development tools, we decided to employ Protégé tool (Protege 2018) for the purpose of developing the ontology of this research. Protégé is an open-source platform developed at Stanford Medical Informatics. The Protégé model is used to represent ontology elements as classes, properties, property's characteristics, axioms, constraints, restrictions, and instances or individuals. The tool also facilitates consistency checks in order to maintain ontology correctness and output consistency at the point of development (Knublauch, et al. 2004, Isiaq and Osman 2012, Dombeu and Huisman 2011). Protégé has been employed by several researchers in their works as the main ontology implementation tool; for example, the work of Ameen, et al. in (Ameen, Khan and Rani 2012), the work of Lloret, et al. in (Lloret, Gutiérrez and Gómez 2015), the work of Yoo, et al. in (Yoo and No 2014), and the work of Taha, et al. in (Osman, et al. 2014). We were encouraged by these works to employ Protégé as the main ontology implementation tool in building and editing our research's ontology.

Our framework is implemented on top of Jena framework. Jena is a Java-based open-source application framework for developing Semantic Web applications. It provides collections of development tools; for example, RDF data processing libraries, RDF data store system which is Triple Database (TDB), Jena SPARQL query engine and its own rule-based inference engine. The framework has a number of predefined reasoning engines. These engines are utilised to support Semantic Web language such as RDFS and OWL and user-defined rules reasoning. The user-defined rules are implemented by the generic rule reasoner. It is mechanism is designed to be more general that can be used for many RDF processing or transformation tasks (Ameen, Khan and Rani 2014b). Rattanasawad, et

al. in (Rattanasawad, et al. 2014) conducted a comparison study to provide a guideline for researchers and developers in choosing rule-based reasoning engines that fulfil their researches' requirements. They review and compare between several rule-based reasoners including Jena reasoning engine. The comparison is established according to these criteria, RDFS/OWL reasoning, reasoning algorithms, rule languages and functions, and supported programming languages. Their results show that Jena reasoning engines are sufficient for our research requirements.

Next section will show the details of our ontology building.

4.6.2 Our Ontology Implementation

Based on the results of ontology specification, conceptualisation and formalisation tasks, the ontology has been built, developed, implemented and encoded in OWL by using Protégé tool. We first identified two roles of our ontology, to express the knowledge closely related to stock investment and to support investors in stock investment decision-making process by intelligently explorer the resultant semantic knowledgebase. This requires defining the main concepts of the ontology including concepts and properties representing the metadata for our targeting domain knowledge and motivation use-case scenario.

In addition, we followed the ontology design principles for defining naming conventions for concepts, properties and instances that recommended and described by Nagypál in (Nagypál 2005) and Noy, et al. in (Noy and McGuinness 2001). Defining naming conventions in an ontology makes the ontology easier to understand, and also it assists avoiding some common modelling mistakes. The recommended design principles can be summarised as,

- It is a common convention to begin concept names with capital letters such as Organisations, Person, Location.
- It is a common convention to begin instance and property names with non-capital letters such as the properties, employerOf and hasStockSymbol; and the instances nasdaq and jeff_jacobson.
- It is common conventions to write names concept or properties in Camel Case when they contain more than one word, except the first letter in properties; for example, the concepts StockSymbol and StockIndex and the properties hasSharePrice and hasIndexValue.
- It is a common convention to use the singular form in concept names such as Organization instead of Organizations.
- It is a common convention to use a proper prefix and suffix for property names and their invers property names. For example, using "has" as prefixes for property name

and “Of” as suffix for the invers property names. For example, employerOf property and hasEmployer invers property.

- It is recommended to provide meaningful names to the intermediate classes and their individual instances and to the main predicate of the N-ary relation.

Furthermore, when modelling a domain, developers need to decide whether to model a specific entity as a property value or as a set of classes; for example, the economic indicators of countries, GDP, unemployment and inflation rates. Is it preferred to simply create a country class and fill in different values for the properties of the economic indicators or to create a class for every indicator. In addition, the developers need to decide whether to model a specific entity as a class or as an instance in an ontology; for example, the occupations of the employees, product manager, advisor and others. Is it preferred to simply create an occupation class and make all types of occupations as an instance or create a class for every occupation. The choice between class and property or between class and instance depend on the applications of the ontology (Noy and McGuinness 2001, Nagypál 2005).

Because our targeted domain-specific knowledge is heavily represented by non-binary relations, we have adopted N-ary relation pattern to represent these relation in the domain-specific ontology model; thus, the resultant semantic knowledgebase is relation oriented modelled. This requires the distinction between the relations in all the extracted and enriched information. The intermediate classes of the N-ary relation model make the choice between class and property or between class and instance straight forward. For example, in the case of the economic indicators of countries, GDP, unemployment and inflation rates, there is an intermediate class for every rate, (kbfwo:HasGDPRelation), (kbfwo:HasUnemploymentRateRelation) and (kbfwo:HasInflationRateRelation). Also, there are main predicates for every N-ary relation, (kbfwo:hasGDPRateNAry), (kbfwo:hasUnemploymentRateNAry) and (kbfwo:hasInflationRateNAry). As explained in the section 4.5, the N-ary relation of these rates will be as explained in Figure 4.8 below.

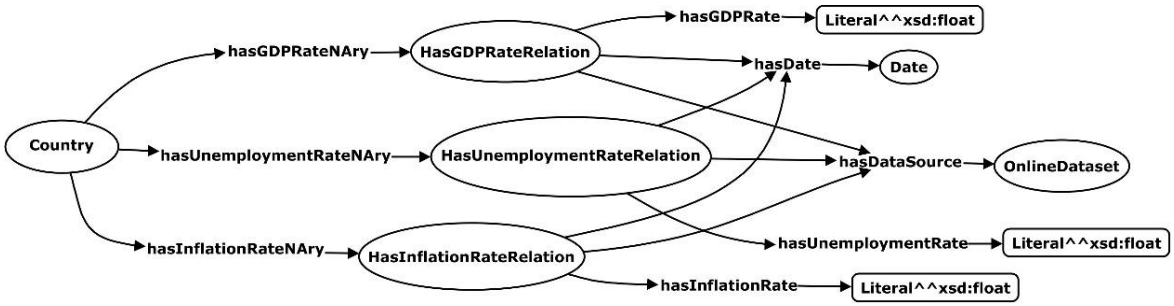

Figure 4.8: Example of GDP Classes and Relations

As shown in the figure, we can add any extra information to the N-ary relation such as the date of the rates and the data source, etc.

However, implementing knowledgebase in OWL requires expressing the extracted information, named entities and their interrelation, as classes, properties, and axioms. The classes are selected depending on the problem domain knowledge, which is financial and economic. The number of classes in our ontology, until writing this thesis, is around 100 classes and the number of properties is around 130 properties.

The classes can be categorised into five types, business entity, N-ary relation, business function, general and system. Table 4.5 and Figure 4.9 below show examples of these classes.

Table 4.5: Categories and Examples of Classes

#	Class Category		Examples
1	Business Entity		Organization, FinancialInstrument
2	N-ary Relation	Relation	HasSharePriceRelation, EmployerOfRelation
		Event	SharePriceChange, IndexValueChange, ProfitMarginChange
3	Business Function		Products, Services
4	General		Person, Location, Date, DataSorces
5	System Function		Request, Decision, Configuration

There are two types of properties, object properties and data type properties; however, the object properties can be categorised into normal properties and N-ary properties. Table 4.6 and Figure 4.9 below show examples of these properties.

Table 4.6: Categories and Examples of Properties

#	Property Category		Examples
2	Object Property	N-ary	hasSharePriceNAry, sharePriceChange
		Normal	hasStockSymbol, employerOf
3	Data Type Properties		hasSharePrice, hasGDPRate

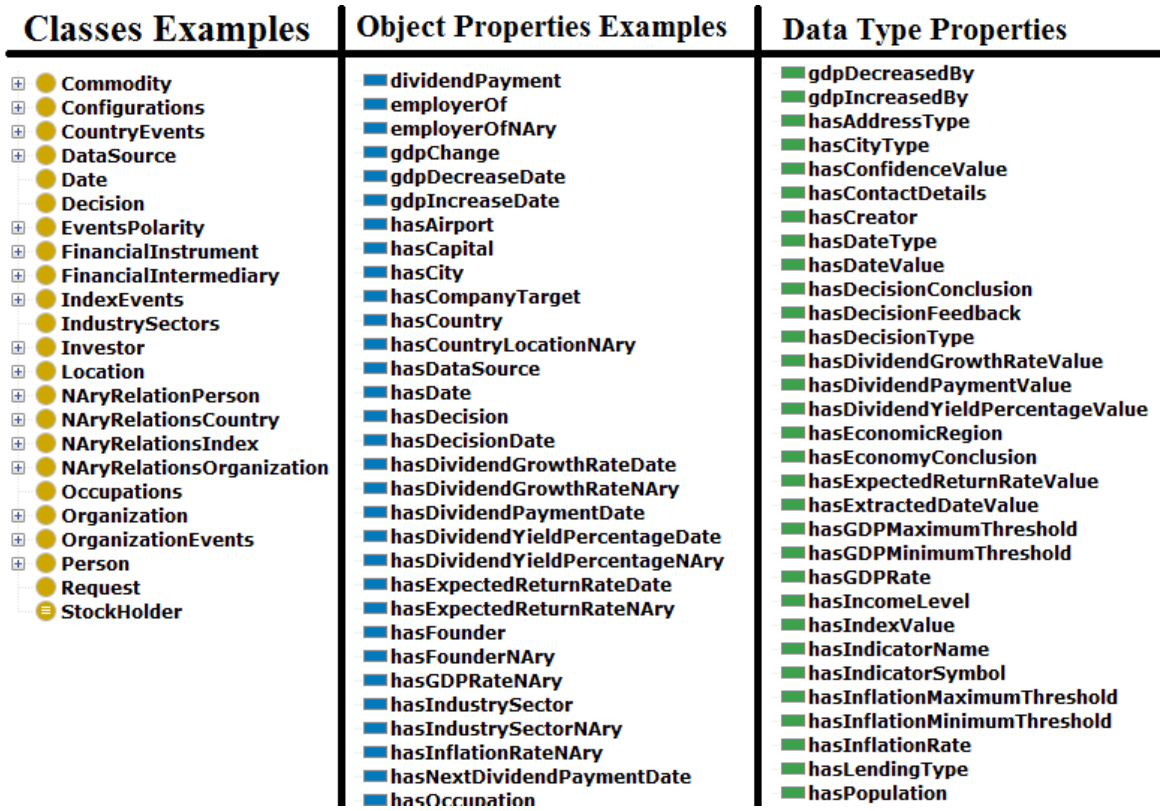

Figure 4.9: Classes and Properties Examples

Figure 4.10 below presents the whole graph of the ontology classes.

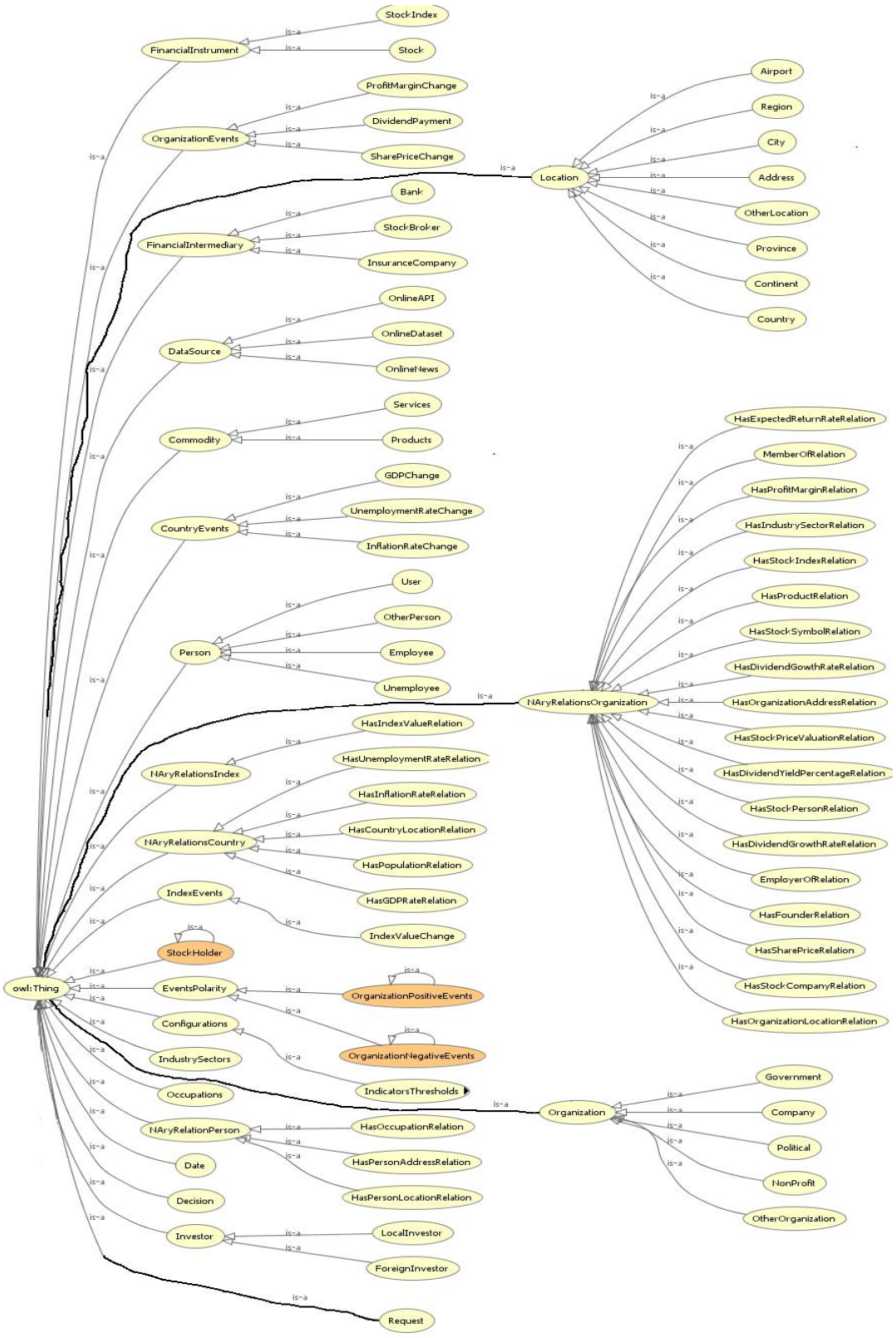

Figure 4.10: Ontology Graph

4.6.3 Ontology Maintenance

Ontologies building is an iterative task, which means that their concepts, relations and axioms are improved, extended or enriched to make ontologies more precise to the growth of the domain knowledge. For example, the information in the economic and financial domain is seems to be ever expanding; hence, the resultant knowledgebase require to be updated regularly. Expanding and updating the information in the semantic knowledgebase require updating and maintaining the semantic model, ontology, because new concepts may arise about a specific domain and they need to be considered in order to capture relative domain knowledge instances. Ontology maintenance is an important task in ontology developing and building processes. It includes adding new elements or updating, refining, merging, and removing existing elements. The elements could be classes, properties or axioms and the operations are considered under ontology maintenance process (Amardeilh, et al. 2013).

However, these processes activities require to be adequately documented for future improvement and maintenance of the developed ontology. Also, the documentation task assists tracing the reason of undertaking some certain modelling decisions for a later stage of the development process or problem solving. Depending on the propose of building the ontology, the documentation might require exclusively focusing on documenting the ontology developing process while others might require focusing on the decisions process that are undertaken for ontology developing (Davies, Studer and Warren 2006, Kapoor and Sharma 2010, Beck and Pinto 2002).

Not only Protégé is utilised to develop the ontology in this research, but also it is utilised to document the ontology development activities. In fact, OWL has several pre-defined annotation properties that can be used to annotate classes, properties, individuals and the ontology itself with various details as meta-data. These details may take the form of auditing or editorial information; for example, comments, creation dates, versions and authors. Also, these details may be referenced to resources such as web pages or other ontologies. These are some annotation properties that can be used to insert details about ontology developing and building processes activities:,

- Documenting the ontology or resources versions and compatibilities information by using the annotation properties (owl:versionInfo), (owl:priorVersion), (owl:incompatibleWith) and (owl:backwardsCompatibleWith),
- Adding meaningful or human readable names to classes, properties and individuals by using the annotation property (rdfs:label),
- Defining a related resources in other ontology by using the annotation property (rdfs:seeAlso)

- Storing the comments for the undertaken decisions in adding new resources or updating, refining, merging, and removing existing resources by using the annotation property (rdfs:comment).

We believe that the documented details in the annotation properties will support tracing the reason of undertaking some certain modelling decisions for a later stage of the development process or problem solving. Figure 4.11 shows a Protégé screen shot of an example of annotation properties usage.

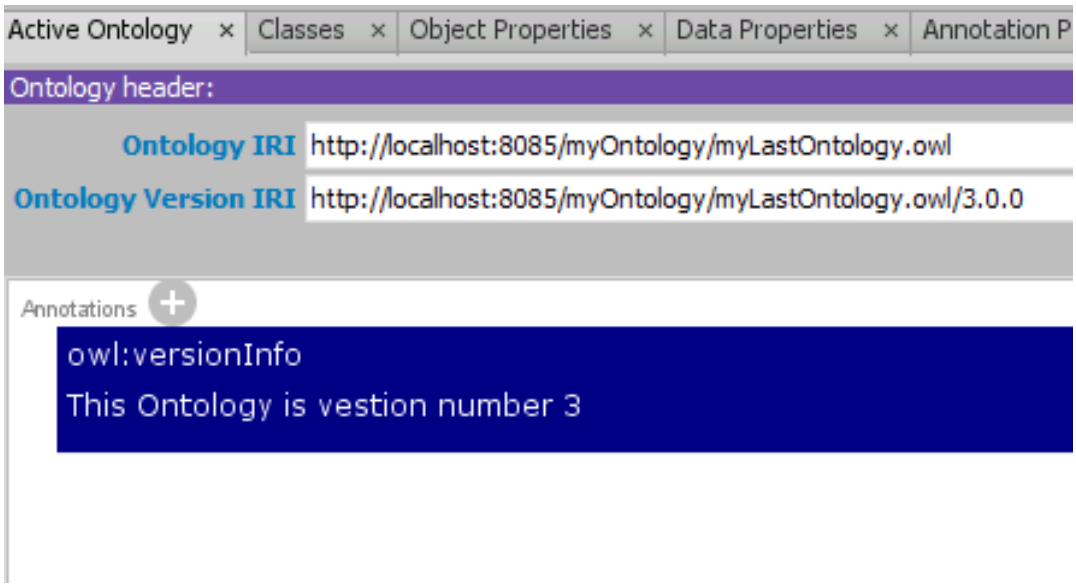

Figure 4.11: An Example Of Annotation Properties Usage

4.7 Summary

In this chapter, we described how Semantic Web Technologies is utilised to model our targeted domain knowledge. Semantic model or ontology describes and combines the corresponding relation between the concepts' instances from different sources and infer new information about these concepts in different contexts and enables the sharing and reusing of domain knowledge. Ontology building is a process that composed of a serious of stages which are, specification, conceptualisation, formalisation and implementation.

Specification task is about identifying the purpose and scope of the ontology. This research work uses the financial information exploration and financial decision-making activities as use-case scenarios for the proposed semantic knowledge-based application framework.

Conceptualisation is about describing the conceptual model for the ontology to meet its specification. This task consists of building an intermediate conceptual model or concept map. We modelled the domain knowledge of the motivation use-case scenario in terms of key concepts, their interrelations and the characteristics of the data as well as the interaction with the target beneficiary groups.

The Formalisation task is about knowledge representation by transforming the Concept Map into a formal model, Ontology. This ontology represents the domain knowledge in a manner that can be reasoned and interpreted by machines. To perform this task, we have utilised the Semantic Web languages to represent, share and process knowledge such as inference and validation. The OWL provides an expressive language for defining ontologies that capture the semantics of domain knowledge. It can be used to describe complex classes and infer new information by using OWL reasoning. However, our targeted domain is heavily represented by non-binary relations and the direct binary relations are not sufficient to represent and model it. Consequently, we have adopted N-ary relation pattern to represent these domain-specific non-binary relations.

Implementation is about implementing the formalised model in an ontology. The ontology has been implemented and encoded in OWL by using Protégé tool. Also, our proposed framework is implemented on top of Jena framework; specifically, for user-defined rule-based reasoning engines. Ontologies building is an iterative task, which means that their concepts, relations and axioms are improved, extended or enriched to make ontologies more precise to the growth of the domain knowledge.

The following chapters will describe Information Extraction from unstructured data approach which is adopted in this research. The extracted information will be constructed in a semantic knowledgebase by using the semantic model which is developed in this chapter.

5 Linguistic Pre-Processing and Named Entity Recognition for Information Extraction

5.1 Introduction

As mentioned in chapter 3 and chapter 4, we adopt a knowledge-based approach to implement the proposed framework. The advantage of this approach is that the targeted domain knowledge is analysed to understand its syntactic and semantic characteristics to be used to model the domain knowledge and then the knowledgebase is constructed by using a formalised semantics which facilitate the accessibility to similar datasets to improve the process of the knowledgebase population and enriching. The population step consists of extracting information from online unstructured data then relating that information to domain concepts and properties in the semantic modelling, ontology.

Extracting information from unstructured data requires applying automatic Information Extraction techniques in order to obtain valuable information from natural language texts. There is an opportunity in adopting knowledge-based approach because understanding the syntactic and semantic characteristics of the domain can aid Information Extraction process. Information Extraction could be considered as a pipeline process. In each stage of that pipeline, the tasks of Natural Language Processing are applied on the natural language texts. Information Extraction process usually starts in recognising the named entities; then, identifying identity relation between named entities, which is co-references resolution. Lastly, extracting the relation between the named entities in a certain event (Cunningham 2005, Farmakiotou, et al. 2000).

In Named Entity Recognition task, the sentence's atomic elements (words or entities) are addressed and classified into predefined types of named entities, such as organizations, place names, persons, dates and numbers. By applying Named Entity Recognition task, additional descriptive information can be extracted from the text about the detected entities such as the title and gender of persons. In entities' co-reference resolution task, the expressions in a document that refer to the same entity are identified. The co-reference relation will be marked between elements. For example, the name of a particular organization or person can be mentioned in the document in different expressions such as "Apple Co.", "Apple" or "it" and "John", "John Brown" or "he" (Piskorski and Yangarber 2013, Karkaletsis, et al. 2011).

Recognising named entities is a central task for processing the natural language texts because of two reasons. The first reason is that the named entities can be used directly in many applied research domains such as in medicine domain. The second reason is that the

recognising the named entities can be considered as a pre-processing step for other advanced natural language processing tasks such as extracting the relation between the named entities (Atdağ and Labatut 2013). In this research, we divided the Information Extraction pipeline into two parts. The first part is about the linguistic pre-processing of sourced unstructured data and recognising Named Entities. The second part is about Relation Extraction. In fact, the first part, also, is responsible on generating linguistic features for all words in all documents to be used to extract relations between the recognised named entities. Next section will present the types of linguistic pre-processing and Named Entity Recognition tools.

5.2 Linguistic Pre-Processing and Recognising Named Entities tools

Several tools has been developed to recognise the named entities. There are many factors should be considered to choose between these tools; for example, the capability to be adapted for a new domain, the data input and output formats and the level of performance and accuracy. However, the accuracy of these tools can vary depending on the considered type of entity, class of text and the complexity of the targeted domain (Atdağ and Labatut 2013). These tools adopt different approaches, some of them adopt the rule-based approaches that rely on hand crafted grammars rules and gazetteer lists such as ANNIE GATE, other tools adopt Machine Leaning techniques that rely on automatic training approach such as Stanford Named Entity Recogniser and many other tools combine the two previous approaches.

Stanford Named Entity Recognition tool is a Java implementation of recognising the Named Entity in unstructured texts based on Machine Learning techniques. It is also known as CRF Classifier because it provides a general implementation of linear chain Conditional Random Field (CRF) sequence models. These models are for 3 classes (Person, Organization, Location) named entity recognisers in English language. Also, the tool package provides various other models for different languages. To create new models, there are a well-engineered feature extractors and many options for defining these feature extractors to retrain the Named Entity classifier for other named entities (Finkel, Grenager and Manning 2005).

A Nearly-New Information Extraction system (ANNIE) is a built-in application in GATE tool (see subsection 2.3 above). It is a pipeline for Natural Language Process and Named Entity Recognition tasks. ANNIE pipeline is formed by the following components, Document Reset, Tokeniser, Gazetteer, Sentence Splitter, POS Tagger, Semantic Tagger and Orthographic Co-reference or OrthoMatcher. All ANNIE components communicate exclusively via GATE's Language Resources (documents or corpora of documents) and Process Resources. The Named Entity Recognition component uses rule-based technique

that rely on JAPE (Java Annotation Patterns Engine) language and gazetteer lists to recognise regular expressions in annotations on documents. (Cunningham, Maynard and Bontcheva 2014).

In this research, we decided to use an adapted pipeline based on the Rule-Based ANNIE tool in GATE NLP platform to apply Natural Language Processing tasks including the Named Entity Recognition. The performance of ANNIE pipeline had been tested in a massive number of online news documented and they have found that it is suitable for this domain. For example, it was adapted to recognise named entities in the work of Ruiz-Martínez, et al. in (Ruiz-Martínez, Valencia-García and García-Sánchez 2012) to analysis sentiment polarity in financial news domain. Additionally, it has been successfully adopted to implement a diversity of Information Extraction application domains with acceptable results. For example, this tool has been adapted to recognise named entities in the work of Faria, et al. in (Faria, Girardi and Novais 2012) to extract information from the Tourism and Legal domains.

5.3 Domain-Specific Information Extraction

There are various factors that influence the performance of the Information Extraction systems. For example, the information items which are extracted by systems can vary in complexity and in specificity. The complexity of the information to be extracted can vary from simple people names to complex events that involve multiple participants. On the other hand, the specificity of the information to be extracted can vary from covering a general domain or more than one domain to a specific domain. Nevertheless, some domains produce documents that use uncommon terms, phrases or syntax. Also, some terms do not have a universal meaning because terms in a document in particular domain might have different meaning in another domain. For example, “Apple” means a company name in an economics and finance domain and a fruit name in food and agriculture domains. The systems which process general information are different from systems which are process specific information. In fact, Information Extraction systems should balance between complexity and specificity, the more complex the knowledge to be extracted, the more specific must be the domain knowledge. The specific knowledge services require Information Extraction techniques to be able to search and extract specific knowledge directly from unstructured text. The specificity of information Extraction process tasks is influenced by text type and domain type. Text type is about the kinds of texts which are processed; for example, online news articles, email messages, companies’ reports and the output of a speech recogniser. Domain type is about the broad subject-matter of those texts; for example, financial news or sport news or technical support information or tourist information (Cunningham 2005).

There is a considerable proportion of unstructured data sources exclusively service specific domains. For instance, there are specific online documents that users interested in politics

or stock exchange news. In other words, not only the domains of interest are specific, but the sources of data also exclusively service that particular domain. Hence, domain specific knowledge offers an opportunity for improving the accuracy of Information Extraction tasks that retrieve the information from corpora of documents for the benefit of end user.

It can be argued therefore that these specific knowledge services should be guided by the domain knowledge. The domain Knowledge should detail what type of knowledge is to be obtained and for which exploration scenario. This scenario should make the IE techniques mediate between the domain text type and the requirements of various types of users. In these cases, domain-specific Information Extraction processing is often required in order to extract useful or interesting information.

The next subsections present the tasks of retrieving the online unstructured data and the pipeline of recognising the named entities.

5.4 Retrieving Online Unstructured data and Textual Content Detection

In the proposed framework phases, the information in the semantic knowledgebase is initiated by extracting information from unstructured online news. This information is constructed in a structured format to be easily explored and understood by machines then it is presented to end users. The unstructured online news contain specific online documents to users interested in stock exchange news. In other words, not only the domains of interest are specific, but the sources of data also exclusively service that particular domain.

In this research, the Information Extraction tasks are applied to domain-specific unstructured documents that are collected from of online economic and finance news; specifically, they retrieved from those online sources that are about stock market news. They are retrieved by using the Rich Site Summary (RSS) feeds. Examples of these RSS feeds are given in Table 4.1 of subsection 4.2.

The Information Extraction tasks annotate the documents with domin-relevant metadata tags. The structured information can be then exploited by knowledge-based applications and presented to users (Costantino, et al. 1997). The users can use this information as a negative or positive indicators to proceed in; for example, stock investment decision-making process.

Nonetheless, the online news Web pages consist of navigational elements, templates, and advertisements in addition to the actual news contents. These boilerplate texts may reduce the Information Extraction quality. To detect the news contents and remove undesirable texts, we employed an open source Java API library “boilerpipe” (boilerpipe 2014). This API library provides algorithms to detect and remove the boilerplate and templates around the main textual content of a web page (Kohlschütter, Fankhauser and Nejdl 2010).

Figure 5.1 below shows an example of an online news website and its news content.

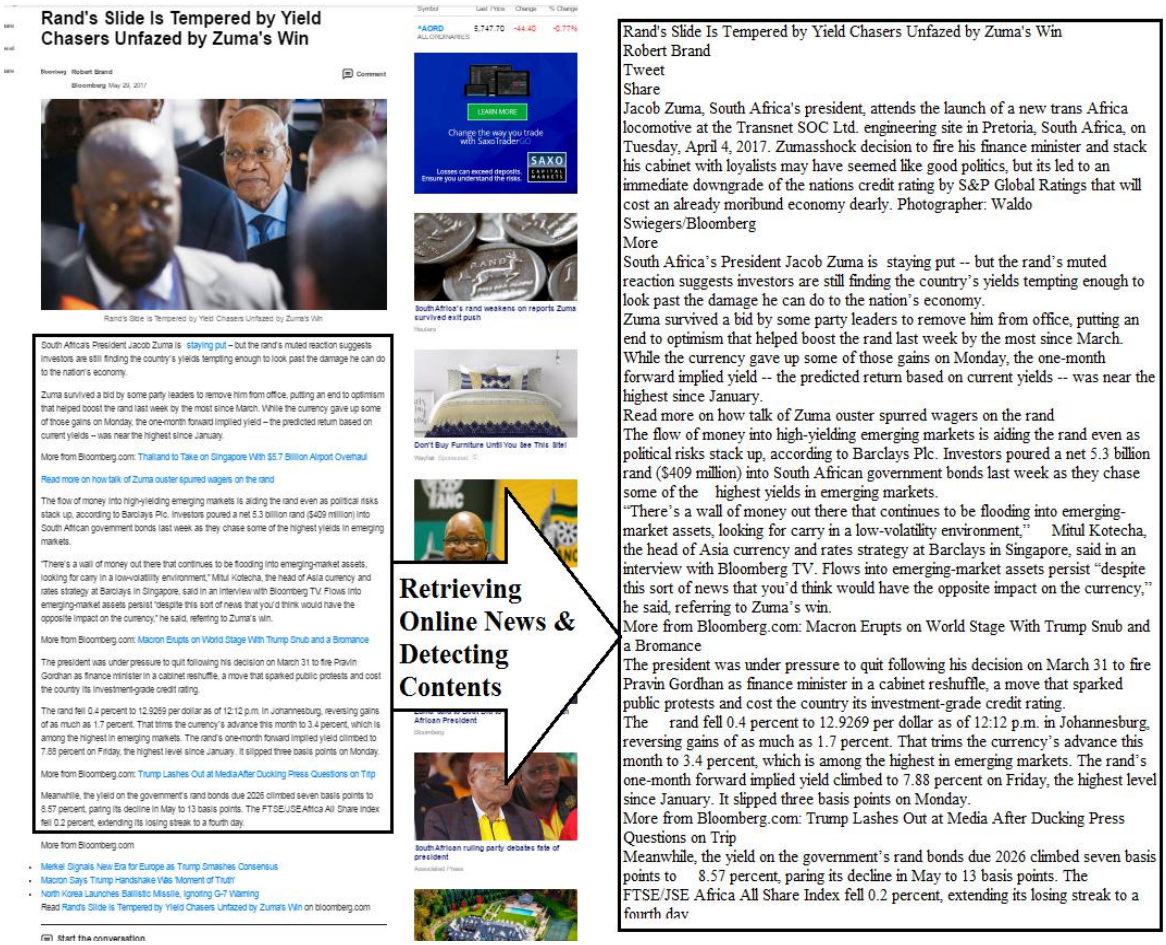

Figure 5.1: The content detection of the online news article

For the purpose of generating training datasets to create the relation classification models, we retrieved and detected the textual contents of more than 18 thousands documents from the online news RSS feeds. These clean plain text news documents are ready to apply the Information Extraction pipeline process tasks.

5.5 Natural Language Pre-Processing Tasks in the Named Entity Recognition Pipeline

An automatic analysis of the linguistic structure of a natural language textual documents is required to support Information Extraction thus constructing a semantic knowledgebase. The automatic linguistic analysis or natural language pre-processing is required not only to recognise the linguistic components in text such as words and sentences, but also to generate linguistic features of those components such as the part of speech type of those

words. These components and their features are necessary to extract the relevant contents of the natural language texts. In addition, they are required for further phases of Information Extraction such as annotating the named entities and the relation between them. The linguistic pre-processing tasks is called Natural Language Processing (NLP) tasks. They include tokenising, sentence splitter, gazetteer lists tagging, Part Of Speech (POS) tagging, morphological analyser, co-references resolution and dependency path tree tagging. The results of these tasks are linguistic features. These features will be used for recognising the named entities, relation instances detection and features generation for ML relation classification. However, the quality of Information Extraction results crucially depends on those tasks. Each task of NLP should be as reliable and precise as possible because errors could be cascaded since the earliest stage hence degrade the overall results (Benetka, Balog and Nørvåg 2017).

Named Entity Recognition can be defined as the identification of the entities which are mentioned in the text. A named entity is an expression or phrase in the text which represents an entity in real world. They could be proper nouns or identification numbers. Recognising named entities in unstructured data is a key feature in Information Extraction systems because it produces a valuable information about the targeted text of extraction. Mapping identified named entities to the relevant concepts such as organisation or person is difficult due to the complexity and specificity of the information to be extracted. The named entities can vary from simple people names to uncommon terms or phrases that do not have a universal meaning. In fact, terms in a document in a particular domain might have different meaning in another domain. Information Extraction in general and Named Entity Recognition in specific have issues that are related to the specificity of the information to be extracted. The knowledge of the targeted domain should be analysed to identify targeted concept sets. Then, the identified entities mapped to the predefined relevant concept (Piskorski and Yangarber 2013). However, Named Entity Recognition can be considered as prerequisite task that can be met by standard techniques as ANNIE.

In this research, the Named Entity Recognition pipeline is based on ANNIE pipeline. The tasks of ANNIE pipeline fall into two categorised, those that are domain-independent, and those that are not. For example, in most cases, the tokeniser, sentence splitter, POS tagger and co-reference resolution modules fall into the former category, while resources such as gazetteer lists and JAPE grammar rules will need to be modified according to the application domain.

The Natural Language Processing tasks in our Named Entity Recognition pipeline are applied on each document by using the GATE NLP Processing Resources. These Process Resource tasks are:

1. Tokenisation Process Resource Task
2. Gazetteer lists tagging Process Resource Task

3. Sentence splitter Process Resource Task
4. Part Of Speech (POS) tagging Process Resource Task
5. Morphological analyser Process Resource Task
6. Named Entity Recognition Rules Process Resource Task
7. Co-references resolution Process Resource Task
8. Dependency path tree tagging Process Resource Task

This pipeline is shown in Figure 5.2 below.

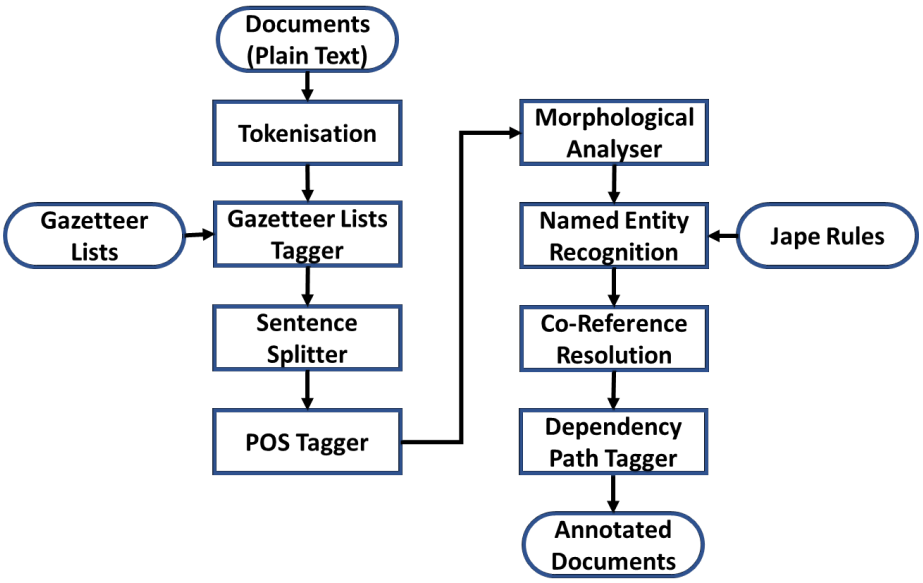

Figure 5.2: Named Entity Recognition and linguistic features generation pipeline

As shown in the figure above, there are two tasks should be executed after annotating the named entities. These are co-reference resolution and dependency path tagger tasks. However, in the next subsections will present all Named Entity Recognition pipeline tasks:

5.5.1 Tokenisation Process Resource

It is the process of splitting the stream of text into simple elements. Examples of these elements are words (such as “Apple” and “increase”), numbers (such as “123”, “5.6”), punctuation (such as “ ? “ and “ . “) and symbols (such as “@” and “£”) besides the space between words. These elements are called tokens. The tokenisation process also known as word segmentation. Because most of Information Extraction systems work with tokens as their input rather than using the raw text, it is an important to use a high-quality tokenisation tool (Maynard, Li and Peters 2008).

The input to this process resource is the text of the documents. All tokens in the documents are annotated and given an annotation type name “Token”. These tokens become input to other processing resources for further processing such as natural language parsing or grammatical tagging.

5.5.2 Gazetteer lists tagging Process Resource

The gazetteers list is one of the important resources that is used to recognise the named entities. In this task, we aim to perform two subtasks, collecting different types of named entities which are related to the economic domain to enrich the gazetteer lists and applying the gazetteer list tagger to annotated the targeted unstructured documents with theses gazetteer lists names.

Collecting gazetteer lists

ANNIE provides a rich list of gazetteers including geographical locations, person names, organization manes and miscellaneous entities such as nationalities, week day names and currency names. In fact, GATE allows the creation of user defined gazetteers. We utilised this facility to enrich these lists by adding more entities to exist lists such as organizations’ names and creating new lists such stock indexes’ names. In addition, we have separated lists of organization from the lists of stock indexes because GATE does not differentiate between them.

We created and developed a set of gazetteer lists to inform the Named Entity Recognition. The entries of these gazetteer lists were collected from the two main sources. The first source is from some organisations web sites which provide these name lists such as Invest Excel website (Khan 2018) that retrieve all companies’ names, their stock symbols and their stock indexes from Yahoo Finance. The second source is the Linked Open Datasets (LOD) such as DBPedia that can be utilised to retrieve company name, person names and location names. LOD datasets should be queries by using SPARQL queries. Below an example of the SPARQL queries which were used to retrieve companies names from DBPedia.

```
SELECT DISTINCT (str(?OrganisationLabel) AS ?OrganisationName)
WHERE {       ?organisation rdf:type dbpedia-owl:Organisation.
                ?organisation rdfs:label ?OrganisationLabel.
    FILTER (lang(?OrganisationLabel) = "en")    }
```

Below is a sample of the organisation names retrieved by using the above query.

```
===============
| OrganizationName        |
|===============
| 3Com                    |
| 7-Eleven                |
| Aardman Animations      |
-------------------------------
```

Table 5.1 below shows examples of gazetteer lists statistics that are collected in this research including their synonyms.

Table 5.1: Examples of gazetteer lists statistics

Gazetteer List Name	Number
Organization	> 16000
Stock Index	≅ 500
Stock Symbol	≅ 25000

Applying the gazetteer lists tagger

The gazetteer lists contain lists of the names in plain text files where each line has one entry name with different features to describe that name. Each list contains a set of names such as person names or locations, organisations, date and numbers. All tokens in the documents are matched and annotated with the gazetteer lists' entries. The gazetteer lists tagger looks up the tokens to match them with the entries in the lists. The matched tokens are annotated and given an annotation type name "Lookup". GATE can include more details to describe the entries in the gazetteer lists to be used as features to the lookups annotations. For example, this name entry line in the Organization list is as in Table 5.2 below:

Table 5.2: Example of name entry line in the companies' gazetteer list

List Name	Major	Minor	Annotating Type	Value	Feature 1	Value 1
company.lst	organization	company	Lookup	Active Care Inc.	Type	Medical Appliances Equipment

If the gazetteer lists tagger finds a token string that matches the name entry value "Active Care Inc." in a document, it will be annotated with an annotation type "Lookup" and with the following features shown in Table 5.3 below:

Table 5.3: The features associated with the companies' gazetteer list annotation type "Lookup"

Feature name	Value
Major	organization
Minor	Company
type	Medical Appliances Equipment

These lookup annotations and their features will be utilised to guide Named Entity Recognition process to identify the entity types in the documents. However, the gazetteer lists are effectively employed if the list of particular named entities class is limited. For

example, it is easy to identify the days of the week in text by referring to an existing list rather than writing complex rules to identify these entities. However, gazetteers can be used to store lists of keywords that can support identifying some entities within documents; for example, the abbreviations in company name (Co.) and the titles in persons names (Mr.).

5.5.3 Sentence Splitter Process Resource

It is the process of segmenting the text into sentences by identifying sentence boundaries between words in different contexts. Sentence splitting techniques are different in terms of determining whether punctuation tokens, such as ".", "?" and ":", mark the end of sentences or not. More complex cases arise when the text being processed is not plain text and contains tables, titles, formulae, or other formatting mark-ups such as HTML tags, hash tags in tweets (Maynard, Li and Peters 2008). This is the reason of employing a tool to detect and remove the boilerplate and templates around the main textual content of the online news documents.

In this task, the texts in the documents are segmented into sentences. These segmentation are annotated and given an annotation type "Sentence". In fact, the sentences are a key element in our Information Extraction pipeline. It is because our Relation Extraction tasks are based on the sentence context. Every entity pair for a targeted relation that appears in a sentence in unstructured data is identified and annotated as a candidate relation.

5.5.4 Part Of Speech (POS) Tagging Process Resource

It is the process of producing a part of speech tag for each word in the text to indicate its lexical syntactic category such as nouns and verbs. This tag is assigned to each annotated token as a linguistic feature. POS tagging of the text is required for Named Entity Recognition and Relation Extraction tasks. Table 5.4 below shows examples of some types POS tags and their symbols which are used by NLP tasks in GATE.

Table 5.4: Examples of POS tags types and their symbols used by NLP in GATE

POS tag	Description
NP	Proper noun - singular
NNP	Proper noun - singular: All words in names usually are capitalized but titles might not be.
NPS	Proper noun - plural
RB	Adverb: most words ending in '-ly'. Also 'quite', 'too', 'very', 'enough', 'indeed', 'not', '-n't', and 'never'.
RBR	Adverb - comparative: adverbs ending with '-er' with a comparative meaning.
RBS	Adverb - superlative
VB	Verb - base form: subsumes imperatives, infinitives and subjunctives.
VBD	Verb - past tense: includes conditional form of the verb 'to be'; 'If I were/VBD rich...'.
VBG	Verb - gerund or present participle
VBN	Verb - past participle
VBP	Verb - non-3rd person singular present
VBZ	Verb - 3rd person singular present
JJ	Adjective: Hyphenated compounds that are used as modifiers; happy-go-lucky.

JJR	Adjective - comparative: Adjectives with the comparative ending '-er' and a comparative meaning. Sometimes 'more' and 'less'.
DT	Determiner: Articles including 'a', 'an', 'every', 'no', 'the', 'another', 'any', 'some', 'those'.
IN	Preposition or subordinating conjunction

In this task, all annotated tokens are processed to produce their Part Of Speech tags. These tags are used to assist recognising the named entities, relation instances and ML features to extract relations between named entities.

5.5.5 Morphological analyser Process Resource

The morphological analyser processes the annotated tokens in the documents to identify the roots of each word considering their part of speech tags. For example, the singular form for the plural nouns (the root of “children” is “child”) and the present from for the past and past participle verbs (the root of “broken” is “break”). These values are added as a linguistic features to the annotated tokens.

In this task, the morphological analyser processes the annotated tokens in the documents to identify the roots of the words. These values are added as a linguistic features to the annotated tokens. They are also used to assist recognising the named entities, relation instances and ML features to extract relations between named entities.

5.5.6 Recognising the Named Entities by using the JAPE transducer

As aforementioned in the previous sections, we adopt the rule-based tool to recognise the named entities ANNIE in GATE. This tool recognises named entities by applying a set of patterns or regular expressions on the text for the deferent categories of names. These patterns utilise the linguistic features that are generated by using Natural Language Processing tasks and gazetteer lists tagging to support recognising a variety kinds of targeted named entities (Grishman 2012). Also, as aforementioned in the previous sections that our Named Entity Recognition pipeline is an adapted version of ANNIE pipeline. We have adapted two important resources, the gazetteer lists and JAPE rules. The gazetteer list adaption is explained in subsection 5.5.2 above and we will present in details of the JAPE rules in JAPE transducer Process Resource adaption in below.

JAPE transducer is also called the semantic tagger. The rules in this tagger are hand-crafted rules which are written in JAPE pattern language. The JAPE grammar language describes patterns to be matched with the context to produce annotations. JAPE provides finite state transduction over annotations based on regular expressions. Patterns can be specified by describing a specific text string or annotation such as those created by the tokeniser and gazetteers look up. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. JAPE rule consists of two parts, left hand side (LHS) and right hand side (RHS). LHS of the rules consist of an annotation pattern description. The RHS consists of annotation manipulation statements. Annotations matched on the LHS

of a rule may be referred to on the RHS by means of labels that are attached to pattern elements. Also, the RHS can consist of any Java code that can be used to manipulate features from previous annotations (Thakker, Osman and Lakin 2009).

The existing JAPE rules in the transducer of ANNIE pipeline covers most of the required named entities in our use-case scenario, which are Organization, Person, Location, Date, Percentage Values and Numbers. However, ANNIE JAPE rules do not differentiate between companies and stock indexes in recognising their named entities. Additionally, ANNIE JAPE rules do not include the recognition of the stock symbols of companies named entities. As a result, we created JAPE rules to recognise the Stock Index and Stock Symbols named entities. Below is a JAPE code example for annotating stock symbol entities that rely on gazetteer lists lookups:

```
Phase:  StockSymbol2
Input: Token Lookup Organization
Options: control = appelt
Rule: GazeStockSymbol2
(
 {Organization}
 ({Token})[0,3]
 {Token.string == "("}
 ({Token})?
 ({Token.string == ":"})?
 ({Lookup.majorType=="ticker"}):stsy
 ({Token})[0,3]
 {Token.string == ")"}
):all
-->
{
        gate.AnnotationSet stsy = (gate.AnnotationSet) bindings.get("stsy");
        gate.Annotation ann = (gate.Annotation) stsy.iterator().next();
        FeatureMap lookupFeatures = ann.getFeatures();
        gate.FeatureMap features = Factory.newFeatureMap();
        features.put("exchange",lookupFeatures.get("Exchange").toString());
        features.put("name",lookupFeatures.get("Name").toString());
        features.put("rule ", "GazeStockSymbol");
        outputAS.add(stsy.firstNode(), stsy.lastNode(), "StockSymbol", features);
}
```

If we apply the JAPE rule above on a document that contains the following sentence:

"State Bank of India (SBI), the nation's top lender by assets, reported on Friday a small increase in bad loans in its fiscal third quarter that was not as much as feared, sending its shares up as much as 6.8 percent."

The JAPE transducer will annotate "SBI" as a StockSymbol named entity with more information as features could be retrieve from Gazetteer lists' entries such as the name of exchange organization name which the company use this symbol in it.

According to our targeted domain knowledge analysis, use-case scenario and the semantic model, ontology, which are presented in chapter 3 and chapter 4 above, the concepts of the targeted named entities that will be extracted from online news are Organizations, Persons, Locations, Stock Indexes, Stock Symbols, Dates and Percentages Values. Although the Named Entity Recognition ANNIE tool in GATE was designed for recognise named entities on news texts, some of those mentioned concepts are not covered by ANNIE Jape rules and gazetteer list. Consequently, we extended ANNIE's JAPE rules and Gazetteer lists to recognise more named entities relevant to our domain of interest, which are stock indexes and stock symbols entities. Table 5.5 below shows the Precision, Recall and F1-measure results of recognising the targeted named entities by using the adapted ANNIE pipeline.

Table 5.5: The Precision, Recall and F1-measure results of recognising the targeted named entities by using the adapted ANNIE pipeline

Annotation Type	Precision	Recall	F1-measure
Date	1.0	1.0	1.0
Location	0.873	0.9524	0.911
Organization	0.9867	0.9107	0.9472
Percent	1.0	1.0	1.0
Person	0.722	0.9643	0.8257
StockIndex	1.0	1.0	1.0
StockIndex	1.0	0.9167	0.9565

The results in the table above have been obtained after applying the adapted ANNIE pipeline on online news documents sample to show how appropriate this tool for our problem domain.

5.5.7 Co-references resolution Process Resource

The co-references resolution, which also is known as Orthomatcher process resource, adds identity relations between named entities which are found by the Named Entity Recognition rules. This resolution does not find new named entities; however, it may correct the annotation type of unclassified named entity by using the annotation type of a matching named entity. There are three types co-references, named such as "Microsoft" and "Microsoft Co.", nominal such as "Microsoft" and "the company", and pronominal such as "Microsoft" and "it" (Clark and González-Brenes 2008).

This process task is applied after recognising and annotating the named entities to track the matched named entities in the whole document. For example, the identity relation between the entity "Apple Inc." and "Apple Company". It also could improve Named Entity Recognition by correcting the annotation types of unclassified named entities based on relations with existing classified named entities.

5.5.8 Dependency path tree tagging Process Resource

It is a natural language parser that accomplish the grammatical structure of sentences; for instance, which groups of words go together as phrases and which words are the subject or object of a verb in the sentence. This process resource attempts to follow the path of the grammatical relations hold between all pairs of words in a sentence such as adjectival complement relation between a verb and an adjective (de Marneffe and Manning 2014). For example, assume that we have the sentence below:

"IBM is the employer of Steve"

If the typed dependency path of this sentence is parsed by using the Stanford parser, the result will be as shown in Figure 5.3 below:

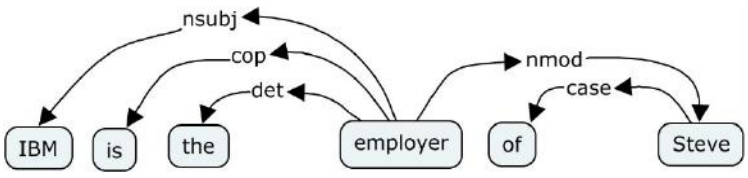

Figure 5.3: The typed dependency path Example

Where "employer" is the governor (head) in this sentence and the grammatical relations between the governor and the dependent words in the sentence are as following (de Marneffe and Manning 2014, De Marneffe, et al. 2014):

- cop (copula): the relation between the complement of a copular verb and the copular verb. A copular verb is a special kind of verb used to join an adjective or noun complement to a subject.
- det (determiner): A determiner is the relation between the head (employer) of and its determiner.
- nsubj (nominal subject): A nominal subject is a noun phrase which is the syntactic subject of a clause.
- nmod (nominal modifier): This relation is used for nominal modifiers of nouns or clausal predicates. It is a noun functioning as a non-core (oblique) argument or adjunct.
- case (case-marking): The case relation is used as a mediator between a modified word and its object including prepositions, postpositions, and clitic case markers.

This process task is applied after recognising and annotating the named entities because the main purpose of this task is extracting a linguistic features for Machine Learning Relation Extraction. We believe that the features, which are related to grammatical relation between the words in the sentence, are effective features because they represent the grammatical structure of sentences. By using this parser, we can add features; for example, which

groups of words go together as phrases and which words are the subject or object of a verb in the sentence or in the candidate relation instance.

Figure 5.4 below shows an example of tagged linguistic features and annotated named entities in a document by using GATE developer.

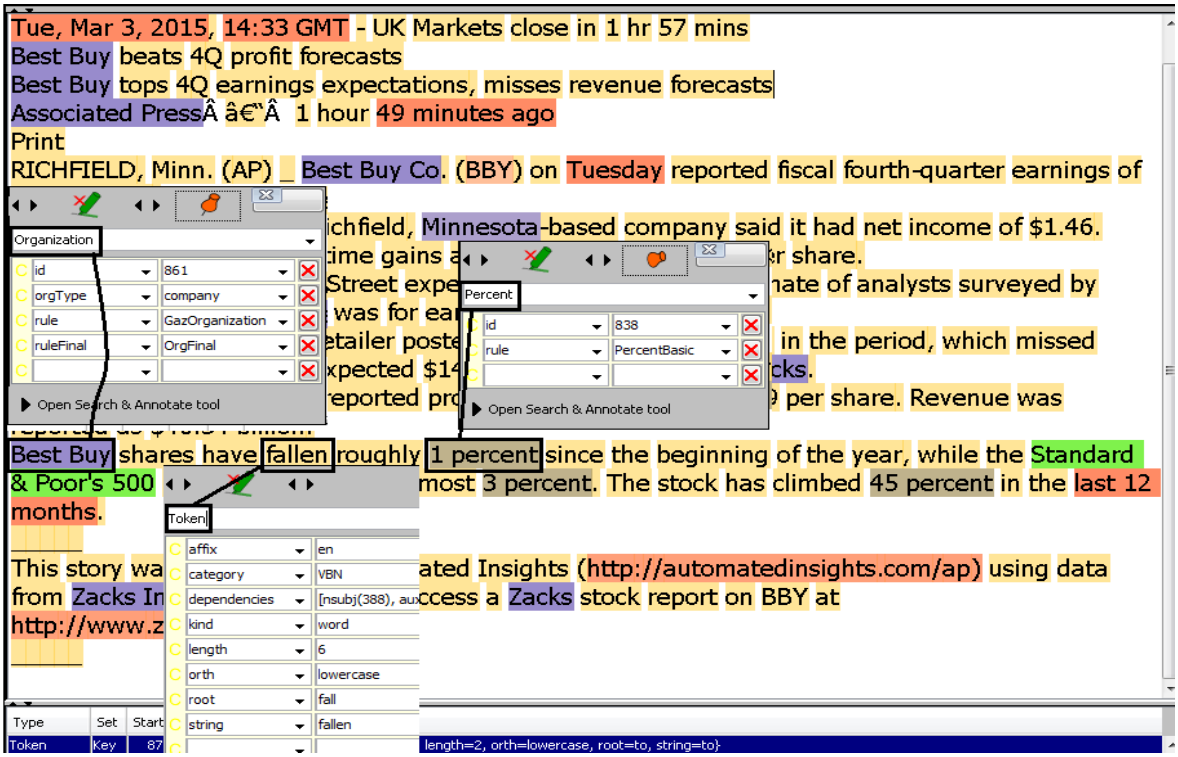

Figure 5.4:GATE developer Interface presenting examples of tagged linguistic features and annotated named entities in a document.

As illustrated in the figure above, the word “fallen” is annotated as a Token annotation type. The POS category is “VBN” which mean a past participle verb. The root of this verb is fall. There are features that could be useful for further processes such as the length of the word and the letters case (upper or lower). Also, this figure shows the Named Entity annotation of type Organization of kind company and Percent of type basic form.

The output of this stage’s tasks are annotated documents. These annotations include tokens and named entities. These tokens have their linguistic features. The named entities and the linguistic features will be used for detecting relation instances and extracting feature vectors for composing the training datasets to train the relation classifiers. Next chapter will present in detail the Relation extraction by using the supervised Machine Learning technique.

5.6 Summary

Obtaining valuable information from natural language texts in the online unstructured data requires applying automatic Information Extraction techniques. In fact, a considerable proportion of these online unstructured data sources exclusively service specific domains, which provides an opportunity in adopting knowledge-based approach. knowledge-based approach is about understanding the syntactic and semantic characteristics of the domain to aid Information Extraction process. Automatic Information Extraction could be considered as a pipeline process. It usually starts in recognising the named entities; then, entities co-references resolution; lastly, extracting the relation between the named entities in a certain event.

We divided the Information Extraction pipeline into two parts. The first part is about the linguistic pre-processing of sourced unstructured data and recognising Named Entities. The second part is about extracting the relations between the recognised named entities in the first part of the pipeline. In fact, the first part, also, is responsible on generating linguistic features for all words in all documents to be utilised in the relation classification in the second part. In this chapter, we presented the first part of Information Extraction pipeline, which is Named Entity Recognition pipeline tasks including the linguistic analysis by applying the Natural Language pre-Processing tasks.

In this research, the Information Extraction tasks is applied on the domain-specific unstructured documents which are collected from of online economic and finance news. They are retrieved by using the Rich Site Summary (RSS) feeds. Furthermore, we employed an open source tool to detect and remove the boilerplate and templates around the main textual content in the web page of the online news to increase the quality of the extracted information.

The Named Entity Recognition pipeline is based on ANNIE pipeline. We applied pre-Processing Natural Language tasks in order to recognise the linguistic components in text such as words and sentences and, also, to generate linguistic features of those components such as the part of speech type of those words. The Natural Language Processing tasks are applied on each document by using the GATE NLP Processing Resources. These Process Resource tasks in the Named Entity Recognition pipeline are:

1. Tokenisation Process Resource Task
2. Gazetteer lists tagging Process Resource Task
3. Sentence splitter Process Resource Task
4. Part Of Speech (POS) tagging Process Resource Task
5. Morphological analyser Process Resource Task
6. Named Entity Recognition Rules Process Resource Task
7. Co-references resolution Process Resource Task

8. Dependency path tree tagging Process Resource Task

The results of these tasks are linguistic features and Named Entities. They will be used for relation instances detection and features generation for Machine Learning relation classification.

6 Relation Classification Using a Hybrid of Rule-based and Machine Learning Approaches

6.1 Introduction

In the previous chapter, the first two stages of Information Extraction pipeline are accomplished. They are generating the linguistic features and annotating the named entities. The resultant linguistic features and named entities will be used to perform the last stage of our Information Extraction pipeline, which is Relation Extraction.

The comprehensive extraction of semantic relations between entities is required for different application areas such as natural language understanding because it is a crucial step in transforming unstructured data into structured knowledge to be queried by software agents by utilising knowledge representation approaches such as Semantic Web Technologies (Konstantinova 2014). As Yang-Turner, et al. in (Yang-Turner, et al. 2013) revealed that Semantic Web technologies are increasingly being adopted for aggregating Web data and assist users to access and make sense of the vast semantic space. The knowledge exploration tools, which are based on Semantic Web technologies, operate on semantically tagged contents using relationships from the underpinning ontologies. These tagged relations are a key to a majority of Information Extraction applications such as semantic search, question answering, knowledge harvesting, sentiment analysis and recommender systems.

In Relation Extraction task, the relationships between the named entities are identified according to the syntactic and semantic characteristics of the problem domain knowledge. Usually, these relations are binary or between two entities; for example, an organisation is an employer of a person. However, when more than one relation are related to each other including a place name and date, it is called non-binary relation extraction or event extraction (see chapter 4 above). Event extraction refers to the task of identifying events in unstructured data and; usually, they involve in extracting of several entities and relationships between them (Piskorski and Yangarber 2013, Karkaletsis, et al. 2011).

This problem of Relation Extraction is formulised by Hong in (Hong 2005) as shown in the form number (6.1) below:

$$(e_1, e_2, s) \rightarrow r \qquad (6.1)$$

Where (e_1) and (e_2) are two named entities existing in sentence (s) and (r) is a label of the relation between the two named entities. However, this formula does not include an important factor for Relation Extraction, which is the features required to extract relations.

These features could be linguistic or structural features and can be employed by both approaches, Rule-based and Machine Learning. Consequently, we extended the formula by adding the features factor as in equation (6.2) below:

$$(e_1, e_2, s, f) \rightarrow r(e_1, e_2) \qquad (6.2)$$

Where (f) is the features of the relation (r) between entities (e_1) and (e_2) in the sentence (s). The relations are extracted according to their features.

The application of Named Entity Recognition pipeline, which is explained in chapter 5 above, produces linguistic features and named entities. The linguistic features and named entities will be utilised for detecting relation instances and extracting feature vectors for training the relation classifiers. Then, the named entities and their interrelations will be populated to the semantic knowledgebase.

The next subsection reviews related works in Relation Extraction approaches and techniques.

6.2 Relation Extraction Related Works

There are two main approaches in Relation Extraction, Rule-based and Machine Learning based. The next subsections explain in detail these approaches and reviews some related published works.

6.2.1 Rule-Based Relation Extraction Approach

The main idea of Rule-based approaches is transforming the linguistic features space into lexical and syntactic patterns to be applied on natural language texts in order to extract relations. However, the relation extractors in these approaches depend on the similarity of the texts and a closed set of relations to be identified. Moreover, the patterns are manually crafted and small variations in these patterns can prevent finding appropriate relations. These patterns also are not straightforwardly applied on other domains (Garcia and Gamallo 2011, Konstantinova 2014). According to Konstantinova in (Konstantinova 2014), rule-based approaches could provide acceptable results if the main aim is to quickly extract relations in a well linguistically defined domains. The Relation Extraction in some of these domains rely on Rule-based systems because there is a sufficient domain knowledge to assist in handcrafting Relation Extraction rules. An example of those domains is the biomedical domain where there are clear medical taxonomies explaining the regularity and the specificity of the terminology in the text that can be assist building the Relation Extraction rules.

Several studies in the literature have reported the application of Rule-based approaches on Relation Extraction in biomedical domain; for example, Funderl, et al. in (Fundel, Kuffner and Zimmer 2007) present an approach to extract relations from free text of biomedical publications' abstracts. The approach is based on natural language pre-processing to produce dependency parse trees and apply a small number of simple rules to these trees. They applied this approach to medical documents' abstracts in order to extract relations between gene and protein.

In a different study by Huang, Zhu and Li in (Huang, Zhu and Li 2006), a new approach was proposed that integrates shallow parsing and pattern matching. It aims to extract protein-protein interactions from texts of full biomedical scientific papers. The approach extracted relations from sentences by a greedy pattern matching algorithm, along with automatically generated patterns. They claim that their approach achieves improvements compared with the traditional pattern matching algorithms.

6.2.2 Supervised Machine Learning Relation Extraction Based Approach

Machine Learning aims to provide increasing levels of automation in the knowledge engineering process by replacing time-consuming human activities with automatic techniques. In terms of supervision, Machine Learning algorithms can be categorised into supervised, semi-supervised and unsupervised algorithms.

Supervised Machine Learning algorithms create the classification models from labelled training data to make predictions about future instances. The model maps the inputs to the desired outputs by determining the class (e.g. the relation candidate) that the new input instances belong to (Song and Roth 2017, Bhavsar and Ganatra 2012).

Unsupervised Machine Learning algorithms typically use clustering techniques to find regularities or patterns in unlabelled data. The clusters that are discovered by Unsupervised Machine Learning algorithms could be useful for seeding a Semi-Supervised Machine Learning algorithms. In Semi-Supervised Machine Learning algorithms, sometimes referred to as self-supervised or weakly supervised, an initial small set of seeds or a set of training instances is supplied to supervised Machine Learning algorithms to begin the training process. These seeds are further used for recognition of new instances. However, the error propagation can pose a serious problem as irrelevant instances at the initial stages could generate more irrelevant instances at later stages and decrease the accuracy of the extraction process. However, the implementation, configuration and evaluation of all types of Machine Learning algorithms should be performed by using a set of trusted labelled

instances to be able to provide an objective evaluation of the methods applied (Yan, et al. 2009, Pundir, Gomanse and Krishnamacharya 2013, Konstantinova 2014).

Supervised Machine Learning based approaches have been widely adopted in information extract from unstructured text, chiefly in Named Entity Recognition and Relation Extraction (Aljamel, Osman and Acampora 2015). Supervised Machine Learning based approaches do not require deep linguistics skills and are therefore more effective than Rule-Based systems requiring the hand-crafting of rule sets (Appelt 1999, Liu 2011).

An example of the use of supervised Machine Learning on Relation Extraction is a study conducted by Hong in (Hong 2005). The extraction task is divided into two subtasks, relation detection and relation classification. The classification features were grouped into lexical, syntactic and semantic type of the entities. The experiments were conducted not only on relation classification but also on relation detection by using different features. The relation classifiers models were created by training SVM algorithm on version 1.0 of the Automatic Content Extraction two (ACE 2) corpus (Mitchell, et al. 2003). The results of those experiments showed that the most accurate classification is achieved upon using all feature sets. In general, the features that do not require any language processing achieved relatively high precision compared to other features.

Another study by Panchenko, et al. in (Panchenko, et al. 2012) propose a method for semantic relation extraction from the abstracts of Wikipedia articles using K-Nearest Neighbour (KNN) and Mutual K-Nearest Neighbour (MKNN) algorithms and two semantic similarity measures, Cosine and Gloss Overlap, to measure the nearest class instances neighbours to the target class. They use the data available from the DBPedia to build a set of definitions of English terms. They built pairs between concepts and definitions. The concept represents an exact one word title of a Wikipedia articles and definition represents a text of the first paragraph of these articles. The experiments described in this work were conducted on a subset of articles with titles containing no numbers and special symbols. Each word was represented as a triple with definitions or features. Their results showed that the number of extracted relations linearly depends on the number of nearest neighbours for both KNN and MKNN. They claim that the algorithms of semantic relation extraction are based on the component analysis and they believe that the semantically similar words have similar definitions or features.

With the same objective, Wang, et al. in (Wang, et al. 2006) investigated relation classification by SVM classifier and explored a diverse set of linguistic features. They applied their method on the Automatic Content Extraction 2004 (ACE2004) training data

(Doddington, et al. 2004). This training dataset consists of 5914 annotated relation instances of 41 relation classes. Each entity pair is assigned to one of these relation classes based on the extracted features. They carried out the experiments to investigate the impact of different features on the performance by adding them incrementally. The results showed that the entity features lead to the best improvement in performance.

However, Rule-based and Machine Learning based approaches can be integrated in Relation Extraction. For example, Minard, et al. in (Minard, et al. 2011) use a hybrid Rule Based and Machine Learning approach to examine the information access improvement in clinical documents concept, assertion, and relation identification. They automatically extract English medical concepts and annotate assertions on concepts by using Conditional Random Fields algorithm. They refined the output of this model by creating rules to correct errors observed when testing on the development corpus. They extracted three types of medical concepts, which are problems, tests, and treatments. Then, the annotation of assertions made on medical problems. Finally, they annotated the relations between concepts by using Support Vector Machines algorithm. Natural language patterns are used to extract features from the input texts, which were in turn used in training the Machine Learning algorithms.

There are two key processes in the supervised ML pipeline that can significantly impact the classification accuracy: the class instances labelling and feature vectors generation; both processes can benefit from formalised knowledge of the problem domain. In information Extraction, domain knowledge can play an important role in understanding the syntactic and semantic characteristics of the problem domain's text and subsequently, in improving Natural Language Processing tasks associated with automating or semi-automating the instances labelling process. For instance, in our implementation of Machine Learning based relation classification, domain-specific knowledge is used to compile some of our training datasets by drawing on relation mentions that feature as ground facts in public datasets such as DBPedia and Freebase by using a distant supervision approach. This approach alleviates the manual annotation effort for relation extraction, which can be a time-consuming and cumbersome task to undertake manually (Daelemans and Hoste 2002, Song and Roth 2017, Jiang, et al. 2012, Lawrynowicz and Tresp 2014).

One of the works published that adopt the distant supervision approach is the work of Mintz et al. in (Mintz, et al. 2009). Their effort utilises a Freebase dataset as a distant supervision source and a dump of the full text of all Wikipedia articles as a source of unstructured data. The training set is assembled from unstructured data sentences containing the entity pair

that appear in a relation mention in Freebase as a ground fact. Then, the linguistic features are extracted to learn a relation classifier. The ML classifier used in this research is a multi-classification logistic classifier optimised by using Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) with Gaussian regularisation. L-BFGS is an optimization algorithm for parameter estimation in ML (Andrew and Gao 2007). They analysed the features performance, which showed that syntactic parse features are particularly beneficial for relation classification. Their overall results showed that the distant supervision approach has the capability of extracting a high precision for a considerable number of relations.

6.2.3 Our Relation Extraction Approach

As aforementioned in the previous subsection, several studies have focused on Relation Extraction; however, we agree with Konstantinova in (Konstantinova 2014) who has drawn attention to the fact that it has still room for improvement. Similar to the work of Minard, et al. in (Minard, et al. 2011), we adopted a hybrid approach integrating Rule-based and Machine Learning based techniques. Our approach relies on Rule-Based techniques for recognising named entities, extracting relation instances and feature vectors from the input unstructured data; then, Supervised Machine Learning techniques are utilised for Relation Extraction based on named entities' relation instances and their feature vectors. For Named Entity Recognition we utilised the Rule-based approach based ANNIE pipeline system. With respect to Relation Extraction, we implemented and evaluated three ML classifiers that are commonly adopted for relation extraction from unstructured text: Support Vector Machine (SVM), Perceptron Algorithm Uneven Margin (PAUM) and K-Nearest Neighbour (KNN).

To generate the labelled instances for Machine Learning training datasets, we applied two methods, manual and automatic. We applied manual method to generate classified instances for irregular relations and we consult an appropriate experts for annotating instances. We applied automatic method to generate classified instances for common relations and we utilised an existing semantic datasets (Linked Open Data Cloud) as distant Supervision sources.

Our framework in general is a knowledge-based framework. Knowledge-based approach is based on understanding the syntactic and semantic characteristics of the domain knowledge. In this research, we employed the characteristics of the problem domain knowledge to aid Machine Learning based Relation Extraction processing tasks. For example, the grammar and the meaning of words in the context of the sentence structure or style of documentation language are engineered in linguistic or structural feature vectors.

Then, the impact of the quality of these features on the accuracy of relation classification was investigated.

6.3 Relation Classifiers

Selecting an appropriate Machine Learning algorithm depends on the problem specification and the nature of the data (Remya and Rama 2014). We implemented and evaluated three different supervised Machine Learning relation classifiers, Support Vector Machine (SVM), Perceptron Algorithm Uneven Margin (PAUM) and K-Nearest Neighbour (KNN). The works of Panchenko, et al. in (Panchenko, et al. 2012), Hmeidi, Hawashin and El-Qawasmeh in (Hmeidi, Hawashin and El-Qawasmeh 2008), Li, Bontcheva and Cunningham in (Li, Bontcheva and Cunningham 2009), Li, et al. in (Li, et al. 2005), and Witten and Frank in (Witten and Frank 2005) reveal that these algorithms are used in Information Extraction tasks with adequate results.

6.3.1 Support Vector Machine (SVM)

SVM is a supervised ML algorithm and it has an advanced performance for a diversity of classification tasks including Information Extraction; specifically, in small training datasets. One of the striking features of SVM is that it has a robust justification for avoiding over fitting (Cunningham, Maynard and Bontcheva 2014, Wang, et al. 2006). SVM is an optimal classifier, which means that it learns a classification hyperplane in the features space with the maximal margin to all training instances (Li, Bontcheva and Cunningham 2009). This work uses the GATE implementation, which is based on Java version of the SVM package LibSVM with exception that the GATE implements the uneven margins SVM algorithm which are described in the work of Li, et al. in (Li, Bontcheva and Cunningham 2009). The most important parameters of this implementation are SVM cost (C, the Cost associated with allowing training errors, soft margin), kernel type (In this research we used the default value of kernel type which is linear) and the uneven margins (τ or tau, setting the value of uneven margins parameter of the SVM) (Li, Bontcheva and Cunningham 2009, Li and Shawe-Taylor 2003).

6.3.2 Perceptron Algorithm Uneven Margin (PAUM)

PAUM is an effective learning algorithm especially for large training datasets. It has been successfully used for document classification and Information Extraction. For a binary classification problem, it checks each instances in the training dataset by predicting their labels. If the prediction is correct, the instance is passed; otherwise, it is used to correct the model. The algorithm stops when the model classifies all training instances correctly. The utilised GATE implementation of the PAUM algorithm proposes two margin parameters, positive (p) and negative (n) margins. These two margin parameters allow the PAUM to

handle imbalanced datasets better. Also, GATE implementation proposes the modification of the bias term parameter (optB) (Li, et al. 2005, Cunningham, Maynard and Bontcheva 2014).

6.3.3 K-Nearest Neighbour (KNN)

KNN is a simple and often its accuracy is enhanced when the number of features is small. It is an instance-based classification, which means that each new instance is compared with K nearest neighbour instances by using a distance matric. The class that has the majority of instances of the closest K neighbours is assigned to the new instance. KNN algorithm shows superior results in classifying documents. However, it is a lazy learning algorithm because it depends only on statistics. KNN has only one parameter (K) which can be tuned heuristically in order to find the best algorithm's performance. We used the implementation of this algorithm that is provided by GATE. This implementation is based on the open source ML package WEKA (Hmeidi, Hawashin and El-Qawasmeh 2008, Witten and Frank 2005, Imandoust and Bolandraftar 2013).

6.3.4 Classification Implementation methods

The algorithms above can implement both binary and multi-class classifiers. Multi-classification is usually solved in terms of multiple binary classifications by using a simple "one-vs-others" or "one-vs-another" models (Li, Bontcheva and Cunningham 2009).

The "one-vs-others" method converts N classes classifier (N>2) into N binary classifiers. Every binary classifier is trained with the positive instances that belong to a specific class and the negative instances that belong to all other classes. In contrast, "one-vs-another" method converts N class classifier (N>2) into N(N−1)/2 binary classifiers of class pairs. Every binary classifier is trained with the positive instances that belong to one class in the pair and negative instances that belong to the other class in the same pair (Aly 2005). Rfkin, et al. in (Rifkin and Klautau 2004) argue that the "one-vs-others" approach is simple, robust and the accuracy of its results is better or similar to other approaches such as the single machine and error-correcting coding approaches besides that it requires less number of models. For these reasons, a number of studies have employed this multi-class approach; for example, the work of Archibald, et. al in (Archibald and Fann 2007) and the work of Chandrashekar, et. al in (Chandrashekar and Sahin 2014). Hence, we adopted the "one-vs-others" method to transform multi-classifier into multiple binary.

The next subsections present how we generated the training datasets, tuned the algorithms' parameters and selected the best feature subsets for relation classification.

6.4 Relation Detection and Generating the Training Datasets

Classifying the relation between the Named Entities in this work is sentence-level Relation Extraction. Every entity pair for a targeted relation that appears in a sentence in unstructured

data is identified and annotated as a relation instance and it is assumed to represent one relation type. These pairs should be chosen to represent relations in the targeted domain ontology. Relation detection grammar rules are encoded using GATE's pattern matching language JAPE (Thakker, Osman and Lakin 2009).

We retrieved more than 18 thousand documents from of online sources that are about stock market news by using the Rich Site Summary (RSS) feeds. Examples of these RSS feeds are given in Table 4.1 of subsection 4.2.

The number of detected relation instances of the targeted relations in this work is shown in Table 6.1. These relation instances will be used to compile the relation classification's training datasets.

Table 6.1: The sentences and relation instances number of all pairs

Annotation Type	Pairs Number
Relation Instances of Person-Organization pair	3619
Relation Instances of Person-Location pai	10682
Relation Instance of Location-Organization pair	3029
Relation Instances of StockSymbol-Organization pair	316
Relation Instances of StockIndex-Organization pair	241
Relation Instances of Organization-Percent pair	1706
Relation Instances of StockIndex-Percent pair	356
Relation Instances of Organization-Date pair	878
Relation Instances of StockIndex-Date pair	394

The training datasets consist of a set of labelled instances. These instances are described by a feature vectors. A supervised ML algorithms analyse the training datasets and creates a model to be used for predicting instances' classes in unlabelled data. There are two methods to generate the labelled instances, manually by human experts or automatically from existing semantic datasets (Pundir, Gomanse and Krishnamacharya 2011). In our research, we applied both of these methods to generate the labelled instances, manually for irregular relations and automatically for common relations. For automatic generation of classified instances, we utilised an existing semantic datasets (Linked Open Data Cloud) as distant Supervision sources by following Mintz, et al. in (Mintz, et al. 2009) distant supervision Machine Learning approach. The next two subsection will present these two methods of generating the labelled instances for the training datasets.

6.4.1 Generating training datasets from online structured datasets

We have employed Semantic Web Technologies to standardise, describe and model our problem domain knowledge. The same standardised metadata is used in public datasets in the Linked Open Data (LOD) Cloud to publish ground facts that are relevant to various problem domains. These ground facts can be used to compile training datasets for relation

classification and enriching the resulting knowledgebase. Hence, we adopted a knowledge-driven distant supervision ML approach to extract common entity pairs' relations by utilising two existing knowledge datasets as a distant supervision sources. These datasets are DBpedia and Freebase. DBpedia contain more than 4.5 million entities and more than 3 billion RDF triples for a diversity of languages. Freebase dataset contains approximately 47.5 million topics and 2.9 billion facts in English language.

The training datasets were built by retrieving the relations between any two entities in a single sentence in the unstructured document that are mentioned in Freebase or DBpedia as ground facts. These relations are assumed to be a class instance or true positive in the training datasets. The mentioned relations in the semantic datasets were extracted by using SPARQL engine of JENA. JENA is a free and open source Java framework for building Semantic Web and Linked Data applications. SPARQL Protocol and RDF Query Language is recommended by W3C and it is a common method for querying RDF stores (Harris, Seaborne and Prud'hommeaux 2013, W3C 2018, Prud and Seaborne 2006).

To illustrate this task, we use the following sentence example from the unstructured data corpus that is used in this work:

"Yesterday Twitter's boss Dick Costolo said he was ashamed at how the site had dealt with abusive online trolls."

The sentence contains the following relation instance:

"Twitter's boss Dick Costolo"

The relation instance contains two entities, Person entity "Dick Costolo" and Organization entity "Twitter".

These two entities' names are used to query the semantic datasets to find if they have any mentioned relation in BDPedia or Freebase. The SPARQL query for this example and its result are shown below.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT (str(?lbl) AS ?result)
WHERE {
    { ?entity1 ?rel ?entity2 .
      ?entity1 rdfs:label "Dick Costolo"@en .
      ?entity2 rdfs:label "Twitter"@en
    }
    UNION
    { ?entity2 ?rel ?entity1 .
      ?entity1 rdfs:label "Twitter"@en .
      ?entity2 rdfs:label "Dick Costolo"@en
    }
    ?rel rdfs:label ?lbl
    FILTER ( lang(?lbl) = "en" )
  }
```

The result of the above query is:

```
------------------
| result       |
========
|"employer" |
-----------------
```

This result indicates that the relation mentioned in the semantic dataset in the form of RDF triple is as follows:

"Dick Costolo employer Twitter"

This relation is mapped into a relation in our domain's ontology as:

"Twitter employerOf Dick Costolo"

Then, the relation is assumed as a class instance or True Positive in the Person-Organization training dataset. Table 6.2 below shows the four training datasets that were generated using distant supervision approach adopted by this work.

Table 6.2: The summary of the collected training datasets by using Distant Supervision method (Doc=Documents)

Entity Pairs Training Datasets	Doc	Relation Classes and Instances		
		Types	instances	Total
Person-Organization (3 classes)	161	founderOf	38	204
		keyPersonIn	107	
		employerOf	59	
Person-Location (4 classes)	636	hasPlace	221	896
		birthplace	233	
		hasNationality	415	
		deathPlace	27	
Location-Organization (1 classes)	281	locatedIn	299	299
StockSymbol-Organization (1 classes)	71	issuedBy	83	83

6.4.2 Generating training datasets manually

Although manual annotation of ML relation instances is labour-intensive task, it is generally considered to be more precise than automatic annotation (Petrillo and Baycroft 2010). In this research we applied manual annotation to generate training datasets to extract uncommon relations between pairs that could not be found in exiting semantic datasets such as DBpedia and Freebase. We employed GATE annotation editing facilities to extract and annotate the training instances for Machine Learning.

We started by applying Jape rules to annotate all relation instance of the relations between the targeted named entities in the sentences in the documents. Then, we annotated these relation instances as True Positive or True Negative manually. Table 6.3 shows the five training datasets that their instances were collected manually.

Table 6.3: The summary of the collected training datasets by using manual method (Doc=Documents)

Entity Pairs Training Datasets	Doc	Relation Classes and Instances		
		Types	instances	Total
StockIndex-Organization (1 class)	44	memberOf	69	69
Organization-Percent (4 classes)	399	shareIncreasedBy	257	753
		shareDecreasedBy	259	
		profitIncreasedBy	155	
		profitDecreasedBy	82	
StockIndex-Percent (2 classes)	91	indexIncreasedBy	115	234
		indexDecreasedBy	119	
Organization-Date (4 classes)	344	shareIncreaseDate	253	525
		shareDecreaseDate	63	
		profitIncreaseDate	157	
		profitDecreaseDate	52	
StockIndex-Date (2 classes)	170	indexIncreaseDate	204	272
		indexDecreaseDate	68	

6.5 Features Extraction

We argue that the sufficient domain knowledge could assist in selecting the features vector as input to the classification algorithms. Consequently, we exploited the domain knowledge to create a new set of features for ML relation classification and expanded on the feature set suggested by Mintz, et al. in (Mintz, et al. 2009) to provide a more comprehensive set of features; for instance, we added dependency paths and entity description features. As the dependency path (grammatical relation) between the related entities is not always apparent (de Marneffe and Manning 2014, Fundel, Kuffner and Zimmer 2007), we took into consideration the dependency paths of all words in the sentence including the candidate relation entities. The entity description features include its Part Of Speech annotation, the entity string and the number of words in the entity.

The features are categorised into three categories, Lexical features, Syntactic Features and Named Entity Features. These features are extracted by using JAPE rules and GATE Embedded; then they have been added to every relation instances in the training datasets. Table 6.4 presents these features list.

Table 6.4: ML Features Vector list

Features Category	Feature Name	Description
Lexical features	poslist	POS of words between entity pairs
	genposlist	General POS of words between entity pairs
	posbefore	POS of three words before the left entity
	posafter	POS of three words after the right entity.
	posentity1	POS of the first entity

	posentity2	POS of the second entity
Syntactic Features	dependencyWords	The words' strings of collapsed typed dependency path between entity pairs
	dependencyKinds	The kinds of collapsed typed dependency path between entity pairs
	dependencyPath	The whole collapsed typed dependency path of the entity pairs' sentences
	directDep	Direct collapsed typed dependency path between entity pairs
	wordsStrSeq	The strings of the words between entity pairs
	depDistance	The number of the collapsed typed dependency words between
Named Entity Features	enttokensno1	The number of tokens in the first entity
	enttokensno2	The number of tokens in the second entity
	order	The order of the entities
	distance	The number of tokens between the two entities
	entityString1	Token string of the first entity
	entityString2	Token string of the second entity
	typeentity1	The type of the first entity
	typeentity2	The type of the second entity

6.6 Parameters Optimisation

ML algorithms' parameter optimisation is the problem of choosing a set of parameters' values for improving the results of ML models. The purpose of parameter optimisation is improving the ML classifiers' performance by tuning the ML algorithms' parameters. Lorena, et al. in (Lorena and De Carvalho 2008) report that there are generally three methods to find the Machine Learning algorithms' parameters optima: use the default values, define the values by grid search and automatic search through optimization techniques such as Genetic Algorithms. Grid-based search is commonly used to perform parameter optimization, where the default values for the ML algorithms' parameters are evaluated against the other values in the grid. In this work, we adopted grid-based search to perform parameter tuning as it is sufficient to satisfy the requirements of the deployed Machine Learning techniques, and is simple to implement in comparison with the computationally expensive automatic optimisation techniques.

Practically, grid search starts with a finite set of reasonable values for each parameter. These values are selected manually in accordance each algorithms' specifications. Then, the selected grid sets are used to train the ML algorithms and evaluate their performance against ground-truth in a k-fold validation process. Finally, the parameters that achieve the highest model performance are chosen (Bergstra and Bengio 2012, Hsu, Chang and Lin 2003). In this work, the finite sets of parameter values for SVM and KNN parameters (C and tau for SVM, K for KNN) were heuristically selected by studying the specifications and recommendations of those algorithms. However, for PAUM's parameters (p, n and optB) values set, we relied on the recommended parameters' values by the work of Li, et al in (Li,

et al. 2002). Table 6.5 shows the parameters of SVM, PAUM and KNN that are selected by using grid search experiments.

Table 6.5: The Grid Search Results of optimum ML algorithms Parameters

ML	P	Grid Result	Description
SVM	C	1	The Cost associated with allowing training errors (soft margin)
	tau	0.8	Setting the value of uneven margins
PAUM	p	10	Positive margin
	n	1	Negative margin
	optB	0.3	The modification of the bias term
KNN	K	1	The number of the nearest neighbour instances

6.7 Tuning The Relation Classifiers

Before start evaluating the relation classifies, we should tune these classifiers to fit the relation classification in our problem domain. Firstly, we should decide the methods and techniques to measure the evaluation of these classifiers. Next section will present the adopted methods and techniques in this research.

6.7.1 Methods and Techniques to Measure Classifiers' Evaluation

Precision and Recall are two factors that are useful to characterise and measure the performance of ML algorithms. Precision is the ratio of the number of the correctly annotated instances (True Positive) to the total number of the annotated instances (True Positive and False Positive). Recall is the ratio of the number of the correctly annotated instances (True Positive) to the total number of the correct instances (True Positive and False Negative). However, based on the nature of the classification tasks of Information Extraction applications, a trade-off or balancing between precision and recall should be made. Obviously, different applications of Information Extraction have different requirements for precision and recall. The balanced measure that combines precision and recall is the traditional F1-measure. The F1-measure is a single scalar value that represents the harmonic mean of precision and recall (Valstar, et al. 2012, Davis and Goadrich 2006, Hattori, et al. 2008, Minkov, et al. 2006). In this research, F1-measure is used as the evaluation measure because we are looking to select a classifier based on a balance between precision and recall. On one hand, our system should ensure that the number of correct annotations (True Positive) are high; on the other hand, both the number of incorrect (False Positive) and missing (False Negative) annotations should be low.

There are two commonly used evaluation methods for ML algorithms, K-fold cross-validation and holdout test. In K-fold cross-validation, the corpus is split into K equal size partitions of documents. The evaluation run is repeated K times (folds). Each partition is used as test dataset and all the remaining partitions as a training dataset for all K folds. The overall Recall, Precision and F1-measure result of this method is the average of the all folds' results. In contrast, in holdout test, a number of documents in the training datasets are randomly selected according to a specified ratio, the default is 66%. All other documents

are assumed to be testing dataset (Shalev-Shwartz and Ben-David 2014, Cunningham, Maynard and Bontcheva 2014). In this work, we used cross validation K-Fold with K=10, which is empirically found to be the best method in practical ML evaluations as reported by Witten et. al in (Witten and Frank 2005). They conducted extensive tests on several different datasets with different learning techniques, and concluded that 10 is the most suitable number of folds to catch the most ML predication errors.

Moreover, there are two different options for computing precision, recall and F1-measure over a corpus, micro averaging and macro averaging. In micro averaging, the corpus is treated as one large document, where True Positive, False Positive and False Negative are counted through the entire corpus, and precision, recall and F1-measure are calculated accordingly. On the other hand, macro averaging computes precision, recall and F1-measure by counting True Positive, False Positive and False Negative on every single document and then averages the results for the entire corpus (Cunningham, Maynard and Bontcheva 2014). Macro Averaging is more appropriate for our problem domain since the sourced financial news articles represent independent documents.

According to Witten, et al. in (Witten and Frank 2005), there is more than one method to plot the evaluation results of ML algorithms performance. These methods depend on the target domain. For instance, the marketing domain uses lift chart by plotting True Positive rate versus training subset size, the communication domain uses Receiver Operator Characteristic (ROC) curve by plotting True Positive rate versus False Positive rate and the Information Retrieval domain uses Precision versus Recall curve. This research computes the evaluation results of ML models in relation classification by drawing the relation between recall and precision in terms of the confidence threshold for classification or the threshold probability classification as it is commonly accepted as the standard in the Information Extraction field. According to Davis, et al. in (Davis and Goadrich 2006) precision versus recall curve is useful to characterise the ML algorithm's performance; specifically, when dealing with imbalanced training datasets.

Consequently, we will use Precision, Recall and F1-measure to measure the accuracy of relation classifiers in macro Averaging cross validation K-Fold with K=10. Also, we draw the relation between recall and precision in terms of the confidence threshold for classification or the threshold probability classification to compute the evaluation results of relation classifiers. As presented in Table 6.2 and Table 6.3, we generated nine different training datasets that cover different relations between different entity concepts in the financial and economic news domain. These training datasets with the features vectors have been utilised to create the ML relation classification models. These models should be optimised to be evaluated before applying them to extract relations from unstructured data. In next subsections, the training datasets were optimised by reducing their classes imbalance and choosing the optimum probability threshold for classifications to reach the optimum results.

6.7.2 Optimising the Relation Classifiers in terms of determined classes imbalance

Generally, the classification models tend to favour the majority classes while incorrectly classifying the instances from the minority classes. According to Agrawal, et al. in (Agrawal, Viktor and Paquet 2015), if the size of one class's instances is much more than other classes' instances in a training dataset, it is considered imbalanced. In our training datasets, specifically those that are generated by using public distant supervision sources (DBpedia and Freebase), the number of negative relation instances is large. This is attributed to the fact that some relations in our unstructured data will be incorrectly assumed to be negative instances as they are not included as ground facts in the sourced public datasets. We believe that these negative relation instances can disrupt the balance between True Positives and Negatives instances of the classes in the training datasets.

This set of experiments attempts to alleviate the classes' imbalance in terms of True Positive and True Negative numbers in order to improve the accuracy of the classification model and to speed up ML processing. In these experiments, we heuristically measure the impact of reducing the number of negative relation instances on the models' accuracy by reducing or removing the relation instances in the documents that are not mentioned in the distant supervision sources. We also explicitly add some negative relation instances in the training datasets of one relation class in order to decrease in the true positive rate while maintaining a low false positive rate as recommended by Mohamed, in (Mohamed, El-Makky and Nagi 2015). Table 6.6 and Figure 6.1 below show the impact of reducing the number of negative Relation Instances on ML models accuracy in terms of F1-measure. As shown in the table and figure, there is a significant difference between the F1-measure values when applying the model of training datasets that contain negative relation instance considerably more than positive relation instances.

Table 6.6: The impact of reducing the number of Negative Relation Instances on ML models accuracy in terms F1-measure for two Automatically Collected Training Datasets

Training Datasets	Positive Relation Instances	Negative Relation Instance	SVM (F1)	PAUM (F1)	KNN (F1)
Location-Organization	299	256	0.716	0.727	0.703
		2730	0.484	0.483	0.479
StockSymbol-Organization	83	55	0.854	0.866	0.818
		233	0.76	0.766	0.787

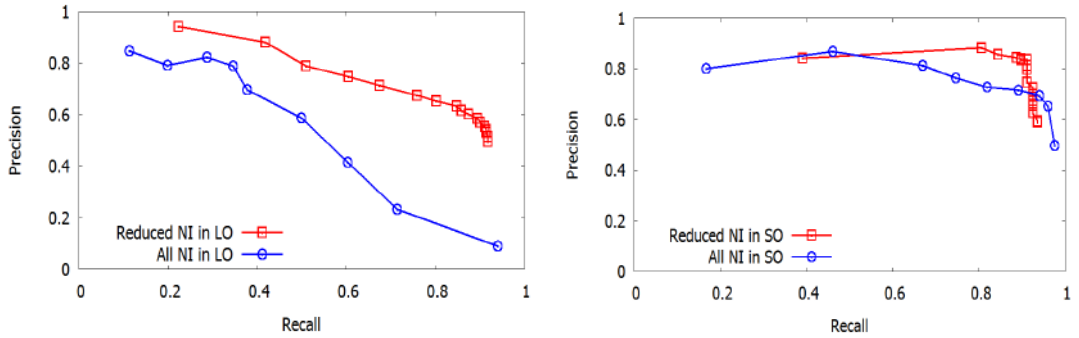

Figure 6.1: SVM model accuracy in terms of the number of non-relevant relation instances(NI) in two entity pairs training datasets, Location-Organization(LO) and StockSymbol-Organization(SO).

Mintz et. al in (Mintz, et al. 2009) utilise multi-class logistic classification for relation extraction, and they report that the negative relations instances had a minor effect on the performance of their classifier. However, for the implemented SVM classification, it is evident from Figure 6.1 that the SVM model accuracy clearly improves as we reduce the number of the True Negative relation instances because the class distribution in the training datasets does play a major role in the performance of most classification algorithms as highlighted by Agrawal, et. al in (Agrawal, Viktor and Paquet 2015).

6.7.3 Optimising the Relation Classifiers in terms of the probability threshold

The probability threshold for classifications was first explored by Lewis in (Lewis 1995). They argue that the best classification results for a set of instances that are assigned to a class if their probability of class membership is greater than a probability threshold ρ, where $0 \leq \rho \leq 1$. For example, with the default probability threshold value of 0.5, the predicted probability value of any instance to be a member of a certain class as a true positive must be greater than 0.5. However, Freeman and Moisen in (Freeman and Moisen 2008) have asserted that the accuracy of the classification models is affected by the value of the threshold. They added that the default threshold value of 0.5 does not necessarily produce a highest prediction accuracy; particularly, when the datasets are highly imbalanced. Therefore, by means of experimentation, we heuristically selected the best threshold value for all classification models on all training datasets by drawing on the correlation between the threshold probability classification and F1-measure.

By means of experimentation, we heuristically selected the best threshold values for all classification models on all training datasets by drawing on the correlation between the threshold probability classification and F1-measure. Our experiments have indicated that

the accuracy of the classification models is affected by the value of the probability threshold. From Figure 6.2 below, we can see that the classification accuracy is better when the probability threshold value is other than the default threshold value 0.5. The peak values of the probability threshold are 0.45 for the Person-Organization model and 0.4 for the Organization-Date model.

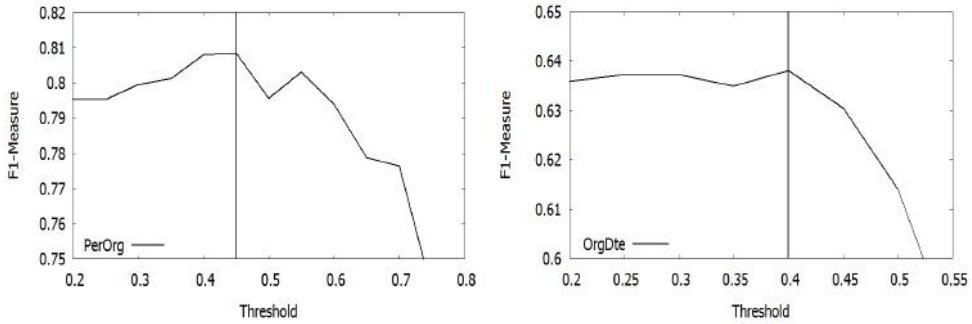

Figure 6.2: Indicates the impact of the probability threshold in the performance of SVM relation classifiers' models. The threshold peaks are 0.45 for the Person-Organization model and 0.4 for the Organization-Date model.

6.8 Relation Classifiers Evaluation Discussion

Table 6.7 presents the results of 10-fold cross-validation of all training datasets in terms of Precision, Recall and F1-measure. Also, the table indicates the best F1-measure in terms of the best probability threshold value.

Table 6.7: The results of 10-fold cross-validation of all training datasets in terms of Precision, Recall and F1-measure

TDS	SVM				PAUM				KNN			
	Thr	P	R	F1	Thr	P	R	F1	Thr	P	R	F1
PerOrg	0.45	0.848	0.775	0.808	0.15	0.855	0.778	**0.814**	0.85	0.809	0.744	0.775
PerLoc	0.5	0.778	0.701	**0.737**	0.15	0.756	0.703	0.728	0.8	0.719	0.627	0.67
LocOrg	0.45	0.537	0.915	0.676	0.5	0.67	0.804	**0.727**	0.75	0.695	0.729	0.71
StsOrg	0.5	0.731	0.897	0.801	0.15	0.832	0.911	**0.867**	0.75	0.845	0.834	0.828
StiOrg	0.6	0.788	0.955	0.855	0.5	0.799	0.979	**0.877**	0.4	0.793	0.913	0.845
OrgPct	0.45	0.726	0.61	**0.662**	0.15	0.703	0.6	0.646	0.7	0.611	0.577	0.594
StiPct	0.5	0.71	0.697	0.703	0.5	0.733	0.721	**0.727**	0.5	0.728	0.686	0.705
OrgDte	0.4	0.647	0.629	**0.638**	0.15	0.645	0.603	0.623	0.45	0.578	0.574	0.576
StiDte	0.6	0.767	0.711	0.737	0.5	0.745	0.745	**0.745**	0.5	0.697	0.697	0.697

From the data in Table 6.7, it is apparent that the Precision values of the classifiers with all training datasets are greater than the Recall values except three training datasets, Location-Organization, StockSymbol-Organization and StockIndex-Organization. These

training datasets has one relation class of positive instances. We added some negative relation instances to these training datasets as a second class for the binary classification; also, to balance between positive and negative instances. The recall is greater than precision because the number of spurious annotated labels is more than the number of missing labels when compared to the training labels. In other words, the number of False Positive instances is greater than False Negative instances. This could occur because the added negatives instances are misleading the classifier to incorrectly annotated them and increase the number of False Positive annotations.

It is apparent from the data of this table also that the probability threshold values are tending to be around the default value (0.5) for SVM classifier, tending to be small for PAUM classifier and tending to be large for KNN classifier when applying those classifiers on the majority of training datasets. The reasons for this can be returned to the different mechanism of these classifiers, and the different sizes and the different characteristics of the datasets. This can be confirmed by Figure 6.3 below. In this figure, we can see that the change in the probability threshold value has less impact on the precision and recall rates when using PAUM and KNN classifiers comparing to the impact of SVM classifier.

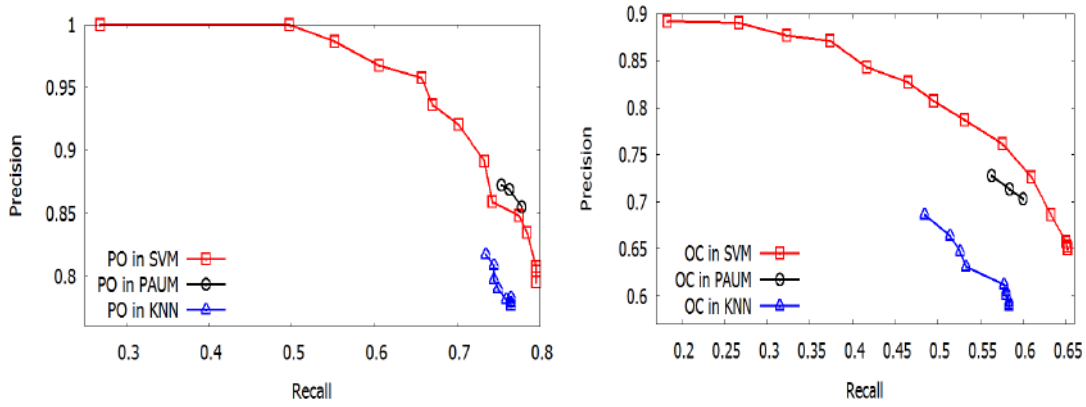

Figure 6.3: Examples of the impact of the probability threshold values change on the precision and recall rates by applying SVM, PAUM and KNN classifiers on Person-Organization and Organization-Percent training datasets.

In addition to the training datasets' generation, ML classification tasks require assigning features vector to a finite set of classes in those training datasets. Features represent distinctive aspects, qualities or characteristics of classes. The accuracy of the classification models not only depend on the quality of the individual features, but also on the best feature combinations. Feature selection is particularly important to relation extraction as most relation extraction methods are based on Machine Learning techniques (Jiang and Zhai 2007, Han, Kamber and Pei 2011). Next chapter will present the feature section approach that is adopted in this research.

7 Feature Selection Optimisation by Using Genetic Algorithms as a Wrapper Approach

7.1 Introduction

Feature selection is crucial because small number of features can result in model underfitting and vast number of features can lead to model overfitting. Searching for an optimal subset would be very expensive especially when the features vector is high-dimensional (Anbarasi, Anupriya and Iyengar 2010). The computational complexity of finding the best features combination (" $2^n - 1$ " for n features) makes it difficult to perform this task using heuristic methods thus necessitating the deployment of automatic search techniques.

The process of feature selection is employed in supervised Machine Learning techniques to reduce the dimensionality of classification problem. It aims to remove the redundant, irrelevant and noisy features for robust training datasets representation. This could save the running time of a learning process, could make ML algorithms build an accurate classification models and could improve the performance of the classification model.

The process of feature selection implementation can be described as follows, the process is initiated by generating candidate feature subset by establishing a certain search strategy. Each candidate subset is evaluated according to a certain evaluation criterion. The evaluation result of the candidate feature subset is compared with the evaluation result of the previous selected feature subset. If it is better, then the previous selected feature subset is replaced by the candidate feature subset. The process of subset generation and evaluation is repeated until a given terminating criterion is satisfied to select the best feature subset. There are many feature selection algorithms have been proposed in the literature. The main differences between these algorithms are the search strategy and evaluation criterion that are adopted by these algorithms (Tan 2007). In next subsection, we will present more details on features selection related works and then we will demonstrate our implementation and evaluation of features selection in this research.

7.2 Feature Selection Background and Related Works

Early studies had highlighted the need for feature subsets selections in relation classification such as the works that are aforementioned in section 6.2. Hong in (Hong 2005) used heuristic methods to select the best feature subsets by distributing the features into three sets and using each set or the combination of the sets to create the classification model to find the best of them, while Wang, et al. in (Wang, et al. 2006) investigated the

impact of feature subsets on the accuracy by applying the manual forward selection techniques by incrementally adding features to features vector to create the classification model. On the other hand, there are some relation extraction efforts such as (Panchenko, et al. in (Panchenko, et al. 2012) and Mintz, et al. in (Mintz, et al. 2009) do not deploy any feature subsets selection techniques.

Feature selection process can be divided into two tasks, generating features subsets and evaluating the generated feature subsets. However, the search space of features is exponentially increased, and it is prohibited for exhaustive search. Hence, many strategies have been proposed in the literature for generating the features subsets (Kumar and Minz 2014, Chandrashekar and Sahin 2014, Bagherzadeh-Khiabani, et al. 2016), which are:

- Sequential search: They are two types of sequential search, forward and backward. In forward selection, the search starts with no features and iteratively adds one feature at a time selected by an evaluation criterion. In backward selection, the search starts with the set of all features and proceeds by discarding one by one the feature yielding the worst estimated classifier accuracy.
- Complete search: Exhaustive search is an optimized search that guarantees the best solution. However, optimal searches need not be exhaustive. Different heuristic techniques can be used to reduce the search space without strengthening the optimal solution.
- Random search: This search comprises all operators that able to generate random features subsets in a single step. For example, Evolutionary Algorithms, Simulated Annealing and Random Mutation Hill-Climbing. After feature subsets are generated, they are evaluated by a certain criterion to measure the targeted classification model accuracy improvement.

However, based on the evaluation criteria, feature selection approaches can be classified into two categories: the Filter approaches and the Wrapper approaches. In other words, the best features are selected whether by involving the targeted classification model in the automatic search techniques, e.g. Genetic Algorithms, or by ignoring the interaction with the targeted classification model (Kumar and Minz 2014, Chandrashekar and Sahin 2014, Bagherzadeh-Khiabani, et al. 2016). Next subsection present a comparison between two evaluation criteria, Filter and Wrapper approaches.

7.2.1 Features Selection Approaches Evaluation Criteria, Filter and Wrapper Approaches

Filter approaches assess the relevance of features by describing a dataset from the perspective of consistency, dependency and distance metrics. All the features are scored and ranked based on certain statistical criteria. The features with the highest-ranking values are selected and the low scoring features are removed. The best feature subset for the

classifier model is selected independently because it ignores the targeted classification model performance on the reduced feature set.

Wrapper approach embeds the targeted classification model performance to assess the relevance of the features. After a search procedure in the space of possible feature subsets is defined and various subsets of features are generated, the evaluation of a specific subset of features is obtained by training and testing the targeted classification model. To search the space of all feature subsets, a search algorithm is wrapped around the classification model (Kumari and Swarnkar 2011, Brester, et al. 2016).

There have been several studies in the literature reporting a comparison between filter and wrapper evaluation criteria. All these studies agree that Filter approach requires less computational resources than Wrapper approach because it does not involve the targeted classification model performance in assessing the selected features subsets every time the features combinations are selected. They agree, also, that the Wrapper approach is more accurate than Filter approach because the Wrapper approach selects the best feature subset by directly involving the targeted classification model performance in accuracy measures to ensure that it is improved. For example, the work of Kumari, et al. in (Kumari and Swarnkar 2011) reveal that Wrapper approaches could be recommended in order to better validate the results and this is the reason for the increased use of wrapper method. The results of an additional work of Xue, et al. in (Xue, Zhang and Browne 2015) (Bing Xue) to compare between Filter and Wrapper approaches show that Filter approaches are usually faster than Wrapper approaches; however, they conclude that if the wrapped targeted classification model in the Wrapper approaches is a simple classification algorithm, the Wrapper Approach can be faster than Filter Approaches. They added that Wrapper approaches often achieve better classification performance than Filter approaches and feature subsets obtained from Wrapper approaches can be general to other classification algorithms.

7.2.2 Genetic Algorithms as Wrapper approach for optimising feature selection

Considering that the ML model performance can be affected not only by an individual feature but also by the combinations of two or more features in a feature set, in this research, we investigate improving the process of feature selection by applying automatic search techniques such as Evolutionary Algorithms. It should be noted from the literature that limited studies are available in applying Genetic Algorithms to solve the features selection optimisation problem for Relation Extraction. This has motivated this work to investigate the application of Genetic Algorithms as a wrapper approach for feature selection. Although this technique is computationally more demanding compared to Filter approaches feature selection, we argue that the computational overhead is not critical to our application and will not impact the performance of our Information Extraction system as

the Feature selection optimisation process is applied as a one-off process to optimise the performance of the machine learning classifies for each target problem domain.

Genetic Algorithms as a Wrapper approaches have been used to solve the features selection optimisation problem in diverse areas. For example, the work of Allami, et al. in (Allami, et al. 2016) propose a method to automatically detect and classify the Bundle Branch Block (BBB). BBB is a delay or obstruction along electrical impulse pathways in the heart. The automated detection and classification of a BBB is important for prompt, accurate diagnosis and treatment of heart conditions. Their method employed Artificial Neural Networks (ANN) algorithm for BBB classification and evaluated using the MIT-BIH Arrhythmia database. The MIT-BIH Arrhythmia dataset contains 48 ECGs of 30 minute recording lengths and two channel ambulatory ECG recordings in digital format with 360 Hz sampling rate. The features vector for the BBB classifier contains Nineteen temporal features and three morphological features were extracted and normalized for each heartbeat from standard ECG recordings obtained from the MIT-BIH Arrhythmia dataset. Genetic Algorithms as Wrapper approach for feature selection is used to improve classification accuracy. The GA as a wrapper improved the diagnostic accuracy by ignoring redundant and noisy features to determine the most significant features. They returned the improvement of the classifications accuracy to the capability of GA as Wrapper approach to integrate various optimal features subsets solutions to enhance the generality of the final features subset solution.

In an area close to Relation Extraction, Hasanuzzaman, et. al in (Hasanuzzaman, Saha and Ekbal 2011) propose an approach to search for the appropriate Machine Learning Classifiers' feature combinations for Named Entity Recognition. They used Maximum Entropy classifier and utilised Genetic Algorithm for feature subsets selection. The proposed approach is evaluated by using three different language datasets (Bengali, Hindi and Telugu). The evaluation results demonstrated the effectiveness of the proposed approach with acceptable overall recall, precision and F-measure values for the three languages.

7.3 Our Implementation of Genetic Algorithms for Feature Selection

As presented above, this research will investigate the application of Genetic Algorithms as a wrapper approach (the evaluation criterion) for feature selection. Genetic Algorithms are considered a favourable choice to solve many optimisation problems including the best features selection. Genetic Algorithms provide a powerful automatic heuristic search (the search strategy) for large, complex spaces (Sastry, Goldberg and Kendall 2014). We believe that the features in the solution space for Relation Classification are loosely related, which makes the utilisation of manual search techniques difficult. Hence, we automate the feature selection process by applying Genetic Algorithms search as a wrapper approach.

In the wrapper approach, the classifier model itself is employed to measure the fitness of features set; in other words, the features selection depends on the classifier model used (Karegowda, Jayaram and Manjunath 2010).

We have adopted the conventional implementation of Genetic Algorithms that generally comprises the initialisation of the solution space population, population reproduction including parent selection and replacement, crossover and mutation operations and defining the fitness function for evaluation (Sastry, Goldberg and Kendall 2014). However, there are several techniques to implement the aforementioned operations; for instance, there are two techniques for population reproduction, steady-state and generational populations and there are several methods for the population initialisation such as randomness, compositional and non-compositional. Similarly, parent selection can be performed using Stochastic Universal Sampling (SUS) or the Roulette Wheel Selection (RWS), and parent replacement can be based on the replacement of the worst parent or the replacement of random parents. The crossover operation could be applied to one or two crossover points in the chromosome and mutation operation could be applied on one or more genes in the chromosome (Kazimipour, Li and Qin 2014, Buzdalov, Yakupov and Stankevich 2015, Sastry, Goldberg and Kendall 2014). We conducted a series of experiments to heuristically determine which technique represents a better fit for our feature selection problem.

In our implementation, the genetic-information or chromosome is represented by a binary string of 1's and 0's (genes) that operate as a feature filter, where every bit or gene in the chromosome represents a certain feature. If the bit value equals one, this means that its feature is selected to participate in constructing the classifier model, otherwise the feature must be removed. The size of the features vector in this work is 20, which means that the size of the chromosome is 20 bits. Figure 7.1 shows the operation of the chromosome filtering.

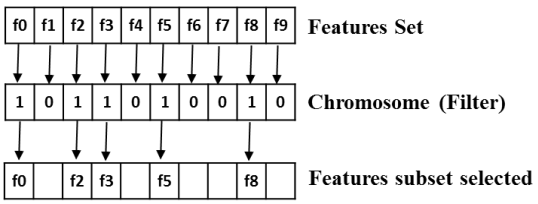

Figure 7.1: Chromosome features filtering

For the purpose of using GA as a wrapper approach, the ML classifiers are utilised to assess features' subsets according to their classification performance. In detail, we define the fitness function using the classification F1 score, which is computed by evaluating the relation classification model using k-fold Cross Validation. The fitness values are computed as follows:

1. By filtering a specified chromosome, a feature subset is generated to train the relation classification model.
2. The generated feature subset is evaluated by applying k-fold Cross Validation on the classification models with the targeted training dataset and feature subset as an input.
3. The resulting F1-score is assumed to be the fitness function value for the specified chromosome or feature subset.

Figure 7.2 below illustrates the flow of feature subsets selections as wrapper approach.

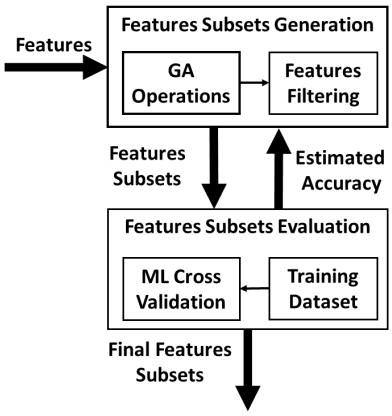

Figure 7.2: GA feature subsets selection as Wrapper Approach.

By means of experimentation, we heuristically selected the Roulette Wheel technique for parent strings selection, adopted two-points and all points for the crossover and mutation operations respectively. For population initialisation, we adopted randomness initialisation. There are two techniques for population reproduction, steady-state and generational techniques. We adopted the steady-state technique with the unconditional replacement of the worst chromosome for the parent replacement strategy because it is commonly used to assist in improving the performance of Genetic Algorithms. Steady-state technique is less computationally intensive than generational technique; for instance, for 20 population size and two parent selection and 50 iteration, it requires 120 fitness calls instead of 1100 fitness calls for generational technique (Lozano, Herrera and Cano 2005, Anu 2013).

Genetic Algorithms have their own parameters that require more experimentation to find the best fit for a specific optimisation problem. These parameters are, initial population size, the number of generations, crossover rate and mutation rate. These parameter values should be adjusted for each problem because they would be related to characteristics of the problem. Small population size might not provide a sufficient sample size for the search space in order to reach an optimum solution. On the other hand, a large population requires more evaluations per generation, which can result in a slow rate of convergence. The crossover rate controls the frequency of applying the crossover operator on the selected

parents to generate offspring. The higher the crossover rate, the more quickly new solutions are introduced into the population. If the crossover rate is too low, the search might be inactive due to the lower exploration rate. Similarly, the mutation rate controls the frequency of applying the mutation operator on the selected parents after applying crossover operator to increase the variability of the population. A low level of mutation rate serves to prevent any given gene position in the chromosome from converging to a single value in the entire population. A high level of mutation yields an essentially random search. Lastly, we needed to determine the optimal number of generations as it is directly related to the number of evaluations or fitness functions calls and hence impacts the efficiency of the Genetic Algorithms implementation (Mills, Filliben and Haines 2015). By means of experimentation, we heuristically established the parameters that represent the best fit for our feature selection problem. The values of the parameters are shown in Table 7.1.

Table 7.1: Our implementation of Genetic Algorithms Parameters

Parameters	Values and Types
The number of generations	100
The population size	20
The crossover rate	0.6
The mutation rate	0.05

The implemented Genetic Algorithm operation to select the best features subset is detailed in the following Pseudo-code:

1: Start:
2: N is the size of the population
3: Pc is the crossover rate and Pm is the mutation rate
4: Let the best solution be S^* and its fitness $F^*(S^*)$ equals 0
5: Generate initial N chromosomes C_i for the initial Population, where $i \in [0,1,\ldots,N)$
6: Evaluate the initial chromosomes C_i, $F(C_i)$;
7: repeat
8: Apply Roulette Wheel technique to select two parents' chromosomes, C_j and C_k, where $0 \leq j,k < N$ and $j \neq k$
9: Generating new chromosomes
10: Apply two points crossover operation on C_j and C_k chromosomes with probability Pc
11: Apply all points mutation operation on C_j and C_k chromosomes with probability Pm
12: Let the new chromosomes be C_j' and C_k', children's chromosomes
13: Evaluate C_j' and C_k', the fitness of the children's chromosomes are $F(C_j')$ and $F(C_k')$
14: Unconditionally replace the children's chromosomes C_j' and C_k' with the worst chromosomes in the population
15: Find the best chromosome C_b with best fitness $F(C_b)$ in the current population, where $0 \leq b < N$
16: Let the current solution S equals the best chromosome C_b and the current fitness F equals $F(C_b)$
17: **if** $F > F^*$ **then**
18: Update the best solution and the best fitness;
19: $S^* = S$;
20: $F^* = F$;
21: **end if**
22: **until** (stopping condition is met)
23: **Return** S^*, F^*
24: **End**

Our implementation of Genetic Algorithms' operations output is the chromosome that has best fitness value in the last generated population. The selected features of this chromosome is considered to be the best for the targeted classifier model.

More details about our evaluation results by conducting a set of experiments concerns feature selection by using Genetic Algorithms in a wrapper approach will be presented in the ensuing subsections. In the first section, we find the best features subset by using our implementation of Genetic Algorithms. In the second section, we evaluate the relation classification models using the selected feature subsets.

7.4 The Results of Genetic Algorithms Feature Selection

We ran our implementation of the Genetic Algorithm using the parameters shown in Table 7.1. Figure 7.3 below illustrates the required number of Genetic Algorithms' iterations required by SVM, PAUM and KNN to select an optimal fitness function value (F1 measure); SVM, PAUM and KNN require 57, 55 and 69 iterations respectively. We conclude that the three Machine Learning algorithms require approximately the same numbers of iterations to reach the optimal fitness value and that 100 iteration is quite sufficient for the Genetic Algorithm to achieve that goal.

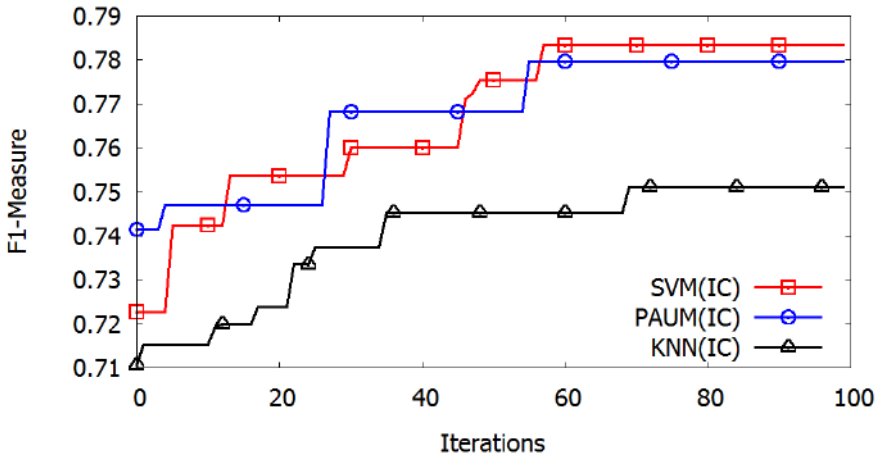

Figure 7.3: The Genetic Algorithm Iterations to select the best feature subset for Stock Index and the percentage increase or decrease training dataset by using SVM, PAUM and KNN ML algorithms.

Table 7.2 shows the number of selected features in every subset for every classifier, SVM, PAUM and KNN, in all training datasets. This table also shows the features in every subset, which are classified into the three categories, Lexical, syntactic and Named Entity category.

Table 7.2: The feature subsets that are selected by using Genetic Algorithms

TDS	ML	Feature Numbers			
		Lexical	Syntactic	Entity	Total
PerOrg	SVM	5	2	6	13
	PAUM	3	2	5	10
	KNN	1	0	5	6
PerLoc	SVM	4	1	7	12
	PAUM	1	2	7	10
	KNN	2	1	5	8
LocOrg	SVM	3	3	5	11
	PAUM	3	2	4	9
	KNN	5	4	4	13
StsOrg	SVM	2	2	2	6
	PAUM	2	5	3	10
	KNN	2	1	2	5
StiOrg	SVM	3	3	3	9
	PAUM	3	2	4	9
	KNN	5	3	1	9
OrgPct	SVM	3	3	5	11
	PAUM	2	5	6	13
	KNN	2	2	5	9
StiPct	SVM	1	3	4	8
	PAUM	2	3	5	10
	KNN	2	3	4	9
OrgDte	SVM	1	1	7	9
	PAUM	4	3	5	12
	KNN	3	2	2	7
StiDte	SVM	1	5	1	7
	PAUM	2	3	2	7
	KNN	-	4	3	7

From the data in Table 7.2 above, it is apparent that the features of the Named Entities category are selected more than the features of the lexical and syntactic categories in the majority of the training datasets. For the relation classifiers that are created by using SVM algorithm, the features of the Named Entities category are selected more than the other categories in 6 datasets (out of 9 datasets). For the relation classifiers that are created by using PAUM algorithm, the features of the Named Entities category are selected more than the other categories in 7 datasets (out of 9 datasets). However, for the relation classifiers that are created by using KNN algorithm, the features of the Named Entities category are selected more than the other categories in only 4 datasets (out of 9 datasets).

These results are consistent with the findings of Wang, et al. in (Wang, et al. 2006) that the entity features lead to improvement in performance because the mentioned relation between two entities is closely related to the entity types.

7.5 Evaluating the Relation Classification Models by using the Selected Feature subsets

The selected feature subsets in the training datasets are employed to create the relation classifiers' models. These models are evaluated by using 10-fold cross validation. Table 7.3 below shows the comparison between the Precision, Recall and F1-measures results of the three relation classifiers models, SVM, PAUM and KNN when the features vectors are optimised by using our implementation of GA on feature selection. Also, the table indicates the best F1-measure in terms of the best probability threshold value.

Table 7.3: Comparing the Classifiers results in terms of Precision, Recall and F1-measure in all training datasets with the optimised features Vectors by using our implementation of GA (Thr=Probability Threshold)

Entity Pairs Type	SVM				PAUM				KNN			
	Thr	P	R	F1	Thr	P	R	F1	Thr	P	R	F1
Per-Org	0.5	0.879	0.782	**0.825**	0.65	0.856	0.777	0.813	0.5	0.83	0.794	0.811
Per-Loc	0.4	0.773	0.741	**0.756**	0.65	0.79	0.717	0.751	0.7	0.752	0.713	0.732
Loc-Org	0.55	0.652	0.844	0.734	0.5	0.676	0.868	**0.758**	0.8	0.776	0.73	0.749
Sts-Org	0.5	0.853	0.897	**0.873**	0.5	0.846	0.897	0.869	0.5	0.85	0.911	0.877
Sti-Org	0.6	0.818	0.99	**0.89**	0.5	0.802	0.978	0.877	0.4	0.794	0.996	0.877
Org-Pct	0.15	0.672	0.672	**0.672**	0.15	0.699	0.634	0.665	0.8	0.698	0.6	0.644
Sti-Pct	0.4	0.773	0.773	0.773	0.5	0.783	0.778	**0.78**	0.5	0.762	0.762	0.762
OrgDte	0.1	0.63	0.63	0.63	0.5	0.658	0.627	**0.642**	0.55	0.617	0.614	0.615
StiDte	0.5	0.8	0.8	**0.8**	0.5	0.772	0.772	0.772	0.5	0.763	0.763	0.763

In addition, Figure 7.4 below indicates examples of a comparison between SVM relation classifier models when using all features and the selected feature subsets by our implementation of Genetic Algorithm in two training datasets (StockIndex-Organization and Person-Location) in terms of the relation between the probability threshold and F1-measure. It is clear that the F1-measure peaks upon probability threshold value (0.6) which is difference from the default (0.5) in StockIndex-Organization training dataset for both full features and selected subsets features. Also, in Person-Location training dataset, the F1-measure peak upon probability threshold value is (0.4) for selected set features and (0.45) for full features.

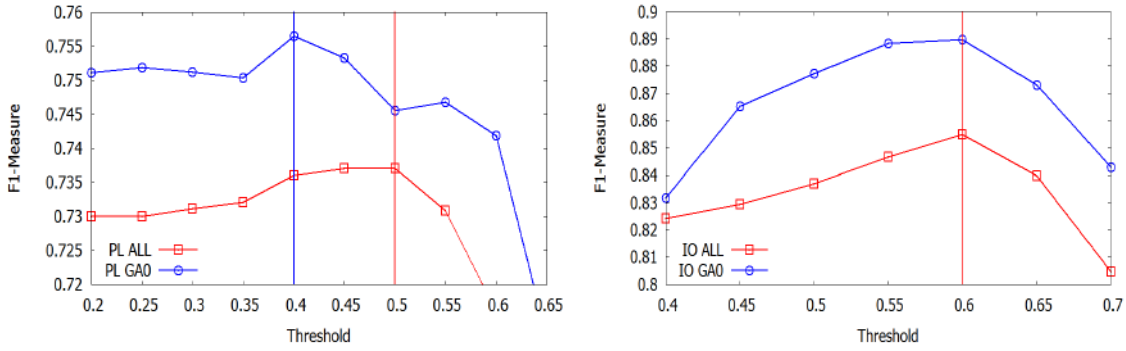

Figure 7.4: Indicates the comparison between SVM relation classifiers' models when using all features and feature subset selected by GA in Person-Location and StockIndex-Organization training datasets in terms of the best threshold.

From the data in Table 7.3 and the data in Table 6.7, we can see that all of the classifiers, SVM, PAUM and KNN, performed significantly better in the reduced feature space optimised by the Genetic Algorithms. As evidenced by data in Table 7.3, our implementation of Genetic Algorithms has improved the accuracy of ML algorithms in all training datasets. It can also be noticed that the improvements registered for SVM and PAUM are more evident compared to KNN. KNN is more sensitive to the irrelevant features, which is corroborated by Imandoust and Bolandraftar in (Imandoust and Bolandraftar 2013).

The comparison of the data in Table 7.3 and the data in Table 6.7 shows that PAUM algorithm outperforms SVM algorithm in 6 out of 9 datasets in the first table; however, SVM algorithm outperforms PAUM algorithm in 6 out of 9 datasets in the second table. It should be noted that the features vectors in the second table are optimised by using our implementation of Genetic Algorithms. This reflects the findings of Weston, et al. in (Weston, et al. 2001) that SVM suffer in high dimensional spaces where some features are irrelevant.

Our experiments have also indicated that the accuracy of the classification models is affected by the value of the probability threshold. The best threshold values for all classification models on all training datasets were empirically selected and were proven that delivering better classification accuracy can be achieved with the probability threshold values other than the default threshold value 0.5 (see Figure 7.4).

It can be observed from Table 7.2 that our implementation of Genetic Algorithm selects features from the Named Entity category more frequently than from the lexical and syntactic categories for the majority of the training datasets. Consequently, we decided to conduct further research to investigate the impact of the features categories on the classifiers' performance. Next section will present this investigation.

7.6 Features Category Selection

This section evaluates the effect of the features of a single category (Lexical, Syntactic or Named Entity) on the accuracy of the relation classification models. We created the models by using training datasets with features of each category alone and with features of all combinations of all categories. The models' evaluation results are compared in Table 7.4. The data in the table indicates that the best F1-measure values are produced when features of named entities category are included in the training in most of the training datasets.

Table 7.4: SVM, PAUM and SVM Classifiers with Categorised Features (FC=Features Category, L=Lexical Features, S=Syntactic Features, E=Named Entity Features, Thr=Probability Threshold, P=Precision, R=Recall, F1=F1 score)

TDS	SVM					PAUM					KNN				
	FC	P	R	F1	Thr	FC	P	R	F1	Thr	FC	P	R	F1	Thr
PerOrg	LE	0.905	0.752	0.819	0.55	LE	0.848	0.787	0.815	0.65	LE	0.823	0.779	0.8	0.75
PerLoc	E	0.762	0.727	0.744	0.4	SE	0.768	0.701	0.733	0.65	E	0.723	0.695	0.709	0.55
LocOrg	E	0.654	0.865	0.743	0.55	E	0.689	0.835	0.753	0.5	E	0.704	0.78	0.738	0.75
StsOrg	L	0.88	0.911	0.891	0.5	LS	0.849	0.911	0.876	0.5	LE	0.852	0.849	0.843	0.9
StiOrg	LE	0.811	0.941	0.866	0.65	LSE	0.799	0.979	0.877	0.5	SE	0.799	0.929	0.855	0.3
OrgPct	SE	0.696	0.642	0.667	0.4	SE	0.681	0.624	0.651	0.15	SE	0.616	0.612	0.614	0.5
StiPct	SE	0.692	0.692	0.692	0.5	LSE	0.73	0.721	0.727	0.5	LSE	0.728	0.686	0.705	0.5
OrgDte	LSE	0.674	0.629	0.638	0.4	LE	0.623	0.61	0.617	0.15	LE	0.591	0.559	0.581	0.65
StiDte	S	0.798	0.798	0.798	0.5	S	0.796	0.796	0.796	0.5	S	0.774	0.774	0774	0.5

Figure 7.5 below shows examples of the impact of features categories combination on the performance of SVM classifier when it is trained by using two training datasets instances, Person-Organization and Organization-Percent. In this figure, there is a clear trend of increasing in F1-measure when the SVM classifier is trained by training datasets includes named entity features category.

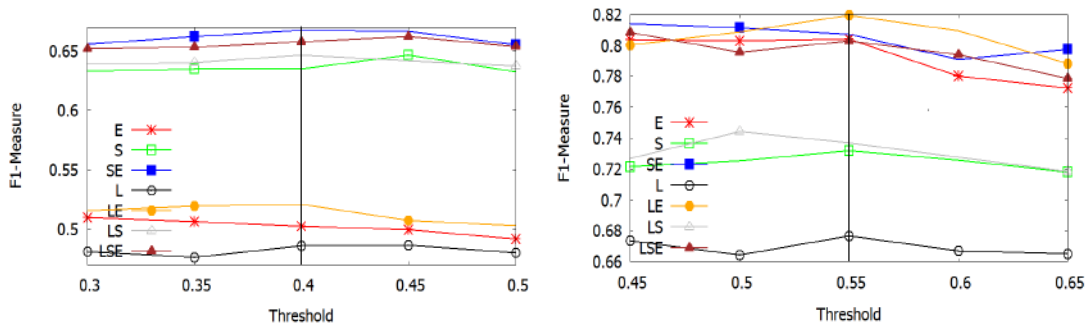

Figure 7.5: Examples of two training datasets to compare the features categories combination to train SVM classifier in terms of F1-measure and the probability threshold.

The results of these experiments illustrate that the models that are created using the Named Entity features combined with lexical and/or syntactic features, exhibit better accuracy than the models that are created without including the Named Entity category for most of training datasets of the tested ML classifiers. The exceptions are in the training datasets of the relation between Stock Symbol and Organization entities when used to train SVM, PAUM and KNN classifiers and between Stock Index and Date entities when used to train SVM and PAUM classifiers. To illustrate the impact of including the named entity features category in the StockSymbol-Organization and StockIndex-Date training datasets when training the ML classifiers SVM, PAUM and KNN, Table 7.5 below presents a comparison between all features categories combination for those training datasets.

Table 7.5: Comparison between the features categories compilations in SVM, PAUM and KNN classifiers with StockSymbol-Organization and Organization-Date training datasets in terms of F1-measure

FC	StockSymbol-Organization (F1)			Organization-Date (F1)		
	SVM	PAUM	KNN	SVM	PAUM	KNN
LSE	0.869	0.867	0.828	0.737	0.745	0.697
E	0.866	0.871	0.813	0.65	0.699	0.709
S	0.801	0.81	0.796	0.798	0.796	0.774
SE	0.878	0.853	0.814	0.771	0.747	0.736
L	0.891	0.862	0.836	0.658	0.69	0.663
LE	0.876	0.869	0.843	0.692	0.711	0.641
SL	0.88	0.876	0.841	0.748	0.777	0.698

Table 7.5 shows that the performance of the classifiers was insignificant declined compared to the performance of the classifiers when they were trained with lexical or syntactic or both category features. For example, in StockSymbol-Organization training dataset, the maximum F1-measure value for SVM classifier when using Entity Features is 0.878 and the maximum F1-measure value for SVM classifier when not using Entity Features is 0.891. The maximum F1-measure value for PAUM classifier when using Entity Features is 0.871 and the maximum F1-measure value for SVM classifier when not using Entity Features is 0.876. We can conclude that these results confirm the previous results of that the features of the Named Entities category are selected more than the features of the lexical and syntactic categories in the majority of the training datasets.

In general, the experimental results highlight the importance of a feature selection process. Furthermore, the classification accuracy of the ML models has improved as a result of deploying our implementation of Genetic Algorithms for optimising the feature selection process. Nevertheless, the performance of Genetic Algorithms in feature subset selection can be further enhanced by reducing the search space (Mills, Filliben and Haines 2015). The next section reports on our attempt to improve the performance of our implementation of Genetic Algorithms by grouping the features represented by the chromosome.

7.7 Reducing our Genetic Algorithm Search Space

Genetic Algorithms assist in solving the optimisation problem within a defined solution search space. The search for an optimal solution could be very complicated in a large search space, we here therefore attempt to reduce the search space by grouping similar and/or interrelated features. Additionally, we believe that representing the related features by a single gene could speed up finding the correlation between features and consequently the best feature subsets. Hence, in this investigation, we consider the possible associations between similar classes of relation classification's features, and analyse the features' frequencies, similarities and/or interrelations in order to reduce the search space for the feature subsets selection optimisation problem. This can be achieved by arranging the associated features into groups that can then be represented by a single gene in the Genetic Algorithms' chromosome. On this basis, we argue that this technique will reduce the Genetic Algorithms' search space and accelerate the process of best features combinations discovery and consequently improve the overall performance of the Genetic Algorithms.

We investigated categorising the features into groups by their similarities/interrelations and also by frequency analysis.

7.7.1 Grouping Features by their Similarities and Interrelations

We can observe from the Machine Learning features vectors listed in Table 6.4 that these features have similarities and interrelations that can be used to group them in order to be represented by one gene in the chromosome of our implementation of Genetic Algorithms. The features' similarities signify sharing the same form or structure such as the tokens' POS, the grammatical dependency paths and the tokens' strings. The features' interrelations represent how the features are relate to the candidate relation entity pairs such as the POS of the tokens before or after the entity pairs.

We have arranged three grouping types to cover most similarities and interrelations of features. In first grouping type we only grouped the features that can be paired. For example, the POS of the tokens before the first entity and the POS of the tokens after the second entity in one group and the first entity's type and the second entity's type in other group. The second grouping type is generated by adding features to groups in the first grouping type. For example, we added the entities' order in the relation instance feature to POS before and after features group. Also, we grouped more features which we believe that they have interrelation such as grouping the list of POS of tokens between entity pairs feature, the list of those tokens' strings feature and the number of tokens between the entity pairs feature. In the third grouping type, we have attempted to focus on the similarity of the features such as the list of tokens' strings between entity pairs and the list of dependency path nodes' strings between entity pairs. We also grouped the features that represent the

number of tokens and nodes between entity pairs. These grouping types 1, 2 and 3 are shown in Table 7.6.

Table 7.6: The Gene Representations of Features Groups type 1, 2 and 3

Gene No.	Grouping Type 1 (15 genes)	Grouping Type 2 (9 genes)	Grouping Type 3 (9 genes)
0	entityString1, entityString2	entityString1, entityString2	entityString1, entityString2
1	enttokensno1, enttokensno2	enttokensno1, enttokensno2	wordsStrSeq, dependencyWords
2	typeentity1, typeentity2	typeentity1, typeentity2	enttokensno1, enttokensno2
3	posentity1, posentity2	posentity1, posentity2	Distance, depDistance, directDep
4	posbefore, posafter	posbefore, posafter, order	typeentity1, typeentity2, order
5	poslist	wordsStrSeq, distance poslist, genposlist	posentity1, posentity2
6	genposlist	depDistance, dependencyKinds	poslist, genposlist
7	order	dependencyWords, directDep	Posbefore, posafter
8	wordsStrSeq	dependencyPath	dependencyPath, dependencyKinds
9	distance		
10	depDistance		
11	dependencyKinds		
12	dependencyWords		
13	directDep		
14	dependencyPaths		

7.7.2 Grouping by Features Frequency

We have analysed the features which are selected by the implemented Genetic Algorithm in all training datasets. Then. we have surveyed the features' frequencies in terms of their occurrences in the subsets selected. Table 7.7 below lists these features and the number of their occurrences in the subsets selected.

Table 7.7: Feature occurrence in the subsets selected by the Genetic Algorithm

Feature Number	Features	Occurrences
0	entityString1	12
1	entityString2	14
2	enttokensno1	15
3	enttokensno2	8
4	typeentity1	11
5	typeentity2	11
6	posentity1	15
7	posentity2	7
8	posbefore	5
9	posafter	9

10	poslist	13
11	genposlist	7
12	order	11
13	wordsStrSeq	13
14	distance	10
15	depDistance	6
16	dependencyKinds	6
17	dependencyWords	6
18	directDep	10
19	dependencyPath	11

After that, we grouped the features that have the same number of occurrence together as listed in Table 7.8 for group type 4.

Table 7.8: The Gene Representations of Features Group Type 4

Gene No.	Grouping Type 4 (11 Genes)
0	entityString1
1	entityString2
2	enttokensno1, posentity1
3	enttokensno2
4	typeentity1, typeentity2, order, dependencyPath
5	posentity2, genposlist
6	Posbefore
7	Posafter
8	poslist, wordsStrSeq
9	distance, directDep
10	depDistance, dependencyKinds, dependencyWord

Table 7.6 and Table 7.8 above show the genes representing the four grouping types. Every group is represented by a single gene, so the chromosomes' sizes are 15, 9, 9 and 11 for grouping types 1, 2, 3 and 4 respectively.

Our implementation of Genetic Algorithms was reapplied to select the best feature subsets according to those features grouping types and the experimental evaluation of the features' grouping is discussed in the next section.

7.7.3 Evaluation and Discussion

Following the same configuration for the initial feature selection optimisation described in section 6.6, we implemented and evaluated our attempt to further enhance that optimisation by reducing the search space (advanced grouping of features by various categories). Table 7.9 compares the results of both optimisation efforts in terms of the resultant relation classification F1-measure for all training datasets. The table also shows the type of features categories in the selected feature subsets. The categories are Lexical, Syntactic and Named Entity.

Table 7.9: A comparison between relation classifiers in terms of F1-Measure after applying GA for grouping feature selection (FG=Features Group Number, L=Lexical Features, S=Syntactic Features, E=Named Entity Features, T=Total Features, Thr=Probability Threshold, TDS=Training Datasets)

TDS	Features' Groups Results										All Features Chromosome Results				
	Model Accuracy					Features									
	ML	Thr	P	R	F1	FG	L	S	E	T	ML	Thr	P	R	F1
PerOrg	PAUM	0.5	0.872	0.787	0.826	1	1	3	8	12	SVM	0.5	0.879	0.782	0.825
PerLoc	SVM	0.55	0.823	0.695	0.753	4	2	3	5	10	SVM	0.4	0.773	0.741	0.756
LocOrg	KNN	0.9	0.828	0.738	0.777	2	2	3	6	11	PAUM	0.5	0.676	0.868	0.758
StsOrg	SVM	0.6	0.88	0.911	0.891	3	4	4	5	13	KNN	0.5	0.85	0.911	0.877
StiOrg	SVM	0.55	0.805	0.986	0.88	1	0	6	6	12	SVM	0.6	0.818	0.99	0.89
OrgPct	SVM	0.3	0.691	0.687	0.689	3	4	4	2	10	SVM	0.15	0.671	0.672	0.672
StiPct	PAUM	0.5	0.772	0.758	0.765	1	3	1	4	8	PAUM	0.5	0.783	0.778	0.78
OrgDte	KNN	0.7	0.657	0.622	0.639	4	3	3	4	10	PAUM	0.5	0.658	0.627	0.642
StiDte	SVM	0.5	0.808	0.808	0.808	2	0	2	0	2	SVM	0.5	0.8	0.8	0.8

The results listed in Table 7.9 are evidence that our implementation of Genetic Algorithms has, in general, improved the accuracy of all relation classifiers for all training datasets. However, the results in the table do not indicate obvious improvement in the relation classifiers' accuracy when we attempt to further improve the relation classifiers' accuracy by reducing the search space (grouping the features). In some training datasets, the accuracy of relation classifiers is higher when feature subsets are selected without grouping them. We attribute this to the small search space of our target use-case, which has less capacity for GA optimisation because Genetic Algorithms are probabilistic search procedures designed to work on large solutions spaces. In fact, reducing search spaces can have risk missing good results because the accuracy of the classification models not only depend on the quality of the individual features, but also on the best feature combinations (Goldberg and Holland 1988, Yong and Sannomiya 2001).

With respect to the performance of the SVM, PAUM and KNN relation classifiers, the data in Table 7.9 indicates that the accuracy of SVM classifier outperforms PAUM and KNN for the majority of the training datasets. The recorded results are consistent with the findings of other studies that utilise ML in relation classification; for example, the study by Li, et al. (2005) found that SVM may perform better than PAUM in small training datasets and they have a close performance in large training datasets. Also, the work of Hmeidi, Hawashin and El-Qawasmeh (2008) reveal that SVM has better F1-measure results than KNN. We believe that PAUM and KNN exhibit better performance than SVM in some training datasets because PAUM is appropriate for imbalanced training datasets and KNN performs better with small number of features.

In general, our findings evidence that our methodology for applying Genetic Algorithms for features selection improves the accuracy of Machine Learning based Relation Extraction.

In the next section, we further assert this claim by comparing it against another solution search method for feature subsets selection.

7.8 A Comparison between Random Mutation Hill-Climbing and Genetic Algorithms

In this section, we attempt to verify that Genetic Algorithms are an appropriate choice for optimising the process of feature subsets selection for the relation classification problem. Hence, we decided to compare our implementation of Genetic Algorithms with Random Mutation Hill-Climbing (RMHC) as their operational dynamics are very similar but simpler.

Since the early times of developing GAs and RMHC algorithm, several studies investigating the comparison between them have been carried out on different problems. For example, Mitchell and Holland in (Mitchell, Holland and Stephanie 1994) who were attempted to answer the question: when will a Genetic Algorithm outperform Hill-Climbing? They claim that understanding the mechanism of Genetic Algorithms and the characteristic of the fitness landscapes of the problem is crucial for deciding when the Genetic Algorithms will be most useful.

Another study by MacFarlane, et al. in (MacFarlane, et al. 2010) compared between Genetic Algorithms and several types of Hill-Climber algorithms including RMHC. The algorithms were applied to solve term selection problem for an information filtering task. Although they observed that both Genetic and Hill-Climbing algorithms appear to be able to improve accuracy of term selection, they did not find evidence that their implementation of Genetic Algorithm has better performance than their implementation of Hill-Climbing algorithm.

In a completely different problem, the authors in (Sakamoto, et al. 2014) compare Hill-Climbing (HC), Simulated Annealing (SA) and Genetic Algorithm (GA) by simulating the node placements problem for achieving the network connectivity and user coverage. Their aim was to find which algorithm, HC, SA and GA, assists achieving the optimal distribution of router nodes, provides the best network connectivity and provides the best coverage in a set of randomly distributed clients. From the simulation results, all algorithms converge to the maximum size of Giant Component; however, HC and SA converge faster according to the number of covered mesh clients.

We believe that our choice of comparing RMHC and GA in features selection optimisation problem for relation classification is consistent with the above presented works that compare HC and GA in other diversity of problems.

RMHC can be considered as a Genetic Algorithm without crossover operation and initial population. The solution neighbour or the new solution in RMHC can be generated by applying a similar mutation operation as in Genetic Algorithms, which could make jumps of varying sizes through the search space (Sastry, Goldberg and Kendall 2014). The other reason of choosing RMHC to compare with our implementation of GAs is to compare

between the complexity of GA with the simplicity of RMHC and answering the question: do we need the computational complexity of GA operations?

In our RMHC implementation, we adopted a similar configuration to that used by Sakamoto, et al. in (Sakamoto, et al. 2014). The RMHC implementation works as in the following Pseudo-code:

```
Start
Generate an initial solution S0;
Evaluate the initial solution S0, F(S0);
Let the current solution S equals the initial solution S0;
Let the best solution S* equals the initial solution S0;
Let the best fitness value F* equals the fitness of the initial solution F(S0);
repeat
    Mutate the current solution S to generate a new solution S';
    Evaluate the new solution F(S');
    if F(S') > F(S*) then
        Update the best solution and the best fitness;
        S*=S' ;
        F* = F(S') ;
    end if
    Update the current solution S = S' ;
until (stopping condition is met)
Return S*, F*
End
```

In order to fairly compare the performance of our implementation of Genetic Algorithms and RMHC for features selection problem, the experiments should be under the same computational conditions, in particular with respect to the fitness evaluation calls as it represents the most critical operational step of search algorithms. It is clear that one run of Genetic Algorithms is more expensive than one run of RMHC in terms of fitness functions calls (Acampora, Pedrycz and Vitiello 2015). As a result, we should run both algorithms with equal number of fitness function calls.

Because we adopted the steady state technique for population reproduction in our implementation of GAs, the number of fitness function calls will be equal to $I \times 2 + P$, where, I is the iterations number of GAs' operations and P is the population size. However, the number of fitness function calls in RMHC is equal to the number iterations of its operations because our implementation of RMHC does not have initial population. Consequently, the number of iterations of RMHC experiments should be equal to the number of our GA fitness function calls.

For the purpose of this experimental comparison, we evaluate optimising the accuracy of the SVM relation classifier on only one training dataset (Location-Organization). The number of iterations in our implementation of the Genetic Algorithms is 50, thus the algorithm makes 120 fitness function calls for a population size of 20; consequently, the Random Mutation Hill-Climbing algorithm should have 120 iterations in order to subject it to the same computational efforts in terms of fitness evaluations. The number of executed

runs for each algorithm is 30, which represent the number of sample runs. The comparison between our implementation of Genetic and Random Mutation Hill-Climbing algorithms are highlighted in the line chart of Figure 7.6 in terms of fitness sample runs, i.e. F1-measure. The results in the figure indicates that Random Mutation Hill-Climbing algorithm outperforms our implementation of Genetic Algorithms in only 4 of the 30 sample runs.

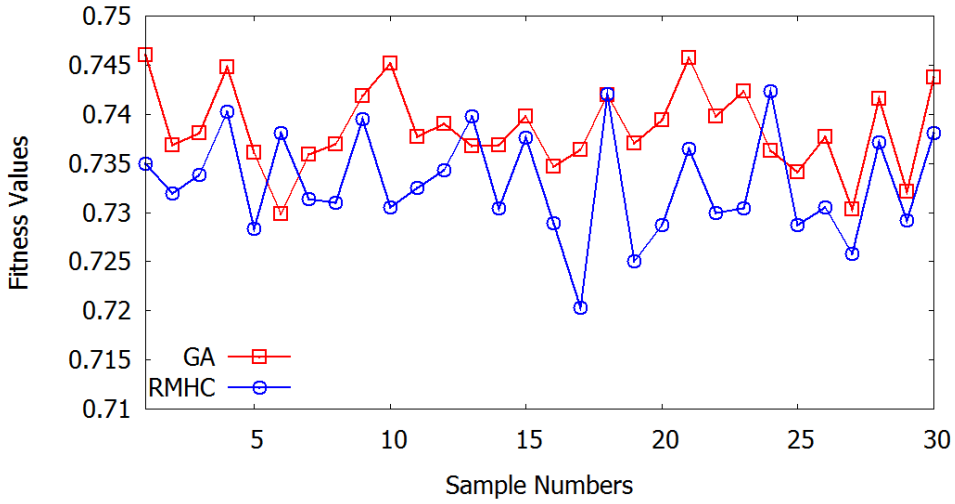

Figure 7.6: GA and RMHC Samples Comparison.

From the data in the line chart in Figure 7.6 above, it is apparent that our implementation of Genetic Algorithms outperforms Random Mutation Hill-Climbing algorithm in the majority of the results' sample runs as our implementation of Genetic Algorithms have higher ranking sample runs than the sample runs of Random Mutation Hill-Climbing algorithm. Nevertheless, in order to further examine any significant difference in the performance of our implementation Genetic Algorithms and Random Mutation Hill-Climbing algorithm, we applied a statistical test to compare their performance in the feature subset selection problem. We considered a Wilcoxon singed rank test procedure to perform a pairwise comparison between the two algorithms' sample runs. Wilcoxon test is a non-parametric statistical procedure for examining the median differences in observations for two samples. It aims to detect if there is a significant difference among the behaviour of the samples of two algorithms' results. Before applying the Wilcoxon procedure test, we should rank the absolute differences of the two sample pairs. First, finding out the difference between each sample pair. Then, the absolute differences of the samples are ranked by ordering them from the smallest to the largest. The rank will be according to the position of the absolute difference of the pair in the ordered list (García, et al. 2009). Table 7.10 shows the fitness values for the sample runs of Genetic and Random Mutation Hill-Climbing algorithms; also,

their paired sample runs differences and the ranks and total ranks of their absolute differences.

Table 7.10: GA and RMHC F1-measure sample runs and their absolute differences ranks

Sample Run #	GA F1-Measure	RMHC F1-Measure	Difference	GA Ranks	RMHC Ranks
1	0.74602175	0.73499995	0.0110218	26	
2	0.7368624	0.7319737	0.0048887	12	
3	0.738097	0.73382115	0.00427585	6	
4	0.7448637	0.7402726	0.0045911	10	
5	0.7361086	0.728381	0.0077276	21	
6	0.7298968	0.73811346	-0.00821666		22
7	0.73591727	0.73139066	0.00452661	8	
8	0.7370021	0.73098475	0.00601735	17	
9	0.74191993	0.73949844	0.00242149	3	
10	0.7452387	0.7305558	0.0146829	29	
11	0.73776346	0.7325595	0.00520396	13	
12	0.73907685	0.7343243	0.00475255	11	
13	0.73682123	0.7398594	-0.00303817		5
14	0.7368653	0.7304085	0.0064568	19	
15	0.7397724	0.73760575	0.00216665	2	
16	0.73471147	0.72893906	0.00577241	16	
17	0.73643947	0.7203119	0.01612757	30	
18	0.7419509	0.7420638	-0.0001129		1
19	0.7370386	0.72499377	0.01204483	28	
20	0.7394399	0.7287488	0.0106911	25	
21	0.7457602	0.7364889	0.0092713	23	
22	0.73983675	0.7299845	0.00985225	24	
23	0.7423382	0.73042387	0.01191433	27	
24	0.7362633	0.7423339	-0.0060706		18
25	0.73413545	0.728746	0.00538945	14	
26	0.7377205	0.7304985	0.007222	20	
27	0.73034245	0.72577304	0.00456941	9	
28	0.74158335	0.7371815	0.00440185	7	
29	0.73213834	0.72924286	0.00289548	4	
30	0.74381757	0.7381317	0.00568587	15	
Total Ranks:				419	46

The Wilcoxon singed rank statistical analysis was applied by using the R package (R 2018) on our implementation of Genetic Algorithms and Random Mutation Hill-Climbing algorithm sample runs under the null hypothesis and at 0.05 significant level (α).

The Wilcoxon test results in R package are shown in below:

data: GA and RMHC
V = 419, p-value = 0.00003453
alternative hypothesis: true location shift is not equal to 0

Where *V* is the sum of the positive ranks (GA results ranks) and p-value is a probability that measures the evidence against the null hypothesis. Lower probabilities provide stronger evidence against the null hypothesis.

It is clear that p-value (0.00003453) is considerably less than the significant level (0.05). This result shows that there is a significant difference between our implementation of Genetic Algorithms and Random Mutation Hill-Climbing algorithm and the null hypothesis is rejected. The statistical test result further evidences that the our implementation of Genetic Algorithms for feature selection outperforms the Random Mutation Hill-Climbing algorithm in terms of improving relation classifiers accuracy.

7.9 Summary

In this chapter, we employed GAs as wrapper approach to optimise the process of features selection in order to reduce the dimensionality of the data and subsequently increases the efficiency and accuracy of the classifiers' operations. GAs has been widely used as a Wrapper approach for features selection with favourable results. To the best of our knowledge, there is no reported work thus far on using GAs for relation classification's features selection.

Because, the configuration parameters of GAs require tuning to find the best fit for a specific optimisation problem, we heuristically established by means of experimentation the optimum values for the GA's initial population size, the number of generations, crossover rate and mutation rate that represent the best fit for our features selection problem for relation classification.

In terms of selecting the best features for relation classification, the research findings indicate that the models that are created using the Named Entity category combined with lexical and/or syntactic features, exhibit better accuracy than the models that are created without including the Named Entity category for most of training datasets of the tested ML classifiers. We can conclude that these results confirm the previous results of that the features of the Named Entities category are selected more than the features of the lexical and syntactic categories in the majority of the training datasets.

Due to the modest search space of the features that are generated from the characteristic of the target domain, our attempt to further improve the performance of the GA by reducing their search space through features grouping did not result in a significant improvement. However, we believe that exploring the similarities and interrelations between features could yield better results for other domains with larger search space and different feature types.

The conducted experiments evidenced that the developed knowledge-assisted ML relation classification model, which was further boosted by our implementation of GAs to reduce the feature space, has resulted in significant improvement in the process of relation extraction.

The experimental results also indicate that amongst the implemented ML algorithms, SVM exhibited the best relation classification accuracy in the majority of the training datasets while retaining acceptable levels of accuracy in the rest in the remaining training datasets.

Finally, we verified that GAs are an appropriate choice for optimising the process of features selection for the relation classification problem by comparing them against a space search algorithm that has similar but simpler operational dynamics, Random Mutation Hill-Climbing (RMHC) by using a non-parametric statistical procedure, Wilcoxon test. The findings demonstrated that the performance of the two algorithms is comparable with the fact that our implementation of GAs is preferable in most instances.

The lessons learned from the findings of applying Genetic Algorithms as a wrapper approach to optimise the features selection in Relation Classification problem could be summarised as follows:

- It is important to perform the feature selection process for improving the accuracy of the Machine Learned based Relation Classification.
- The solution space of features selection in Relation Classification problem are loosely related, which makes the utilisation of manual search techniques difficult.
- Genetic Algorithms provide a powerful automatic heuristic search for large, complex spaces than small solution spaces.
- Although Genetic Algorithms as a wrapper approach is computationally more demanding, this is not critical to our application and will not impact the performance of our Information Extraction system as it is a one-off process.
- Reducing search spaces for Genetic Algorithms can have risk missing good results because the accuracy of the classification models not only depend on the quality of the individual features, but also on the best feature combinations.
- Exploring the similarities and interrelations between features could yield better results for other domains with larger search space and different feature types.

8 Constructing and Exploiting the Semantic Knowledgebase

8.1 Introduction

As the published volume of Semantic Web data is increasingly growing, the question of how typical web users can access this body of this heterogeneous knowledge becomes of crucial importance. There is a growing amount of research on interaction paradigms that allow end users to benefit from the expressivity of Semantic Web standards and their facility to intelligently explore information. Furthermore, the interest in adopting Semantic Web Technologies for knowledgebase representation and exploitation is increasing as they are capable of supporting advanced data exploration and decision-making use-case scenarios (Marie and Gandon 2014, Fafalios and Tzitzikas 2013).

The previous chapters of this thesis have described the core technologies that underpin our efforts into the knowledge representation and Information Extraction. These technologies (Semantic Web and Machine Learning) have been utilised in information extracting and semantic modelling the problem domain knowledge. The developed domain model (or ontology) is employed as a unifying structure to describe a common representation for semantics of the extracted information. In this chapter, we will describe how to utilise these technologies 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. The main task of knowledge-based applications is the inference task on the semantic knowledgebase because it draws conclusions from that knowledgebase. The inference mechanism can be achieved by utilising the Semantic Web Technologies in the knowledge-based applications to solve complex problems and to provide effective decision support (Davies, Studer and Warren 2006).

In this chapter, we will present in detail the construction of the semantic knowledgebase. Then, we will review the application of Semantic Web Technologies in two aspects of accessing that knowledge, supporting the decision-making process and semantic knowledgebase exploration.

8.2 Constructing the Semantic Knowledgebase

The process of constructing the knowledgebase will go through three stages, Information Extraction, ontology population and knowledgebase enriching. In the stage of Information Extraction, the unlabelled unstructured documents are taken as an input and the out of this

stage is annotated documents with named entities and their interrelations. The ontology population stage, the annotated documents inserted into the semantic knowledgebase. This semantic knowledgebase is enriched by using external sourced semi-structured and structured data source to produce the final semantic Knowledgebase.

The implementation of these stages have been achieved by using open source tools, Protégé, GATE and JENA. Protégé tool was used to build and edit the ontology, the GATE tool has been used in applying Natural Language processes tasks to extract information and Jena tool is used in populating the semantic knowledgebase. Building the ontology was based on the concept map of the problem domain knowledge. It contains a formal model that defines concepts and their relations in standard languages, RDF, RDFS and OWL. The targeted named entities and their interrelations are listed in Table 4.4.

Online data in majority of domains is often subjected to change and evolve over time due to the dynamic nature of knowledge. For example, new facts are becoming known while some of the older ones need to be revised and/or retracted at the same time. This evolution should be addressed by adding new facts to the knowledgebase (Toledo, Chiotti and Galli 2012, Nováček, et al. 2008). Although this issue is beyond the scope of this study, it is worth to mention that the Information Extraction component of the proposed Semantic Web application should be designed to be dynamic in terms of the ability of making the semantic Knowledgebase constantly up to date. This update can be achieved periodically or on demand regarding the application consumers requests because the economic and finance domain is a dynamic domain and causes a rapid information influx (Li, et al. 2014a). The process of decision-making support and knowledge exploration should be achieved regarding a newly extracted information.

The details of constructing the semantic knowledgebase stages and their implementation are presented in Figure 8.1 below and explained in next three subsections.

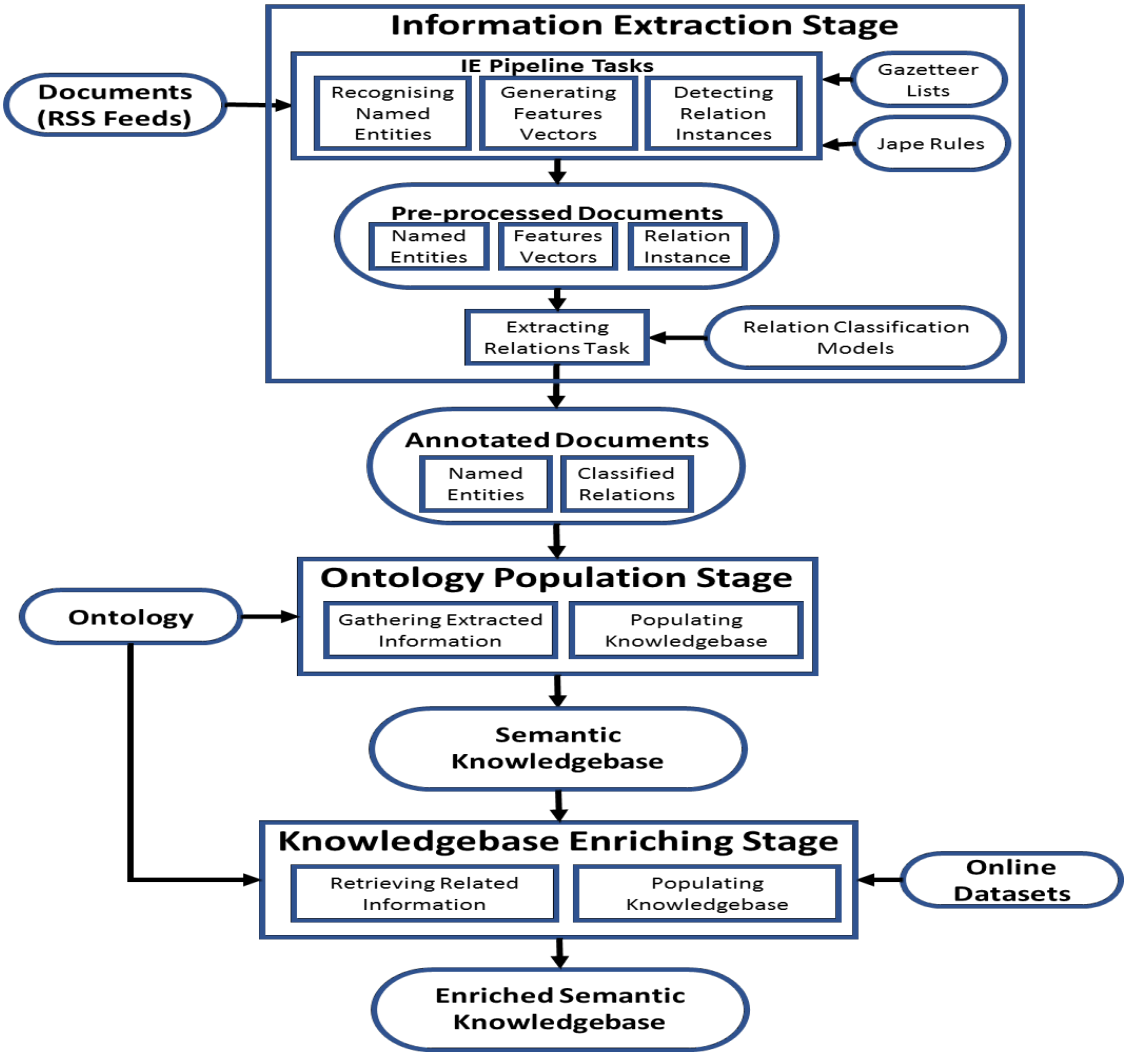

Figure 8.1: The process of Constructing Semantic Knowledgebase Stages

8.2.1 Information Extraction Stage

As we mentioned in chapter 5, the main tasks of Information Extraction pipeline process in this research are recognising the named entities, detecting the relation instances and generating features vectors. These tasks have two objectives, composing the training datasets for training the relation classifiers to create the classification model, and preparing the unlabelled online news texts ready for applying these classification models to extract relations between the targeted named entities. These named entities and their interrelations are mapped to the ontology as instances and properties in the semantic knowledgebase.

As aforementioned in chapters 5, 6, 7, we conducted a serious of experiments that utilise the ML algorithms to report on the favourable implementations/configuration for successful Information Extraction for our targeted domain. The results of these experiments indicate that the classifiers which are created by using SVM algorithm outperform PAUM and KNN algorithms in the majority of training datasets. As a result, we employed SVM algorithm to create the relation classification models to be applied on unlabelled data.

We applied these models onto the pre-processed unlabelled relation instances to extract new relations between the annotated entities. The output data of this step is a corpus of annotated documents with named entities and their interrelations.

Table 8.1 below shows an example of a sentence from RSS feeds news article and all its related details which is retrieved from CNN Money source.

Table 8.1: Example of a sentence from RSS Feeds news Article

Source	CNN Money
Article Title	Microsoft CEO celebrates 'pivotal year' as Office 365, cloud make gains
URL Link	http://www.cnbc.com/2016/07/19/microsoft-ceo-satya-nadella-celebrates-pivotal-year-as-office-365-cloud-make-gains.html
Creator	Harriet Taylor
Article Date	19 July 2016
Sentence	Microsoft stock traded up as much as 4.25 percent to $55.34 in after-hours trading on Tuesday after the company easily topped analysts' expectations

After applying the Named Entities Recognition pipeline, there are three named entities can be recognised and two relation instances can be detected in this sentence. In addition, the ML features are generated to be used to classify the relation instances. There are two relation classes predicted after applying the Relation Classifiers. Table 8.2 below shows these named entities, relation instances and the predicted relation classes.

Table 8.2: Example of Recognising Named Entities and Extracting Relations Between them

Relation Instance	Entity 1	Entity 2	Predicted Relation Classes	confidence scores
Microsoft stock traded up as much as 4.25 percent	Microsoft (Organization)	4.25 percent (percentage)	shareIncreasedBy	0.78769803
Microsoft stock traded up as much as 4.25 percent to $55.34 in after-hours trading on Tuesday	Microsoft (Organization)	Tuesday (Date)	shareIncreaseDate	0.60409284

This table also shows confidence scores of extracted relations. These confidence scores are based on the probability of the correctness of entity pairs' relation. Confidence score refers to a classification model's estimate of the probability that a potential relation instance is a correct relation. These annotations representing the named entities and their

interrelations. They will be inserted into the Knowledgebase using the ontology as explained in next section.

8.2.2 Ontology Population Stage

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. In addition, it enriches the ontology that can be used for a variety of exploration scenarios to provide aid in a specific subject matter (Du and Zhou 2012, Lupiani-Ruiz, et al. 2011, Maynard, Li and Peters 2008).

The annotated named entities and their interrelations in the documents are transferred to a the Semantic Web RDF model by using our Semantic Web ontology. The named entities are related to an appropriate concepts as instances in the semantic Knowledgebase. Then, we mapped the relations between those named entities to the suitable property, data type or object, in the ontology as relation instances in the semantic Knowledgebase.

Returning to the provided example in the previous section and presented in Table 8.1 and Table 8.2, the classified and annotated relations between the named entities can be mapped into RDF triple as below:

"Microsoft → shareIncreasedBy → 4.25 percent"

"Microsoft → sharesIncreaseDate → Tuesday"

In the subsection 4.4.2 in chapter 4, we mentioned that we have adopted N-ary relation pattern to represent our domain-specific non-binary relations in the resultant semantic knowledgebase. This is because characteristics of our targeted domain is heavily represented by non-binary relations. We applied relation-centred or relation-as-class pattern for N-ary relations to implement all triples that represent the entities and their relations besides all details related to them in the semantic knowledgebase such as the article's details. We believe that providing the information about data source such as author, title, date and URL link is critical for end users because they increase the reliability on the information. In addition, the confidence scores values of the predicted relations that could be used to rank the extracted relations to generate a list of the most confident relations (Mintz, et al. 2009).

Examples of the triples which are inserted into the semantic knowledgebase are:

kbfwo:microsoft → rdf:type → kbfwo:Company

kbfwo:microsoft → rdfs:label → "Microsoft"

kbfwo:microsoft → kbfwo:sharePriceChange → kbfwo:sharepricechange_1

kbfwo:sharepricechange_1→ rdf:type → kbfwo:SharePriceChange

kbfwo:sharepricechange_1 → kbfwo:shareIncreasedBy → “4.25 percent”^^xsd:String

kbfwo:sharepricechange_1 →kbfwo:hasDate → kbfwo:tuesday_1

kbfwo:weekday_1 →rdf:type → kbfwo:Date

The complete graph of the triples above are depicted in Figure 8.2 below.

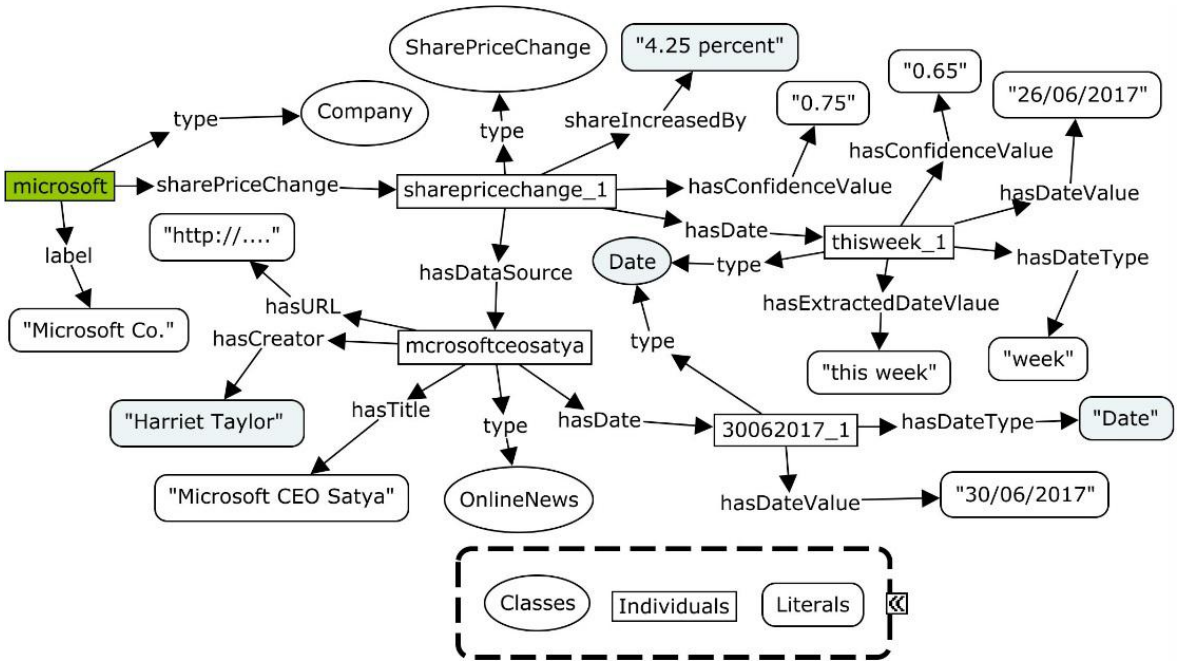

Figure 8.2: Populated Triples Graph Example

It should be noted from the above figure that the data type of literals are not included in the this triple graph. For example, the type of “0.75” should be “^^xsd:float”, the type of “Volatility returns” should be “^^xsd:string” and the type of “30/06/2017” should be “^^xsd:dateTime”. In addition, the date of share price change “this week” is mapped to a real date of the first day of that week by referencing it to the date of article. During this process, the “rdfs:label” annotations (they are only “label” in the figure above) are related to the instances and resources to provide a human-readable version of resources’ names.

8.2.3 Enriching the Semantic Knowledgebase by Sourcing Online Datasets

Ontology Population occurs in both knowledgebase that has no instances and those that have already has been populated. When ontology population is performed on those that already have instances the process is known as enriching the knowledgebase. 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 (see Table 4.2).

For instance, the example sentence illustrated in Table 8.1, Table 8.2 and Figure 8.2 is enriched by adding more information about the company and its country by using Crunchbase dataset, Yahoo Finance web service API and World Bank Linked Data endpoint. We detail below the methodology for exploiting these data sources.

1- Crunchbase dataset

This dataset is provided by Crunchbase Incorporation. It contains information about hundreds of thousands of public and private companies globally. The ground facts, which are retrieved from Crunchbase dataset, are about companies including information about industry sectors, founders, employees, products, location and stock symbols.

The ground facts that are retrieved from Crunchbase dataset about companies including information about industry sectors, founders, employees, products, location and stock symbols. According to CrunchBase developers, the data is unique, due to the engaged community of users who update company and team profiles. They validate funding data with the venture community through the CrunchBase Venture program. For the profile of the companies who are actively fundraising, they partner with AngelList, EquityNet, AgFunder and others. By actively tracking RSS and Twitter, their team keeps on top of all recent fundings profile in the CrunchBase Daily. All of these programs, over time, build out the connections within the global entrepreneurial community.

Crunchbase is available through a REST API under the CreativeCommons license. There are different kinds of licenses available. One of these licenses is the academic research license. It is a limited access license to check out the Open Data Map and explore the 2013 snapshot. However, we have used the linked data version of this dataset that can be downloaded as a RDF data dump file. It is available in this link:

http://km.aifb.kit.edu/sites/crunchbase/crunchbase-dump-201510.nt.gz

This RDF data dump has been constructed by Färber, et al. in (Färber, Menne and Harth 2017)

To retrieve these ground facts we applied SPARQL queries on the RDF Crunchbase dataset. Below is an example of one of these queries.

```
SELECT DISTINCT ?name ?proName
WHERE {
        ?OrganizationUri a cb:Organization;
                cb:name ?name;
                  cb:stock_symbol ?StockSymbol;
               cb:products ?Products.
        OPTIONAL {?Products a cb:Product;
                    cb:name ?proName}
        FILTER (!isBlank(?OrganizationUri))
        FILTER (str(?StockSymbol) = ?entity)
}
```

This query is for retrieving the companies' products where "cb" is the prefix of the main URI reference of the names used in Crunchbase dataset.

We mapped these triples to N-ary pattern adopted in this research and explained in pervious sections. The graph example of these triples is shown in Figure 8.3 below.

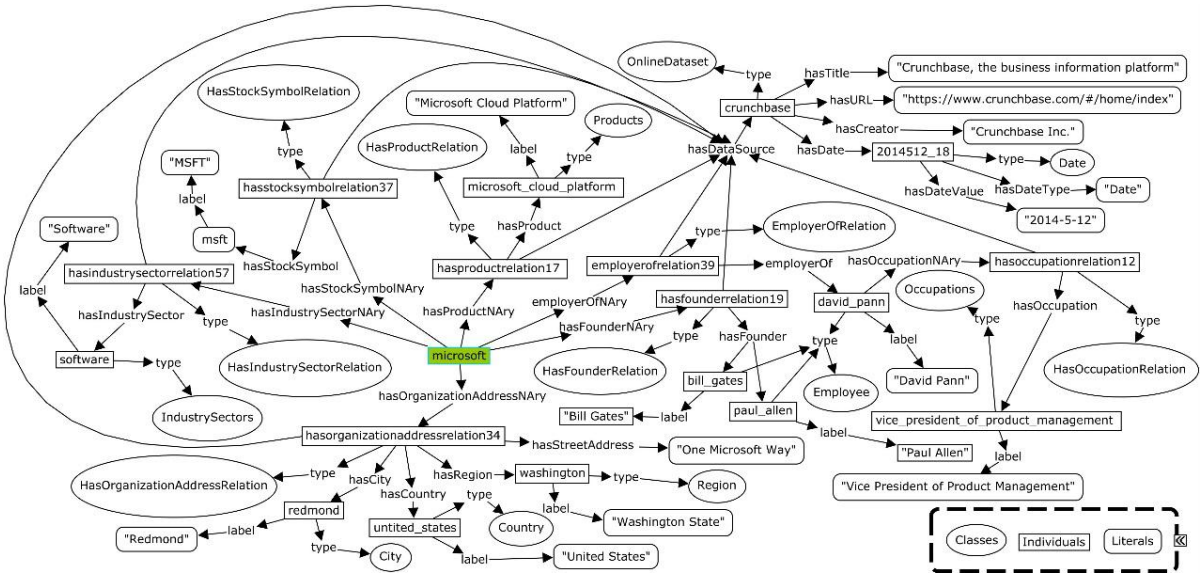

Figure 8.3: Enriching Triples Example by using Crunchbase dataset

2- Yahoo Finance API

Yahoo Finance API provides a simple method to retrieve free stock quotes. They are including information about stock prices, dividend payments, dividend yield percentage and the history of these information. Yahoo Finance has numerous partners that provide quote data. Some of these partners provide real time data such as NASDAQ and NYSE stock exchange. Static quotes and other information, such as historical data, are supplied by independent providers. For example, the fundamental company data provided by Capital

IQ. According to Yahoo Finance support team, the data sources cover a wide range of markets and indexes around the world such as Caracas Stock Exchange in Venezuela, Borsa Istanbul in Turkey, Singapore Stock Exchange in Singapore, Qatar Stock Exchange in Qatar, Mexico Stock Exchange in Mexico, Athens Stock Exchange in Greece and London Stock Exchange in United Kingdom. All information provided by Yahoo Finance is for informational purposes only, it is not intended for trading purposes or advice. When accessing the Yahoo! Finance API, it is agreed to not redistribute the information retrieved from it.

This Yahoo service returns stock data in a Comma Separated Values (CSV) file. This file can be processed manually by using editing tools such as MS Excel application or automatically by using programming languages such as JAVA. The Yahoo finance API is a REST based service. It provides a URL to be assembled with the required parameters for every piece of information related to the targeted company stock. The base URL for this service is:

"http://download.finance.yahoo.com/d/quotes.csv"

Where "quotes.csv" is the returned stock data file.

These ground facts require adding a specific symbols of the targeted stocks and specific parameters for the type of information about that stock symbol to be retrieved by using the base URL. For example, the symbol (?s) for specifying the stock symbols name and the (?f) for specifying the parameters of required stock information. Below is an example for a base URL with symbols and parameters:

http://download.finance.yahoo.com/d/quotes.csv?s=MSFT&f=npd1dr1";

where:

?s=MSFT means that data retrieved is about the stock of Microsoft "MSFT"

?f=npd1dr1 is representing the parameters of the required data about the targeted stock. Where "n" is for company name, "p" is for stock price, "d1" is for last trade date, "d" is for dividend per share and "r1" is for the dividend payment date.

The complete list of these symbols and parameters can be found in this link:

http://www.jarloo.com/yahoo_finance/

Below is an example of how the triples of the stock price of a specific company is added to the semantic knowledgebase.

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
kbfwo:hassharepricerelation_1 → kbfwo:hasDataSource →

kbfwo:yahoo_finance_api_service_to_return_stock_data

Then, the retrieved information from the Yahoo Finance API are mapped to N-ray relation pattern triples and inserted into the semantic knowledgebase. The graph example of these triples is shown in Figure 8.4 below.

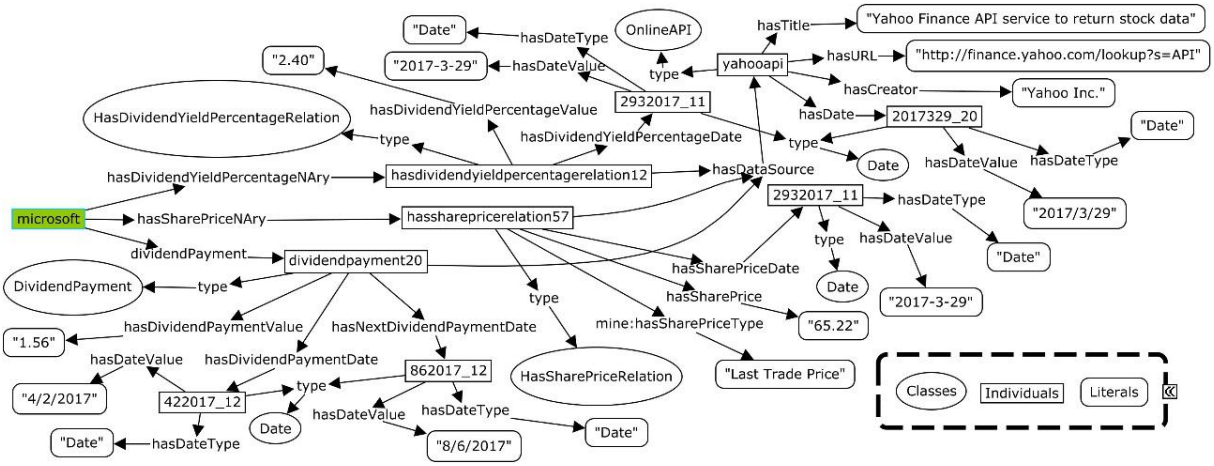

Figure 8.4:Enriching Triples Example by using Yahoo Finance API

3- World Bank Open Data

The World Bank is a vital source of financial and technical assistance to developing countries around the world. They offer a growing range of free, easy-to-access tools, research and knowledge to help people and countries address the world's development challenges. For example, the Open Data website offers free access to comprehensive, downloadable indicators about development in countries around the globe such as GDP, inflation and unemployment. World Bank provides a diversity of methods to access this data such as REST based Web API, downloading excel format files and SPARQL endpoint. We have used the REST PPI of World Bank data website to access the required data directly. Below is an example of the structures of URL based API query:

http://api.worldbank.org/countries/US/indicators/SL.UEM.TOTL.NE.ZS?MRV=1&format=xml

where “US” is country code of United States of America, “SL.UEM.TOTL.NE.ZS” is the employment rate indicator, “MSV” is to fetch most recent values based on the number specified with this parameter (=1) and “xml” is to return data in a file of XML format. This API query will return the most recent value of the unemployment rate indicator of United States and the data will be returned in XML file. This file can be processed pragmatically to retrieved the required information. More information about World Bank Data API can be found in this link:

https://datahelpdesk.worldbank.org/knowledgebase/articles/889392-api-documentation

Also, the World Bank data is published using the Linked Data design principles, Semantic Web Technologies. This linked data is collected from World Bank API using the XML format preference. The purpose of the World Bank Linked Data is to allow consumers and publishers to link to other Linked Open datasets for more information. There is a public SPARQL endpoint available that accepts SPARQL queries. The endpoint link is:

http://worldbank.270a.info/sparql

Below is a SPARQL query to extract the GDP rate of France in year 2012.

```
SELECT ?countryLabel ?GDPA
WHERE {
        GRAPH g-indicators: {
                ?obvGDPA        property:indicator              indicator:NY.GDP.MKTP.KD.ZG ;
                                sdmx-dimension:refArea          ?countryURI;
                                sdmx-dimension:refPeriod        year:2012;
                                sdmx-measure:obsValue           ?GDPA. }
        GRAPH g-meta:{
                ?countryURI     a                       dbo:Country ;
                                skos:prefLabel          ?countryLabel;
                FILTER (str(?countryLabel) = "France")}
        }
```

The ground facts that are retrieved from World Bank online dataset are about the economic indicators of companies' countries including information about GDP, unemployment and inflation rates of countries' economy besides other information such as population number.

Below is an example of how the triples of a GDP rate of a specific country are added to the semantic knowledgebase.

kbfwo:united_states → kbfwo:hasGDPRateNAry → kbfwo:hasgdpraterelation_2
kbfwo:hasgdpraterelation_2 → kbfwo:hasGDPRate → "2.59614804050973"^^xsd:float
kbfwo:hasgdpraterelation_2 → kbfwo:hasIndicatorName→
"GDP growth (annual %)"^^xsd:strin
kbfwo:hasgdpraterelation_2 → kbfwo:hasIndicatorSymbol →
"NY.GDP.MKTP.KD.ZG"^^xsd:string
kbfwo:hasgdpraterelation_2 → kbfwo:hasDate → kbfwo:201511_2
kbfwo:201511_2 → kbfwo:hasExtractedDateValue → "2015"^^xsd:string
kbfwo:hasgdpraterelation_2 → kbfwo:hasDataSource → kbfwo:world_bank_open_data

Then, the retrieved information from the World Bank Open Data is mapped to N-ray relation pattern triples and inserted into the semantic knowledgebase. The graph example of these triples is shown in Figure 8.5 below.

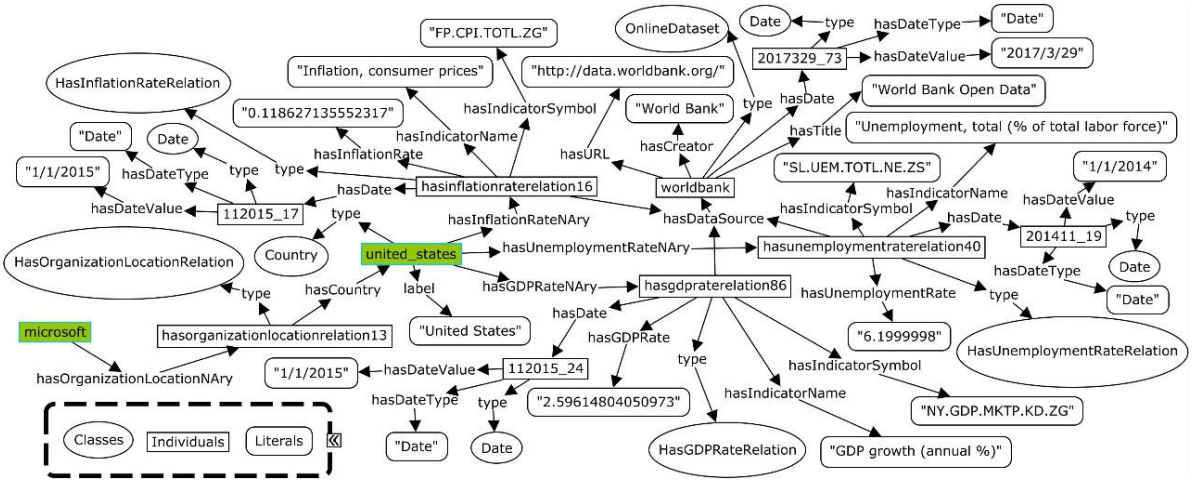

Figure 8.5: Enriching Triples Example by using World Bank Online dataset.

To conclude, we demonstrated above that the formalism in modelling the semantic knowledgebase provides a great opportunity to leverage domain-relevant facts that are published in structured and semi-structured data sets.

The output of this stage is an enriched semantic knowledgebase which is understandable by machines. For storing the resultant semantic knowledgebase, we have employed the RDF triple store of Jena framework, Triple Database (TDB). It is a native triple store and successfully used for storing and managing semantic facts, which are published in RDF triple model (Bunakov 2015).

8.3 Domain-Specific Data Requirements for the Decision-Making Process

In some domains, the extracted information requires further processing to apply the reasoning tasks for decision-making activities such as numeric calculations for some of the decision's factors. These are domain-specific requirements and 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. In our problem domain use-case scenario, the companies' performance numeric rates are required to be calculated using other existing numeric rates. These performance numeric rates will be calculated in accordance to the formulation of the decision-making task as explained in the motivating scenario in chapter 3. It adopts the constant growth DDM, which is also known as Gordon growth model, to calculate the intrinsic value of the stock (V) and compare to the current companies' stock prices (P).

The intrinsic value is calculated by relating it to its expected dividends in the next year time period (D), the required rate of return by investor (k) and the expected growth rate in dividends (k) (see equation (3.1)). The companies dividends payments history (D_i) and the companies' current stock prices (P) are supposed to be exist in the semantic knowledgebase or they can be extracted from an appropriate data source such as Yahoo Finance API. However, the expected growth rate in dividends (g) and the required rate of return by investor (k) should be calculated by using equations (3.4) and (3.5) respectively. Then, the intrinsic value of the stock (V) is calculated by using the equation (3.3).

The calculated values and rates will be inserted into the semantic knowledgebase as triples. These triples are represented in N-ary relation pattern. The example below shows these triples for a company (kbfwo:microsoft):

- **The N-ary relation triples of the current stock price:**

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

- **The N-ary relation triples of the dividend yield rate**

kbfwo:microsoft → kbfwo:hasDividendYieldPercentageNAry →
kbfwo:hasdividendyieldpercentagerelation_1
kbfwo:hasdividendyieldpercentagerelation_1 → kbfwo:hasDividendYieldPercentageValue →
"2.40"^^xsd:float
kbfwo:hasdividendyieldpercentagerelation_1 → kbfwo:hasDividendYieldPercentageDate →
kbfwo:2932017_1
kbfwo:2932017_1 → kbfwo:hasDateValue → "2017-3-29"^^xsd:date

- **The N-ary relation triples of the dividend growth rate**

kbfwo:Microsoft → kbfwo:hasDividendGrowthRateNAry → kbfwo:dividendgrowthrate_1
kbfwo:dividendgrowthrate_1 → kbfwo:hasDividendGrowthRateValue → "0.07709492"^^xsd:float
kbfwo:dividendgrowthrate_1 → kbfwo:hasDividendGrowthRateDate →
kbfwo:dividendgrowthratedate_4
kbfwo:dividendgrowthratedate_1 → kbfwo:hasDateValue → "2017-3-29"^^xsd:date

- **The N-ary relation triples of the expected return rate**

kbfwo:microsoft → kbfwo:hasExpectedReturn RateNAry → kbfwo:expectedreturnrate_1
kbfwo:expectedreturnrate _1 → kbfwo:hasExpectedReturnRateValue → "0.10109492"^^xsd:float
kbfwo:expectedreturnrate _1 → kbfwo:hasExpectedReturnRateDate →
kbfwo:expectedreturnratedate_4
kbfwo:dividendgrowthratedate_1 → kbfwo:hasDateValue → "2017-3-29"^^xsd:date

- **The N-ary relation triples of the price valuation value**

kbfwo:microsoft → kbfwo:hasStockPriceValuationNAry → kbfwo:stockpricevaluation_1
kbfwo:stockpricevaluation_1→ kbfwo:hasStockPriceValuationValue → "70.01118"^^xsd:float
kbfwo:stockpricevaluation_1→ kbfwo:hasStockPriceValuationDate →
kbfwo:stockpricevaluationdate_4

kbfwo:stockpricevaluationdate_4 → kbfwo:hasDateValue → "2017-3-29"^^xsd:date

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 (P) is less than the intrinsic value of the stock (V), the decision should be to buy or hold the stock; otherwise, sell the stock.

In the next subsection, we will explain the details of exploiting the semantic knowledgebase.

8.4 Exploiting the Semantic Knowledgebase: Decision Support and Information Exploration

This research work uses the financial information exploration and financial decision-making activities as motivating scenario for the proposed framework. The resulting semantic knowledgebase is intelligently exploited to support the stock investment decision-making process by adopting a Semantic Web based method to deliver inferred new and interesting facts to end users. Our motivating scenario is about assisting individual investors who would like to decide whether to invest in individual stocks or sell and reinvest in other individual stocks. The exploitation of the semantic Knowledgebase is approached through designing and implementing a roadmap for developing a knowledge-based application by employing the Semantic Web Technologies.

Our motivating scenario presented two use-cases. The first use-case scenario is that the investor requests a support in making a stock investment decision in a specific company. The second use-case scenario is that the investor explores the semantic knowledgebase to make the decision by him/herself (see chapter 3 above). After constructing the semantic knowledgebase and applying domain-specific data pre-processing, the system receives and processes the user's request and delivers the request' answer. The interrogation of the knowledgebase will be according to the user request and the response could be producing the recommended decision or exploring the semantic Knowledgebase. Figure 8.6 below illustrates the workflow for exploring semantic knowledgebase and the integrated Decision Support System. The workflow starts with applying an 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.

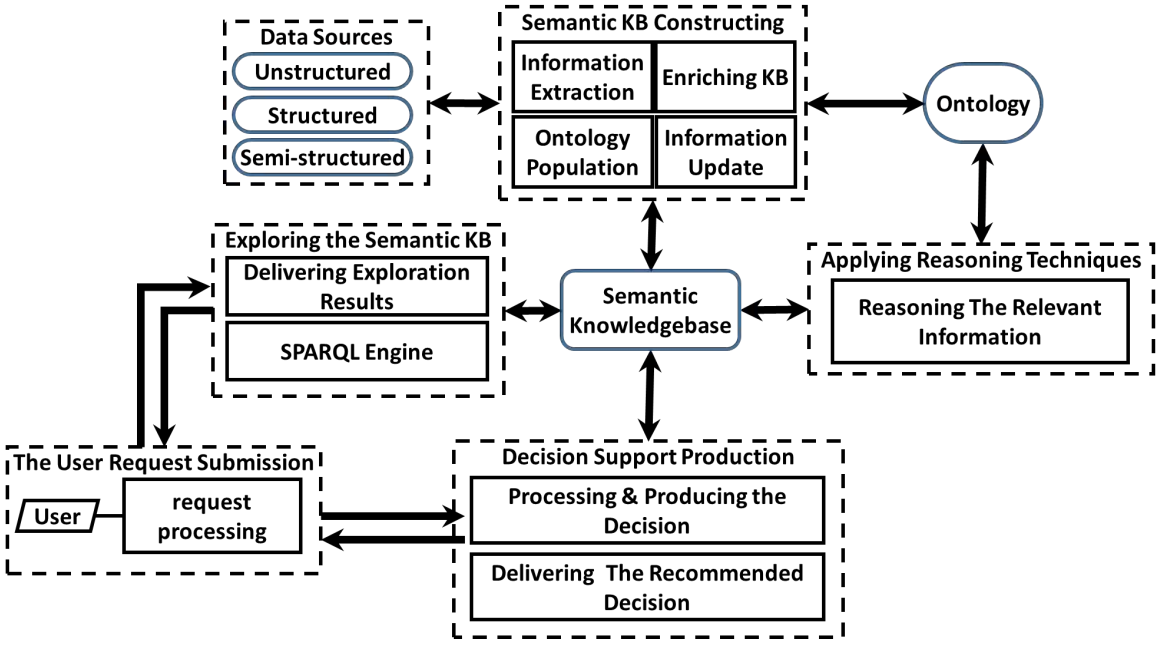

Figure 8.6: The workflow exploring semantic knowledgebase and the integrated Decision Support System

8.4.1 Applying Ontology Reasoning Techniques on the Semantic Knowledgebase

As aforementioned in chapter 4, there are two categories of reasoning in Semantic Web context, Ontology OWL reasoning and user-defined rule-based reasoning. In this stage, we have applied OWL reasoning that can achieve many tasks such as automatic class subsumption and automatic Individuals classification. These tasks correspond to standard Description Logics reasoning tasks. Description Logics allows specifying a terminological hierarchy using a restricted set of First-Order formulas. The equivalence of OWL and description logics allows OWL to exploit the considerable existing body of DL reasoning fulfil important logical requirements. These requirements include concept satisfiability, class subsumption, class consistency, and instance checking (Wang, et al. 2004).

Below, we drive some examples of applying the ontology reasoning on our resultant semantic knowledgebase to achieve different tasks.

Example 1: Assuming the relevant classes representing (kbfwo:User), (kbfwo:Company), (kbfwo:Stock), (kbfwo:StockHolder) and (kbfwo:HasStockRelation). Let us consider the following triples are exist in the semantic knowledgebase,

kbfwo:kwakeb → rdf:type → kbfwo:Company
kbfwo:hadi → rdf:type → kbfwo:User
kbfwo:shares001→ rdf:type → kbfwo:Stock
kbfwo:hasstockrelation001→ rdf:type → kbfwo:HasStockRelation
kbfwo:hadi → kbfwo:hasStockNAry → kbfwo:hasstockrelation001
kbfwo:hasstockrelation001→ kbfwo:hasStock → kbfwo:shares001

If we would like to classify persons and organisations individuals as stock holders in a specific class, (kbfwo:StockHolder), we can apply the OWL existential restrictions. 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 those are members of a specific class. It can be represented by using Manchester syntax as below:

Class: D EquivalentTo: P some C

Where:

D is an named class to be equivalent to the unnamed restriction class

C is the named class that has the individuals which they will be used to define the individuals of the unnamed restriction class.

P is the property that is used to link between its right hand side individuals of C class and the individuals of the unnamed restriction class.

However, because we are using N-ary representation, the above formals will be as in the Manchester syntax form below:

Class: D EquivalentTo: P some (P1 some C)

Where P is the main N-ary relation and P_1 is one of the properties that is used to link between the instances of intermediate 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)

It is worth noting that there are two restriction classes, external and internal, and the external restriction class is the target to be equivalent to (kbfwo:StockHolder).

The meaning of this restriction is that exactly those individuals will belong to the anonymous internal restricted class that has at least one (kbfwo:hasStock) property with an individual belonging to a given class description (kbfwo:Stock). Those individuals exactly will belong to the anonymous external restricted class that has at least one (kbfwo:hasStockNAry) property with an individual belonging to the anonymous internal restricted class.

It is worth noting, also, that we use (owl:equivalentClass) to relate the restriction to the class being described, (kbfwo:StockHolder). The equivalent classes are sometimes referred to as a Necessary & Sufficient criteria. This is because the restriction specifies necessary and sufficient conditions for being a stock holder. Anyone who is a stock holder must own at least one company share, and anyone who has at least one company share is a Stock holder. In other words, not only are the conditions necessary for membership of the class (kbfwo:StockHolder), they are also sufficient to determine that any individual that satisfies them must be a member of the class (kbfwo:StockHolder).

Returning to "Example 1", since (kbfwo:hadi) has as a (kbfwo:hasStockNAry) with (kbfwo:hasstockrelation001) and (kbfwo:hasstockrelation001) has a (kbfwo:hasStock) relation with (kbfwo:shares001). By iterating through all the individuals in an OWL ontology, querying for subsets of named individuals with certain properties can be achieved. A reasoner would derive the following statement:

kbfwo:hadi → rdf:type → kbfwo:StockHolder

Example 2: In the same way, the companies' events can be classified as positive and negative events such as the increase and decrease of profits margins in the news. Assuming the relevant classes representing (kbfwo:Company), (kbfwo:OrganizationPositiveEvents), (kbfwo:OrganizationNegativeEvents) and (kbfwo:SharePriceChange). Also, Let us consider the following triples are exist in the semantic knowledgebase,

kbfwo:microsoft → rdf:type → kbfwo:Company
kbfwo:ibm → rdf:type → kbfwo:Company
kbfwo:sharepricechange001→ rdf:type → kbfwo:SharePriceChange
kbfwo:profitmarginchange001→ rdf:type → kbfwo:ProfitMarginChange
kbfwo:microsoft → kbfwo:sharePriceChange → kbfwo:sharepricechange001
kbfwo:ibm → kbfwo:profitMarginChange → kbfwo:profitmarginchange001
kbfwo:sharepricechange001 → kbfwo:shareIncreasedBy → "3%"^^xsd:string
kbfwo:profitmarginchange001 → kbfwo:profitDecreasedBy → "1%"^^xsd:string

If we would like to classify the polarity of these events in a specific classes, (kbfwo:OrganizationPositiveEvents or kbfwo:OrganizationNegativeEvents), we can apply the OWL existential restrictions as in example above. It can be represented by using Manchester syntax as below:

Class: kbfwo:OrganizationPositiveEvents EquivalentTo: (kbfwo:hasShareIncreasedBy some xsd:string)
or (kbfwo:hasProfitIncreasedBy some xsd:string)

And,

Class: kbfwo:OrganizationNegativeEvents EquivalentTo: (kbfwo:hasShareDecreasedBy some xsd:string) or (kbfwo:hasProfitDecreasedBy some xsd:string)

It is worth noting that only restriction class required to be included in the axioms is the external restriction classes because we only need to classify the events instances (of N-ary classes) which are (kbfwo:sharepricechange001 and kbfwo:profitmarginchange001) and the restriction classes is to be equivalent to the appropriate polarity class, (kbfwo:OrganizationPositiveEvents) or (kbfwo:OrganizationNegativeEvents). Applying the reasoner would derive the following statements:

kbfwo:sharepricechange001 → rdf:type → kbfwo:OrganizationPositiveEvents
kbfwo:profitmarginchange001 → rdf:type → kbfwo:OrganizationNegativeEvents

Example 3: the OWL object properties characteristics such as functional, inverse functional, transitive, symmetric, asymmetric, reflexive and irreflexive can be used to enhance reasoning about properties. Below is an example of reasoning about the functional characteristic of the object properties (kbfwo:hasStockSymbolNAry) and (kbfwo:hasStockSymbol). Also, these classes are exist in an ontology, (kbfwo:Company) and (kbfwo:HasStockSymbolRelation). These classes have individual members as in the triples below.

kbfwo:spring_co → rdf:type → kbfwo:Company
kbfwo:msft → rdf:type → kbfwo:Stock
kbfwo:hasstocksymbolrelation001 → ref:type → kbfwo:HasStockSymbolRelation

Also, consider the following N-ary relation triples are present in the semantic knowledgebase,

kbfwo:spring_co → kbfwo:hasStockSymbolNAry → kbfwo:employerofrelation001
kbfwo:employerofrelation001 → kbfwo:hasStockSymbol → kbfwo:msft

Because both object properties (kbfwo:hasStockSymbolNAry) and (kbfwo:hasStockSymbol) have a functional characteristic, this means that for any given individual, there can be at most one out going relationship along the property for that individual. However, if multiple individuals are specified as values for that property then the reasoner will infer these values to denote the same individual. For example, if the triples below are exist in semantic knowledgebase:

kbfwo:spring_co → kbfwo:hasStockSymbolNAry → kbfwo:employerofrelation002
kbfwo:employerofrelation002 → kbfwo:hasStockSymbol → kbfwo:msft2

Then, the reasoner will infer that (kbfwo:employerofrelation001) and (kbfwo:employerofrelation002) are the same and (kbfwo:msft) and (kbfwo:msft2) are also the same. Accordingly, the reasoner will generate the following new triples:

kbfwo:employerofrelation001 → owl:sameAs → kbfwo:employerofrelation002
kbfwo:msft → owl:sameAs → kbfwo:msft2

8.4.2 The User Request Submission Component

The developed semantic Knowledgebase can be exploited either via intelligent exploration or by explicitly requesting decision support recommendation. In the implementation use-case scenario the information about the user and her/his request that will be inserted in the semantic knowledgebase includes:

- The type of the request, Recommendation or Exploration" and its date.
- The profile of the user who launched the request such as user's name, gender, date of birth and nationality.
- The targeted investment company and the kind of information to be explored.

The knowledge interrogation component receives the request from an investor with his complete information. Then, the system will check the semantic Knowledgebase whether the investor has any historic information related to previous requests or whether the investor is local or foreign regarding to the targeted investment company location and the user's nationality to be used in decision-making process.

Our ontology contains a class called (kbfwo:Request). Each single request submitted to the Semantic Web application by a user triggers the instantiation of a new request instance as a member of that class. Then, a set of ontology individuals and assertions on them that fully describe a specific request will be added. For example, the owner of the request, and the target company. To exemplify this process, consider that a user (Hadi) submits a request to get a recommendation for stock investment in a company called "Microsoft".

The system will search the knowledgebase to check if the user is exist in the knowledgebase and has a history of requests. If the uses does not exist, the system will generate a new instance for the user then generate the required triples for the user's and her/his request's information in the knowledgebase. In addition, if the targeted company is not exist in the knowledgebase, the system will attempt to update its information from the available data sources. The following two triples should be in the knowledgebase before or after the user request.

kbfwo:microsoft → rdf:type → kbfwo:Company
kbfwo:hadi → rdf:type → kbfwo:User

Then, the system will generate the triples related to the request details and insert them into the knowledgebase such as the request type, date and targeted company if the request type is recommendation. Below is the triples that are related to the example above.

kbfwo:request001→ rdf:type → kbfwo:Request
kbfwo:hadi → kbfwo:hasRequest → kbfwo:request001
kbfwo:request001 → kbfwo:hasRequestType → "Recommendation"^^xsd:string
kbfwo:request001 → kbfwo:hasCompanyTarget → kbfwo:microsoft
kbfwo:request001 → kbfwo:hasRequestDate → kbfwo:requestdate001
kbfwo:requestdate001 → kbfwo:hasDateValue → "01/07/2017"^^xsd:dateTime

In addition, this component is responsible on delivering the response to user's request, exploration or recommendation. If the user requests knowledgebase exploration, the system will model the request into a semantic query and apply it to retrieve information to assist her/his investment decision. Then, the system present the answer of that query to the user.

If the user requests a recommendation from decision-making support system, the system will apply the required inference semantic rules techniques to process the decision-making. Then, the system will produce and deliver the recommended decision to the user. Also, the system presents the information that is used to make the decision to the user such as the performance rates of the targeted company and its related events in the news. The ontology of our system contains a class called (kbfwo:Decision). The system will generate the triples related to the produced recommended decision and insert them into the knowledgebase. The recommended decisions are produced by the system will be based on the available data. The instantiation of each instances of class (kbfwo:Decision) is done incrementally in a sequence of steps while the request is processed during the subsequent decision-making process phases.

8.4.3 The Recommended Decision Production Component

If the use-case scenario is that the user requests a recommended decision to support her/his stock investment in a specific company, the system starts gathering the relevant information for a given decision-making problem.

The task of the decision support production component is describing the user request to select the background knowledge to interpret the data and deduce the consequences of the investment advice for the user by reasoning over the relevant information. Based on the user query, our implementation attempts to present end users with inferred facts information from the existing information in the semantic knowledgebase to produce recommended stock investment decisions that are based on the available data related to the companies' stocks investment and user profile. If the information is not available in the semantic Knowledgebase, the system will attempt to update the semantic knowledgebase by extracting information from the available external data sources.

To explain the stock investment decision-making process, we drive the following use-case scenario:

Suppose there is a user (kbfwo:hadi) who submits a request (kbfwo:request_1) to ask for a recommendation of stock investment (buy, hold or sell shares) for a specific company (kbfwo:mircosoft). The system will recommend a decision according to some information related to the targeted company that exist in the semantic knowledgebase. The process of this stock investment decision-making is based on three categories of information, company information, country economic information, and online news information.

8.4.3.1 Stock investment decision-making process by analysing company information:

The recommended investment decision will be taken according to the fact that whether the current stock prices is under-valuated or not. This decision requires retrieving the current stock price (P) and the intrinsic value of the stock (V). In other words, if the current stock price (P) is less than the intrinsic value of the stock (V), the decision should be to buy or hold the stock; otherwise, sell the stock. This decision can be converted into rules as below. The intrinsic value of the stock is considered and inserted into the knowledgebase based on the stock investment model explained in chapter 4. However, if the required information is not available or outdated, it should be retrieved from the appropriate data sources and the required performance rates of the targeted company should be recalculated. Below is the pseudo code of the stock investment decision-making process by using company information:

1: **Begin**
2: The user submit a recommendation request with the targeted company
3: **If** the required information for processing the decision-making is available in the semantic knowledgebase and up to date, **Then Go to 5.**
4: Extract the required company information to process the decision-making from appropriate external data sources.
5: Calculate the required companies' performance numeric rates.
6: Retrieve the value of the current stock price (P)
7: Retrieve the intrinsic value of the stock (V)
8: **If** (P<V) **Then**
9: The recommended decision is "The current stock price is under valuated. The user should buy or hold the stock of this company."
10: **Else If** (P>V) **Then**
11: The recommended decision is "The current stock price over valuated. the decision should be to sell the stock of this company."
12: **End If**
12: Delivery the recommended decision to the user
13: **End**

As aforementioned, OWL comes with a set of powerful reasoning tasks with well-understood computational properties. OWL reasoning tasks include subsumption, satisfiability, consistency, instance checking and realisation. Since OWL axioms and class expressions are variable-free and modelling constructs of OWL not always adequate; for example, there are statements may not be expressed simply in OWL and OWL may not suffice for all applications. To achieve decidability, OWL trades expressiveness for reasoning efficiency. To leverage OWL's limited relational expressiveness and overcome modelling shortcomings that OWL alone would insufficiently address, integration of OWL with rules can be an alternative paradigm for reasoning process. User-defined rules on top of the ontology allow expressing richer semantic relations that lie beyond OWL's expressive capabilities and couple ontological and rule knowledge. In rule-based reasoning, reasoners apply rules with data to reason and derive new facts. When the data match the rules

conditions, the reasoners can modify the knowledge base; for example, fact assertion or retraction, or to execute functions (Hitzler, Krotzsch and Rudolph 2009, Ye, et al. 2015).

In this research, we have used Jena reasoning engine in the Jena framework. It supports three rule types of reasoning process, forward chaining, backward-chaining and A hybrid execution. However, we have transferred the pseudo code above to Jena forward chaining rules. Forward chaining is a bottom-up computational model. It starts with a set of known facts and applies rules to generate new facts whose premises match the known facts. The inference moves forward from the facts toward the goal. The Jena rules for the pseudo code above are shown below.

- **The first condition rule, (P<V)**

```
[rule1: (kbfwo:hadi kbfwo:hasRequest kbfwo:request_1)
( kbfwo:request_1   kbfwo:hasCompanyTarget ?com)
(?com  kbfwo:hasSharePriceNAry ?suri)
(?suri  kbfwo:hasSharePrice ?P)
(?com  kbfwo:hasStockPriceValuationNAry ?vuri)
(?vuri  kbfwo:hasStockPriceValuationValue ?V)
lessThan(?P, ?V)
->
( kbfwo:decision_1   kbfwo:hasDecisionConclusion
                    'buy or hold the stock because it is under valuated.'^^xsd:string)
( kbfwo:decision_1   rdf:type  kbfwo:Decision )
( kbfwo:decision_1   kbfwo:hasDecisionDate  kbfwo:currentdate_1 )
( kbfwo:currentdate_1   kbfwo:hasDateValue 'Current Date'^^xsd.date )
( kbfwo:request_1   kbfwo:hasDecision  kbfwo:decision_1 )]
```

- **The second condition rule, (P>V)**

```
[rule2: (kbfwo:hadi kbfwo:hasRequest kbfwo:request_1)
( kbfwo:request_1   kbfwo:hasCompanyTarget ?com)
(?com  kbfwo:hasSharePriceNAry ?suri)
(?suri  kbfwo:hasSharePrice ?P)
(?com  kbfwo:hasStockPriceValuationNAry ?vuri)
(?vuri  kbfwo:hasStockPriceValuationValue ?V)
greaterThan(?P, ?V)
->
( kbfwo:decision_1   kbfwo:hasDecisionConclusion
                        'sell or do not buy the stock because it is under valuated.'^^xsd:string)
( kbfwo:decision_1   rdf:type kbfwo:Decision )
( kbfwo:decision_1   kbfwo:hasDecisionDate  kbfwo:currentdate_1 )
( kbfwo:currentdate_1   kbfwo:hasDateValue 'Current Date'^^xsd.date )
( kbfwo:request_1   kbfwo:hasDecision  kbfwo:decision_1 )]
```

The Jena rules are bound to rules reasoner engine, then, the rules are executed. The reasoner will generate the following triple.

kbfwo:decision_1→ rdf:type → kbfwo:Decision
kbfwo:decision_1 → kbfwo:hasDecisionDate → kbfwo:currentdate_1
kbfwo:currentdate_1 → kbfwo:hasDateValue → "22/07/2017"^^xsd.date
kbfwo:request_1 → kbfwo:hasDecision → kbfwo:kbfwo:decision_1

The tripe of the recommended decision could be:

kbfwo:decision_1 → kbfwo:hasDecisionConclusion →
'buy or hold the stock because it is under valuated.'^^xsd:string

Or:

kbfwo:decision_1 → kbfwo:hasDecisionConclusion →
'sell or do not buy the stock because it is under valuated.'^^xsd:string

Then, the recommended stock investment will be delivered to user.

8.4.3.2 Stock investment decision making process by analysing country economic information:

As explained in section 3.3.1.1, the economic stability of the country of the targeted company is a crucial factor that impacts the stock investment decision-making process. Because the economic situation in a specific country influences the profitability of company's stocks in the that country. The confidence in stock markets increases when the prices of the stocks continue to grow. Usually, Individual investors are affected by the positive or negative public announcements of the economic indicators because the process of making decisions and executing trades could take different amount of time for buy or sell stocks. In general, their behaviour is central to the stability of country's economy.

In this research, we have employed some of critical economic indicators to measure the country economies' stability, which are Gross Domestic Product (GDP), inflation and unemployment rates. However, the stability of macroeconomics must be measured collectively by all of these indicators because they are interdependent. Although there is no precise guidelines for a known critical or threshold values for them, we adopted the threshold values of GDP, unemployment and inflation rates advocated by Cashell in (Cashell 2006) and Pollin, et al. in (Pollin and Zhu 2006). These threshold values are presented in Table 3.1. In addition, we adopted the relation between economy indicators and the stability of the countries' economy that presents in Table 3.2. These stability relations are examples of general relationships between the economy indicators that are provided by Kolovson in (Kolovson 2014). We adopted these economic indicators, their threshold values and their stability relations solely to explain the role of rule-based reasoning in stock investment decision-making process by using the country's economy stability information. Below is the pseudo code of the stock investment decision-making process by using country information:

1: **Begin**
2: Extract the country of the targeted company
3: **If** the required information about the targeted country for processing the decision-making is available in the semantic knowledgebase and up to date, **Then Go to 5**.
4: Extract the required country information to process the decision-making from appropriate external data sources.
5: Retrieve the value of the GDP rate (GDP)
6: Retrieve the value of the inflation rate (INFL)
7: Retrieve the value of the unemployment rate (UNEM)
8: Retrieve the minimum (GMIN) and maximum (GMAX) thresholds of GDP rate
9: Retrieve the minimum (IMIN) and maximum (IMAX) thresholds of inflation rate
10: Retrieve the minimum (UMIN) and maximum (UMAX) thresholds of unemployment rate
11: **If** (minGDP <= GDP <= maxGDP) and
(minUNEM <= UNEM <= maxUNEM) and
(minINFL <= INFL <= maxINFL) **Then**
12: The recommended decision is "The economy situation of this country is acceptable for stock investment."
13: **Else If** (minGDP <= GDP <= maxGDP) and
(minUNEM <= UNEM <= maxUNEM) and
(INFL >= maxINFL) **Then**
The recommended decision is "The economy situation of this country is acceptable for stock investment." Else,
14: **Else If** (minGDP >= GDP) and
(UNEM >= maxUNEM) and
(INFL >= maxINFL) **Then**
15: The recommended decision is "There is a risk in stock investment in this country."
16: **Else If** (minGDP >= GDP) and
(UNEM >= maxUNEM) and
(minINFL >= INFL) **Then**
17: The recommended decision is "There is a risk in stock investment in this country."
18: **Else If** (minGDP < GDP < maxGDP) and
(minUNEM >= UNEM) and
(INFL >= maxINFL) **Then**
19: The recommended decision is "There is a risk in stock investment in this country."
20: **End If**
21: Delivery the recommended decision to the user
22: **End**

As in stock investment decision-making process by using company information, we have used forward chaining format of Jena reasoner engine to transfer the pseudo code above to rules for two examples of the stability situation in the Table 3.2, which are:

1- Jena rules when the three indicators are stable:

```
[rule3: (kbfwo:request_1  kbfwo:hasCompanyTarget ?com)
(?com kbfwo:hasOrganizationLocationNAry ?luri)
(?luri kbfwo:hasCountry ?cntr)
(?cntr kbfwo:hasGDPRateNAry ?guri)
(?guri kbfwo:hasGDPRate ?gdp)
(?cntr kbfwo:hasUnemploymentRateNAry ?uuri)
(?uuri kbfwo:hasUnemploymentRate ?unem)
(?cntr kbfwo:hasInflationRateNAry ?iuri)
(?iuri kbfwo:hasInflationRate ?infl)
(kbfwo:request_1 kbfwo:hasDecision ?deci)
(?g1 kbfwo:hasGDPMaximumThreshold ?maxGDP)
(?g2 kbfwo:hasGDPMinimumThreshold ?minGDP)
```

```
(?f1 kbfwo:hasInflationMaximumThreshold ?maxInfl)
(?f2 kbfwo:hasInflationMinimumThreshold ?minInfl)
(?u1 kbfwo:hasUnemploymentMaximumThreshold ?maxUnem)
(?u2 kbfwo:hasUnemploymentMinimumThreshold ?minUnem)
ge(?gdp, ?minGDP) le(?gdp, ?maxGDP)
ge(?unem, ?minUnem) le(?unem, ?maxUnem)
ge(?infl, ?minInfl) le(?infl, ?maxInfl)
 ->
(?deci kbfwo:hasEconomyConclusion
                'The economy situation of this country is acceptable for stock investment.')];
```

2- The Jena rules when GDP and inflation rates are low and unemployment rate is high.

```
[rule3: (kbfwo:request_1  kbfwo:hasCompanyTarget ?com)
(?com kbfwo:hasOrganizationLocationNAry ?luri)
(?luri kbfwo:hasCountry ?cntr)
(?cntr kbfwo:hasGDPRateNAry ?guri)
(?guri kbfwo:hasGDPRate ?gdp)
(?cntr kbfwo:hasUnemploymentRateNAry ?uuri)
(?uuri kbfwo:hasUnemploymentRate ?unem)
(?cntr kbfwo:hasInflationRateNAry ?iuri)
(?iuri kbfwo:hasInflationRate ?infl)
(kbfwo:request_1 kbfwo:hasDecision ?deci)
(?g1 kbfwo:hasGDPMaximumThreshold ?maxGDP)
(?g2 kbfwo:hasGDPMinimumThreshold ?minGDP)
(?f1 kbfwo:hasInflationMaximumThreshold ?maxInfl)
(?f2 kbfwo:hasInflationMinimumThreshold ?minInfl)
(?u1 kbfwo:hasUnemploymentMaximumThreshold ?maxUnem)
(?u2 kbfwo:hasUnemploymentMinimumThreshold ?minUnem)
le(?gdp, ?minGDP)
ge(?unem, ?maxUnem)
ge(?infl, ?maxInfl)
 ->
(?deci kbfwo:hasEconomyConclusion
                  'There is a risk in stock investment in this country.')];
```

After these rules are bounded to rule reasoner engine to be executed, the reasoner will generate the following triple.

The type of the recommended decision which is rated to the countries' economy could be:

```
kbfwo:decision_1 → kbfwo:hasEconomyConclusion →
                'The economy situation of this country is acceptable for stock investment.'^^xsd:string
```

Or

```
kbfwo:decision_1 → kbfwo:hasEconomyConclusion →
                        'There is a risk in stock investment in this country.'^^xsd:string
```

Then, the recommended stock investment will be delivered to user.

8.4.3.3 Stock investment decision making process by using online news information:

One of the important factors that affects the human behaviour; subsequently, stock investment decision-making is the relevant events published in online news. It is because the activities of stock markets are performed by human such as selling and buying stocks. As news articles will influence our decision and our decision will influence the stock prices. However, the texts in these online news articles are written in human languages and required to be extracted and processed automatically to support the stock investment decision-making process.

In our approach, the main task of the proposed framework is extracting information from unstructured online news and presenting it to investors as structured information to be easily explored and understood by machines. In the pre-processing stage, the system has classified the news as positive and negative events. As explained in Example 2 of section 8.4.2, the positive news instances will be members of (kbfwo:OrganizationPositiveEvents) class and the negative news instances will be members of (kbfwo:OrganizationNegativeEvents) class.

The recommended decision production component will present the negative and positive news to the investor with the recommend decision that is delivered to investor. Then, the investor can use this information as a negative or positive indicators to decide whether to proceed in stock investment process.

8.4.4 Exploring the Semantic Knowledgebase Component

Intelligently exploring the semantic knowledgebase and retrieving appropriate information are very important techniques for semantic knowledgebase applications. In this section, we focus on intelligent exploration of the semantic knowledgebase by using Semantic Web Technologies. Semantic Web Technologies allow for systemic and standardised modelling and compilation of knowledge for the targeted problem domain which provides for the deep understanding of published (processed) semantic data. In addition, Semantic Web Technologies are capable of supporting advanced exploration scenarios and solve complex information needs such as supporting the decision-making process.

In exploration scenarios, where the investor intends to explore the semantic knowledgebase to independently make decision, the knowledge interrogation component initially checks whether the requested information is available in the semantic knowledgebase. If it is available, the system presents the relevant information to the user. If it is not, the system attempts to extract that information from the relevant unstructured, semi-structured or structured data sources.

In order to make sense of the data in the semantic knowledgebase and enable views and queries over semantic knowledgebase, we employed an ontology for modelling that

knowledge to be used as vocabularies for the domain knowledge. In this context, our goal is to support the users in discovering and understanding our problem domain knowledge and answering specific questions about a specific task; for example, support stock market investment decision-making process through semantic knowledgebase exploration.

On the one hand, our semantic knowledgebase is designed by using Semantic Web Technologies to be understandable by machines. On the other hand, humans are the real consumers of that semantic knowledgebase thus end users should have usable tools and simple methods to explore the Web of Data.

The query languages' specifications, which will be employed to explore semantic knowledgebase, should be capable to explore that knowledge representation standard. Because the resultant semantic knowledgebase is represented and stored in RDF triples standard, we utilised SPARQL query language 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. The outcomes of SPARQL queries can be results sets or RDF graphs (Prud and Seaborne 2006).

SPARQL allows users to specify a graph pattern containing variables to query RDF data. Then, this pattern is matched against the targeted RDF data source. All matched RDF triples will be returned to user. 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 our 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. The tabular form of the result of the above query example is below.

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

The use-case scenario of exploring the semantic knowledgebase is topic-oriented. For each topic such as companies' information and countries' information, a knowledge elements of facts that provide information on this topic is specified. Each fact is assigned to SPARQL queries for its selection under specific conditions derived mainly from the user request. For instance, the companies' information can be included into the following facts frames:

- The current stock price of a specific company.
- The minimum price of a specific company in the last year.
- A list of performance rates or a specific performance rate for a specific company
- The news events related to a specific company
- The location of a specific company
- A list of a management team of a specific company
- A list of companies in a specific industry sector that are located in a specific country

And the countries' information can be included into the following facts frames:

- The economic indicators of a country of a specific company
- The population of a specific country in a specific
- The capital city of a specific country

However, SPARQL engines can be utilised in information exploration to support investors in making stock investment decisions. For example, this question,

"What are the stock prices of the companies in United States and belong to Software industry sector? What are the stock symbols of those companies? What is the data source of those prices? The results should be in descending ordered by price."

Can be modelled into this SPARQL query,

```
SELECT DISTINCT (str(?ON) as ?CompanyName)
                (str(?ISN) as ?industrySectorName)
                (str(?SSN) as ?StockSymbolName)
                (str(?SP) as ?StockPrice)
                (str(?SC) as ?Currency)
                (str(?PD) as ?PriceDate)
                (str(?CN) as ?CountryName)
                (str(?DS) as ?PriceDataSource)
 WHERE {
?company a kbfwo:Company.
?company rdfs:label ?ON .
?company kbfwo:hasStockSymbolNAry ?ssNAry.
 ?ssNAry kbfwo:hasStockSymbol ?StockSymbol.
 ?StockSymbol rdfs:label ?SSN .
?company kbfwo:hasIndustrySectorNAry ?isNAry.
?isNAry kbfwo:hasIndustrySector ?industrySector.
?industrySector rdfs:label ?ISN.
?company kbfwo:hasOrganizationLocationNAry ?olNAry.
?olNAry kbfwo:hasCountry ?country.
?country rdfs:label ?CN.
?company kbfwo:hasSharePriceNAry ?spNAry.
```

```
  ?spNAry kbfwo:hasSharePrice ?SP .
  ?spNAry kbfwo:hasSharePriceCurrency ?SC .
  ?spNAry kbfwo:hasSharePriceDate ?DateResource.
  ?DateResource kbfwo:hasDateValue ?PD.
  ?spNAry kbfwo:hasDataSource ?dsNAry.
   ?dsNAry kbfwo:hasTitle ?DS.
  FILTER (regex(?ISN, "software" , "i"))
  FILTER (regex(?CN, "United States" , "i"))
} ORDER BY DESC(?SP) LIMIT 2
```

After executing the query above by using Jena SPQRQL engine, it produces the following result.

CompanyName	industrySectorName	StockSymbolName	StockPrice	Currency	PriceDate	CountryName	PriceDataSource
"Microsoft"	"Software"	"MSFT"	"73.8"	"USD"	"2017-7-25"	"United States"	"Yahoo Finance API service to return stock data"
"IBM"	"Software"	"IBM"	"146.56"	"USD"	"2017-7-25"	"United States"	"Yahoo Finance API service to return stock data"

This result can be processed by the system to be delivered to the user in appropriate format.

Another query example to answer the following question,

"Which countries do have stable economy and what are the value of their economy indicators, GDP, Inflation and Unemployment rates? What are the sources of these data?"

Also, this question can modelled by the system in SPARQL query below,

```
SELECT DISTINCT
        (str(?CN) as ?Country)
        (str(?GR) as ?GDPRate)
        (str(?GY) as ?GDPYear)
        (str(?DSTG) as ?GDPDataSource)
        (str(?UR) as ?UnemploymentRate)
        (str(?UY) as ?UnemploymentYear)
        (str(?DSTU) as ?UnemploymentDataSource)
        (str(?IR) as ?InflationRate)
        (str(?IY) as ?InflationYear)
        (str(?DSTI) as ?InflationDataSource)
where {
?country a kbfwo:Country.
?country rdfs:label ?CN.
?country kbfwo:hasGDPRateNAry ?gdpNAry.
?gdpNAry kbfwo:hasGDPRate ?GR.
?gdpNAry kbfwo:hasDate ?gdpdateResource.
?gdpdateResource kbfwo:hasExtractedDateValue ?GY.
?gdpNAry kbfwo:hasDataSource ?datasourcegdp.
?datasourcegdp kbfwo:hasTitle ?DSTG.
?country kbfwo:hasUnemploymentRateNAry ?uneNAry.
?uneNAry kbfwo:hasUnemploymentRate ?UR.
?uneNAry kbfwo:hasDate ?unedateResource.
?unedateResource kbfwo:hasExtractedDateValue ?UY.
?uneNAry kbfwo:hasDataSource ?datasourceune.
?datasourceune kbfwo:hasTitle ?DSTU.
?country kbfwo:hasInflationRateNAry ?infNAry.
?infNAry kbfwo:hasInflationRate ?IR.
?infNAry kbfwo:hasDate ?infdateResource.
?infdateResource kbfwo:hasExtractedDateValue ?IY.
```

```
?infNAry kbfwo:hasDataSource ?datasourceinf.
?datasourceinf kbfwo:hasTitle ?DSTI.
?tempGDPx kbfwo:hasGDPMaximumThreshold ?maxGDP.
?tempGDPn kbfwo:hasGDPMinimumThreshold ?minGDP.
?tempUnemx kbfwo:hasUnemploymentMaximumThreshold ?maxUnem.
?tempUnemn kbfwo:hasUnemploymentMinimumThreshold ?minUnem.
?tempInflx kbfwo:hasInflationMaximumThreshold ?maxInfl.
?tempInfln kbfwo:hasInflationMinimumThreshold ?minInfl.
FILTER ((?minGDP <= ?GR) && (?GR <= ?maxGDP))
FILTER ((?minUnem <= ?UR) && (?UR <= ?maxUnem))
FILTER ((?minInfl <= ?IR) && (?IR <= ?maxInfl))
}LIMIT 2
```

After executing the query above by the SPARQL engine, the result will as below. Also, this result can be processed by the system to be delivered to the user in appropriate format:

```
-------------------------------------------------------------------------------------------------------------------------------------------
|Country          |GDPRate | GDPYear | GDPDataSource | UneRate | UneYear | UneDataSource | InfRate | InfYear | InfDataSource|
=============================================================================================
|"United States"|"1.62"    |"2016"    | "World Bank"   | "4.87"   | "2016"  | "World Bank"     | "1.26"   | "2016"  | "World Bank" |
|"France"         |"1.19"    |"2016"    | "World Bank"   | "10.03"  | "2016"  | "World Bank"     | "0.18"   | "2016"  | "World Bank" |
-------------------------------------------------------------------------------------------------------------------------------------------
```

Users can access and explore our 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. According to W3C, a SPARQL endpoint is a conformant SPARQL protocol service that enables users, human or machines, to query a semantic knowledgebase via the SPARQL language. The format of the query results returned from these endpoints can be processed by machines. Therefore, a SPARQL endpoint is mostly conceived as a machine-friendly interface towards the semantic knowledgebase. In this research, we employed SPARQL Engine in Jena framework for querying the semantic knowledgebase our Semantic Web application and the SPARQL endpoint by using Fuseki sever in Jena package. Jena provides an extension point interfaces that allows different storage implementations to be used with the common Jena APIs for RDF, ontologies and SPARQL query.

SPARQL is an expressive query language on RDF graphs that can be used in a straightforward manner to filter facts, construct new derived facts, and specify complex patterns concerning the properties of multiple facts. W3C has recommendations for updating SPARQL. SPARQL 1.1 Update recommendation (W3C 2018) ; for example, adds a critically important new feature which is the capability to insert facts into and delete facts from semantic knowledgebase. Also, W3C defined various SPARQL entailment regimes (W3C 2018) recommendation to allow users to specify implicit knowledge about the vocabulary in an RDF graph and a mechanism to express navigation patterns through regular expressions. The semantics of SPARQL under entailment regimes is specified for the conjunctive fragment, where queries are represented as sets of RDF triples with variables and query answers are directly provided by the entailment relation of the regime (Arenas, Gottlob and Pieris 2014, Kostylev and Grau 2014, Rinne 2012).

SPARQL as a query language shares many features with other query languages such as the Structured Query Language (SQL) for querying the relational databases. SPARQL and SQL engines provide standard interfaces to the data and defines a formalism by which data are viewed. However, they differ from each in important aspects. In general, SPARQL is a relatively simple language when compared to SQL. In addition, SQL query describes a new data table that is formed by combining two or more source tables. On the other hand, SPARQL queries can describe a new graph that is formed by describing a subset of a source RDF graph. That graph, in turn, may be the result of having merged together several other graphs. The inherently recursive nature of graphs simplifies several detailed issues that arise in SQL queries. For instance, SPARQL queries do not rely on a subquery construct in many cases because the same effect can be achieved with a single query (Allemang and Hendler 2011).

8.5 Summary

In this chapter, we presented in detail the construction process of the semantic knowledgebase. Then, we looked at Semantic Web Technologies application on accessing the resultant knowledgebase on the subject of two use-case scenarios, supporting the stock investment decision-making process and exploring the semantic knowledgebase. The process of constructing the knowledgebase went through three stages, Information Extraction, ontology population and knowledgebase enriching. We believe that the formalism in modelling the semantic knowledgebase provides a great opportunity to leverage domain-relevant facts that are published in structured and semi structured datasets.

The resultant semantic Knowledgebase can be used for a variety of exploration scenarios to provide assistance in a specific subject matter. We adopted the Semantic Web Technologies to model, reason and interrogate our problem domain knowledge by developing a knowledge-based application. This application integrates the semantic knowledgebase exploration and Decision Support System. The interrogation of the Knowledgebase will be according to two use-case scenarios. The first use-case scenario is that the investors request a 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, delivering the request' answer. The request answer could be producing the recommended decision or exploring the semantic Knowledgebase.

As a part of the framework, we have applied OWL reasoning to achieve many tasks such as automatic class subsumption and automatic Individuals classification. Moreover, we utilised the rule-based reasoning to develop decision-rules based on the ontology and execute them to derive a stock investment specific recommendations. In addition, we employed SPARQL to explore the semantic knowledgebase because it is capable to

support advanced exploration scenarios and solve complex information needs such as supporting the decision-making process by allowing users to express queries across diverse RDF data sources. Because we have adopted N-ary relation pattern to represent our domain specific non-binary relations in the resultant semantic knowledgebase, we have considered the requirements of the intermediate resources in the N-ary relations patterns in representing, reasoning and querying our problem domain knowledge.

We believe that Semantic Web Technologies can support the decision-making process and it is capable of represent the user's request, the data relevant for the user request and answering the user request by producing recommended decisions. The advantages of utilising Semantic Web Technologies to represent the required domain knowledge for decision-making activities are enabling the integration of heterogeneous data sources to be processed by the Decision Support System and enabling the utilisation of the logical reasoning for some of the inference steps of the decision-making process.

9 Framework Application Requirements

9.1 Introduction

The previous chapters discussed the components and the utilisation of our proposed framework detailing its processes. This framework refers to a generic architectural paradigm of Information Extraction, integration and exploitation. During the implementation of this framework, we investigated most of the problems 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, we addressed a number of challenges in employing Semantic Web Technologies in modelling and intelligent exploration of semantic knowledge bases 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.

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, et al. 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.

In this chapter, we will summarise our experience on addressing the above mentioned challenges by using the motivating scenario discussed in chapter 3 as a use-case. These challenges will be considered by domain experts and knowledge engineers as a roadmap for employing the Semantic Web Technologies for the knowledge user to intelligently exploit knowledge in similar problem domain. Next section will present the questions which are raised by the challenged implementation tasks and the motivating scenario answers of those questions.

9.2 Use-case Scenario's Questions and the Framework's Answers

The challenged implementation questions that are raised by the motivating scenario and the answers of those question by the proposed framework are as below:

Scenario Question 1) What are the attributes that characterise the domain applications that can adopt the proposed framework?

- The availability of domain-specific unstructured, semi-structured and structured data sources.
- The targeted beneficiary groups should be interested in integrating semantic knowledge bases exploitation and decision-making support activities in specific domains.

Scenario Question 2) What is the role of the knowledge-based approach in Information Extraction and knowledge representations?

- The knowledge-based approach is based on analysed domain knowledge to understand its characteristics, which are the linguistic and structural features.
- The characteristics are employed in Information Extraction techniques, Rule-based and Machine Learned based to extract information from unstructured online data.
- The characteristics are employed in capturing the domain-relevant, concepts, arguments, facts and events to create an inference model to infer new facts from the extracted information and classify events that might be of important for end users.
- The extracted information from the unstructured online data is populated into a semantic knowledgebase by using the semantic model or ontology.
- Then, semantic knowledgebase is enriched by a retrieved information from semi-structured and structured online data.

Scenario Question 3) What are the pre-processed tasks that will be applied on the resultant semantic knowledgebase and why?

- The domain-Specific data requirements for pre-reasoning consideration. The required date include companies' performance numeric rates calculations from the existing data.
- The reasoning tasks include compiling both classification rules that are hard-wired into the ontology such as first predicate logic's Necessary & Sufficient conditions and axioms to classify events and infer information from the existing information in the semantic knowledgebase.
- The resultant inferred and the classified news events might be of importance to support end users decision-making process.

Scenario Question 4) What kind of request the knowledge-based application can receive and process?

- The application receives a request from an investor with his complete information such as the personal name, the nationality, date of birth and the name of the company which he/she would like to invest in.
- The application will check the semantic Knowledgebase whether the investor has any historic information related to previous requests.
- The application can process two user requests scenario, requesting a decision-making support and exploring the semantic knowledgebase.

Scenario Question 5) What kind of tasks will the system apply on semantic knowledgebase to answer the user's request of stock investment decision-making support?

- The system describes the user request to select the background knowledge to gather, store, and integrate the information relevant for a requested stock investment decision-making problem in the targeted company.
- The system applies Rule-based reasoning over the relevant information to produce recommended stock investment decision which is based on the available data related to the companies' stocks investment and the details of the users' request.
- The system will deliver that recommended decision to the user and present the information that is used to make the decision to the user. The system will deliver three types of recommended information advices, information related to the companies' performance, information related to countries' economy and information related to companies' events in the online news.
- If the required information for the decision-making process is not available in the semantic Knowledgebase, the system will attempt to update the semantic knowledgebase by extracting information from the available data sources, unstructured, semi-structured or structured data sources.

Scenario Question 6) What kind of tasks will the system apply on semantic knowledgebase to answer the user's exploration request?

- The application describes the user request to select the background knowledge to gather, store, and integrate the information relevant for the user's exploration requested.
- Expressing the users request into the appropriate query language, SPARQL, because the semantic knowledgebase is represented and stored in RDF triples standard.
- The query engine in the application accepts the queries and then applies them on the semantic knowledgebase to produce a result set. The produced results should reflect the contents of the knowledgebase.

- The application processes the results to be presented to the user in appropriate style.
- The application checks whether the requested information is available in the semantic knowledgebase. If it is available, the system presents the relevant information to the user. If it is not, the system attempts to extract that information from the relevant unstructured, semi-structured or structured data sources.

Scenario Question 7) What are the roles of the knowledge stakeholders, problem domain knowledge expert, knowledge engineer and knowledge user?

- 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 engineer translates the knowledge map into a machine comprehensible and usable format and stores it in a semantic knowledgebase.
- These activities should be accomplished according to the requirements of the knowledge user.

9.3 The Implementation Phases of the Knowledge-based Framework

From the preceding chapter 3, the proposed knowledge-based framework has four phases, which are:

- Phase one (Analysing and Modelling the Domain Knowledge).
- Phase two (Natural Language Pre-processing, Named Entity Recognition and Relation Classification).
- Phase three (Constructing and Enriching the Semantic Knowledgebase).
- Phase four (Applying Reasoning Techniques and Exploiting the Semantic Knowledgebase).

The tasks in those phases can be grouped into two types. The first type includes the tasks that should be configured to fit a 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. The second type includes the tasks that can be applied on any domain; for example, constructing, enriching and exploring the semantic Knowledgebase.

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 Natural Language Processing in Information Extraction to ontology engineering and intelligent inferencing in knowledge representation. 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 proposed knowledge-based framework is based on our extensive research efforts in presenting a semantic knowledgebase to be accessed by a recommender system. This framework is for developers to follow and emphasise their efforts on the problems of other and similar domains. The attributes that characterise the domain applications that can adopt the proposed framework are, the availability of domain-specific unstructured, semi-structured and structured data sources and the availability of decision-making process related to the target domain to support beneficiary groups who are interested in that domain knowledge to be explored and decision-making supported.

For example, this framework can be applied on the domain of the observed and forecasted environmental conditions such as weather, air quality and pollen. This domain is primary for the assessment of sanitary risks and reasoned daily life decisions for the entire population. In fact, end-users are increasingly aware of decisions related to this domain. The available information that is related to this domain comes from online sources. However, the predictions of the upcoming environmental conditions in these sources are vary largely and the quality of the offered data is uncertain. The data sources of this information are unstructured, semi-structured and structured. It could be also qualitative indices, presented in tables, distribution curves or colour scales. This data requires to be transfers into semantic knowledgebase for the personal context of end-users and in the context of a specific decision-making process to support end user (Wanner, et al. 2015).

We are following the recommendation that such framework should be of a rich interoperable platform, which can accommodate communication with high level of understanding amongst framework composition components and 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 (Van Heijst, Schreiber and Wielinga 1997). Table 9.1 below shows the main phases' tasks, their deliverables, resources and expertise:

Table 9.1: The main phases' tasks, their deliverables and resources

Phase	Tasks	Deliverable	Resources and Expertise
1	Analysing the domain knowledge to understand the syntactic and semantic characteristics and capture the key concepts and their interrelations for exchanging information about that domain.	Domain's the syntactic and semantic characteristics, the Key Concepts and Their Interrelations.	Domain Expert
	Domain conceptualisation	Knowledge Map	Concept map tools. Domain Expert
	Semantic model designing	Ontology	Semantic Web Technologies and Tools. Knowledge Engineer
2	Retrieving unstructured data and Detecting the main textual content	Cleansed Unstructured data	Boilerplate remover tools. Knowledge Engineer
	Applying Natural Language pre-Processing tasks	Unstructured data with linguistic features	NLP tools. Knowledge Engineer
	Utilising Online semantic datasets to inform Named Entity Recognition (Gazetteers)	Unstructured data with linguistic features and Gazetteer lists	Semantic Web Technologies and Tools Knowledge Engineer
	Recognising named entities and parsing the typed dependency Path	Unstructured data with annotated named entities with more linguistic features	NER tools Knowledge Engineer
	Detecting relation instances between the targeted entity pairs for relation classification.	Unstructured data with annotated relation instances.	JAPE rules Domain Expert and Knowledge Engineer
	Composing training datasets automatically by utilising distant supervision sources and manually from annotating the unstructured data.	Relation Classification Training datasets	JAPE rules and SPARQL queries Domain Expert and Knowledge Engineer
	Extracting features for relation classification training datasets.	Training datasets with features	JAPE rules Domain Expert and Knowledge Engineer
3	Creating relation classification models	Relation Classifiers	ML libraries
	Evaluating the relation classifiers by configuring the training datasets and applying feature selection optimising	Optimised Relation Classifiers	ML libraries and GA algorithms Knowledge Engineer
	Applying the relation classifiers on unlabelled unstructured data to extraction relation	Unstructured data with annotated named entities and their interrelations	ML libraries Knowledge Engineer
	Ontology population to transfer the annotated entities and interrelations into the semantic knowledgebase	Semantic knowledgebase	Semantic Web Technologies and Tools Knowledge Engineer
	Investigate the online semantic datasets to enrich the Semantic knowledgebase	Enriched Semantic knowledgebase	Semantic Web Technologies and REST API Tools Domain Expert and Knowledge Engineer
4	Apply ontology reasoning and developing techniques to improve intelligent exploration and supporting the decision-making	Semantic Knowledgebase with hard-wired rules to infer new knowledge	Semantic Web Technologies and Tools Domain Expert and Knowledge Engineer
	Evaluating the Functionality of the proposed framework	The final framework	Programing Resources Domain Expert and Knowledge Engineer

According to Hebeler, et al. in (Hebeler, et al. 2011), the Semantic Web application consists of several discrete components. They fall into two major categories: major Semantic Web components and the associated Semantic Web tools. The Semantic Web components are ontology and semantic Knowledgebase. They include the activities knowledge representation and construction, which are explained in detail in previous chapters. The associated Semantic Web tools come in three types: construction tools to build and evolve a Semantic Web application, interrogation tools to explore the semantic knowledgebase, reasoners engines to add inference to and expand the semantic knowledgebase. However, there are available Semantic Web frameworks that package these tools into an integrated suite. These frameworks integrate tools that construct and manipulate a knowledgebase. They are usually composed of three basic kinds of components, storage, inference and access of knowledge. These components are interconnected to allow the interaction between them. Storage components are repositories of RDF data. Access components are usually query processors that provide the retrieval and modification of information. Inference components are reasoning engines that apply interpretation of OWL semantics to new information in the knowledgebase. To achieve effective framework functionality, we adopted Java programming language environment as the development platform to facilitate the automation and communication activities amongst the framework components and tasks.

As aforementioned in chapter 4, we employed Protégé tool for the purpose of developing the ontology of this research. Moreover, the semantic Web based application is implemented on top of Jena framework. Jena provides collections of development tools; for example, RDF data processing libraries, RDF data store system (Triple Database (TDB)), SPARQL query engine and reasoning engines including rule-based inference engine.

Application's user interface is an crucial component because it enables humans to use Semantic Web applications. It takes requests from the users and presents the responses of these requests to the user in a visual form. Web applications, in general, employs different technologies to content definition, layout definition, and User Interface logic such as HTML, CSS, and JavaScript, respectively (Pohja 2011). However, most of Semantic Web frameworks such as Jena provide an APIs and standalone SPARQL endpoint components for accessing the information in the knowledgebase.

For the developed application's user interface, we have used Jena APIs based on Java Web Application user interface technologies, which uses the HTTP protocol for communication between client and server. In addition, we linked a SPARQL endpoint to the semantic Knowledgebase in TDB store. SPARQL endpoints provide an ideal medium for retrieving semantic data. We employed Jena Fuseki server as an HTTP-based SPARQL endpoint. Fuseki has a user interface for server monitoring and administration. It provides

the SPARQL 1.1 protocols for query and update. It is tightly integrated with TDB to provide a robust, transactional persistent storage layer, and incorporates Jena text query and Jena spatial query. It can be used to provide the protocol engine for other RDF query and storage systems.

To illustrate these tools in the workflow of the developed application, we re-draw Figure 8.6 in chapter 8 above to include and show the Semantic Web tools role in the application workflow. The new figure is shown in Figure 9.1 below. This figure shows these tools in each architecture components of the semantic-based Application.

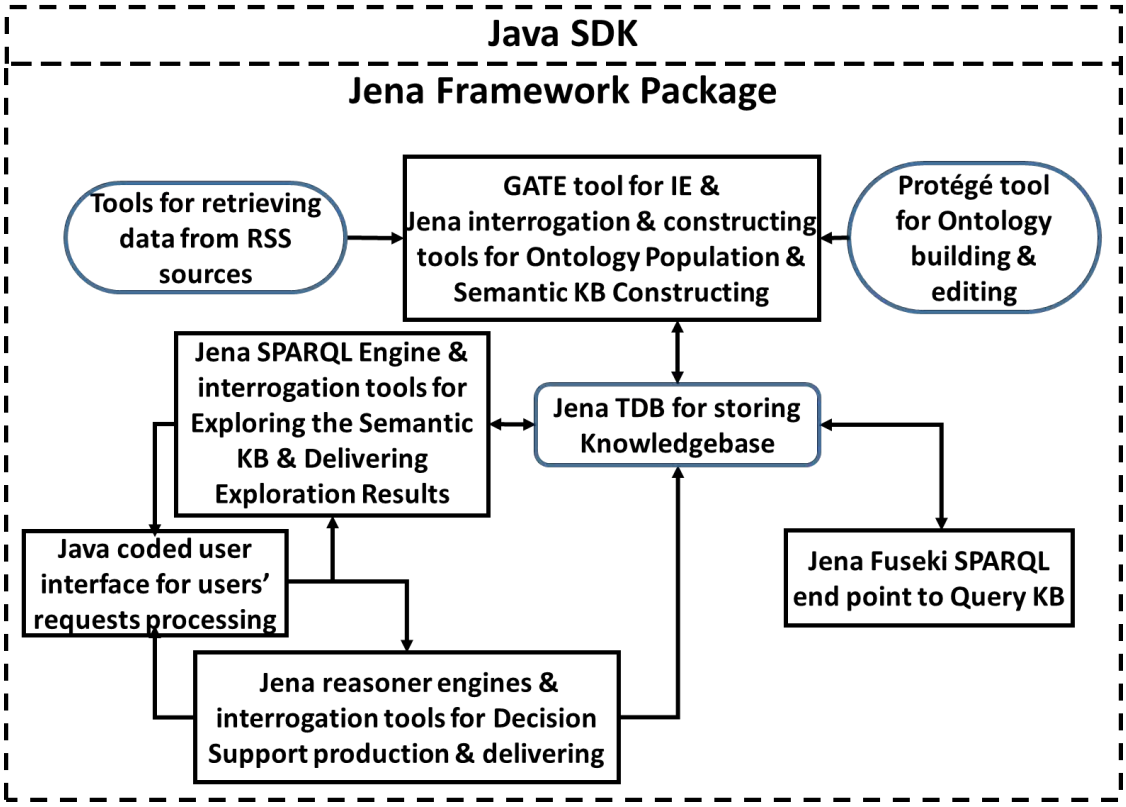

Figure 9.1: The tools in each architecture components of the semantic-based Application

The next subsection will demonstrate the aspects of knowledge representation, knowledge accessibility and the knowledge sources quality, which should be realised in Semantic Web engineering.

9.4 Semantic Knowledgebase representation:

Practically, transferring the knowledge map into model and constructing the semantic knowledgebase require a complete collaboration between knowledge engineer and domain expert in accounting for the knowledge user requirements. The knowledge engineer should consulate domain expert to decide whether the knowledge model captures all the relevant key concepts and interrelations in the domain that are required to perform the tasks for achieving the knowledge user requirements. Domain experts, in general, are responsible on employing all domain distinctions to produces a knowledge map. However, building the knowledge model, ontology, should be accomplished by knowledge engineer such as which knowledge representations approach and which reasoning techniques must be employed for which steps in the reasoning process. Knowledge map can be considered as a communication vehicle between domain expert and the knowledge engineer. The terminology used in the knowledge map is easier to be understood by non-expert of domains because the vocabularies used are clear without detailed knowledge about the particular interpreters (Van Heijst, Schreiber and Wielinga 1997).

The knowledge representation engineering should realise the fact of that the sematic knowledgebase is understandable by machines. This could be managed by the knowledge representation approaches to model the application domain knowledge in a semantic model in terms of concepts and their interrelations. The formalisation of the languages in the adopted knowledge representation approaches should allow expressiveness and inference for the knowledge. The expressiveness of these languages determines the level of accuracy in the representation of the semantics in the problem domain knowledge and the inference capability is derived from the reasoning techniques adopted by these languages. To represent the application domain knowledge, we have adopted Semantic Web Technologies to formalise it which includes languages to describe the knowledge in terms of motivating scenarios. Semantic Web Technologies and languages provide a uniform framework for capturing the semantic in the domain knowledge and offering powerful representation facilities and reasoning techniques. We employed these technologies in a range of tasks such as data modelling, reasoning and querying.

Practically, we developed a semantic model, ontology, for modelling, managing and representing the problem domain knowledge in terms of axiomatic definitions and taxonomic structures. The developed ontology has highly structured model of concepts covering the processes, objects, and attributes of that domain including their complex relations, i.e. N-ary relations. This model provides formal definitions and axioms that constrain the interpretation of these terms. The activities of developing our ontology composes are, specification, conceptualisation, formalisation and implementation. in the development of the ontology, we followed the evolving life cycle to enhance these activities. In an evolving life cycle, the developer can return from any stage to any stage of the

development process. If the ontology does not satisfy evaluation criteria and does not meet all requirements found during a specific activity, the developing that activity is revised and improved. It is worth noting that ontologies building is an iterative task, which means that their concepts, relations and axioms are improved, extended or enriched to make ontologies more precise to the growth of the domain knowledge. It does not mean the developer start over an ontology in each iteration, it only improves the existing one. Any part of the ontology that was identified as lacking quality or not meeting the desired requirements is improved.

Our developed semantic Web based application mainly relies on Semantic Web reasoning technologies including the ontology reasoning and Rule-based reasoning. The performed inference operations on the ontology are:

- Automatic Classification/Subsumption: It automatically determines if a given class is a subclass of another class (superclass) in the ontology. Accordingly, this will classify all members of the subclasses as members of their superclasses. This, in fact, will assist the basis for query processing. For example, the class Company is a subclass of the class Organization as shown in Figure 9.2 below. This means that all individual members of class Company are members of the superclass Organization.

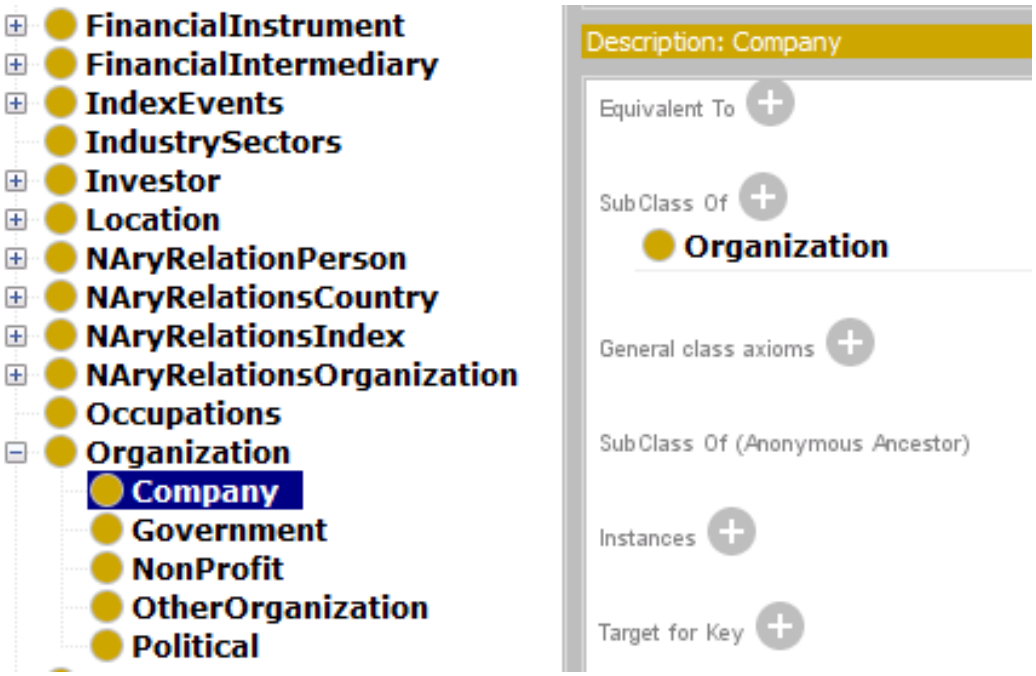

Figure 9.2: A Company class is a subclass of the Organization class

- Automatic Individuals classification: It retrieves a property fillers according to some constraints on relationships between individuals' classes to build new classes. It is achieved by restriction constructors that allow describing individual members of restricted classes in terms of constraints on relationships with other individual members of other defined classes. For example, classifying the events which are mentioned in the online news and related to a specific company are classified as positive and negative events such as the increase and decrease of profits margins in the news. Figure 9.3 below shows the axiom of classifying the companies events as positive and negative events.

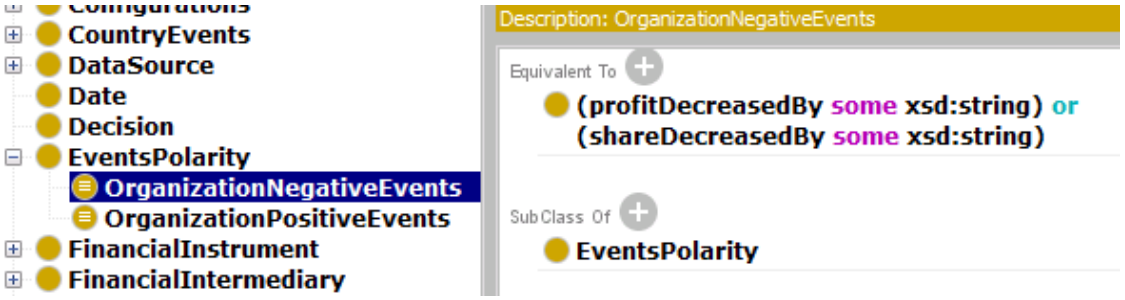

Figure 9.3: Classifying the Company's Events as Positive and Negative Events

- User-defined rules: These rules infer new facts from the existing knowledge to produce recommended stock investment decisions which are based on the available data related to the companies' stocks investment and the details of both users and their request (see section 8.4.3 above).
- Consistency checking: It determines if an ontology that has been constructed is logically consistent and no contradictions in the semantic knowledgebase. For example,
    - Checking the satisfiability of a concept by determining whether a description of the concept is not contradictory or whether an individual can exist that would be instance of the concept.
    - Checking whether the individual is an instance of a concept without violating the descriptions of the concept.

However, we performed the consistency checking task to assess knowledge quality as it is one of tasks of the knowledge representation dimension of our knowledge quality evaluation methodology. In Semantic Web model based, the model is a set of axioms. The knowledgebase should satisfy the interpretation of all axioms in the model. To check for consistency, we loaded our sample semantic knowledgebase into a Jena reasoner engine.

The Jena Reasoner engine checks whether the semantic knowledgebase is consistent. It detects if any entity is a member of disjoint classes, detects if any class or property is misplaced in the triples, detects the misuse of owl:DatatypeProperty or owl:ObjectProperty through the ontology, detects the misuse of the domain and range in the resources' properties with a certain values and detects the inconsistent values that are generated by a particular set of schema axioms for all properties in the semantic knowledgebase.

For example, assuming the relevant classes representing (kbfwo:Company) and (kbfwo:Government) and these classes are disjointed in the ontology.

kbfwo:Company → owl:disjointWith → kbfwo:Government

Let us consider that the following triples are extracted from the unstructured data and mapped into the semantic knowledgebase,

kbfwo:federal_reserve → rdf:type → kbfwo:Company
kbfwo:federal_reserve → rdf:type → kbfwo:Government

The reasoning engine will detect the conflict between these two triples; thus, the inconsistency in the semantic knowledgebase. It is because the individual (kbfwo:federal_reserve) cannot be a member of the disjointed classes (kbfwo:Company) and (kbfwo:Government) in the same time.

Figure 9.4 shows the result of performing the consistency checking task on the semantic knowledgebase by using the Validity Report tool in Jena framework and employed in the developed Semantic Web based application.

Figure 9.4: The result of checking the consistency of the semantic knowledgebase

N-ary Relation Representation:

We investigated employing Semantic Web languages to represent the N-ary relation. It appears from that investigation that N-ary relations model is not new for RDF modelling. We modelled the N-ary relation by creating an intermediate resources, Classes and Individuals (see section 4.5 above). These intermediate resources should be considered when creating axioms for classifications and patterns for querying.

For example, there is an N-ary relation triples In the sample semantic knowledgebase describing the relations between “Microsoft” company and “David Pann” employee as follows:

kbfwo:microsoft → kbfwo:employerOfNAry → kbfwo:employerofrelation39
kbfwo:employerofrelation39 → kbfwo:employerOf → kbfwo:david_pann

where:

kbfwo:microsoft → rdf:type → kbfwo:Company
kbfwo:david_pann → rdf:type → kbfwo:Employee
kbfwo:employerofrelation39 → rdf:type → kbfwo:EmployerOfRelation

The class (kbfwo:EmployerOfRelation) and its individual member (kbfwo:employerofrelation39) are the intermediate resources. Figure 9.5 below shows the classes, intermediate classes and N-ary relation properties of the above triples in the ontology.

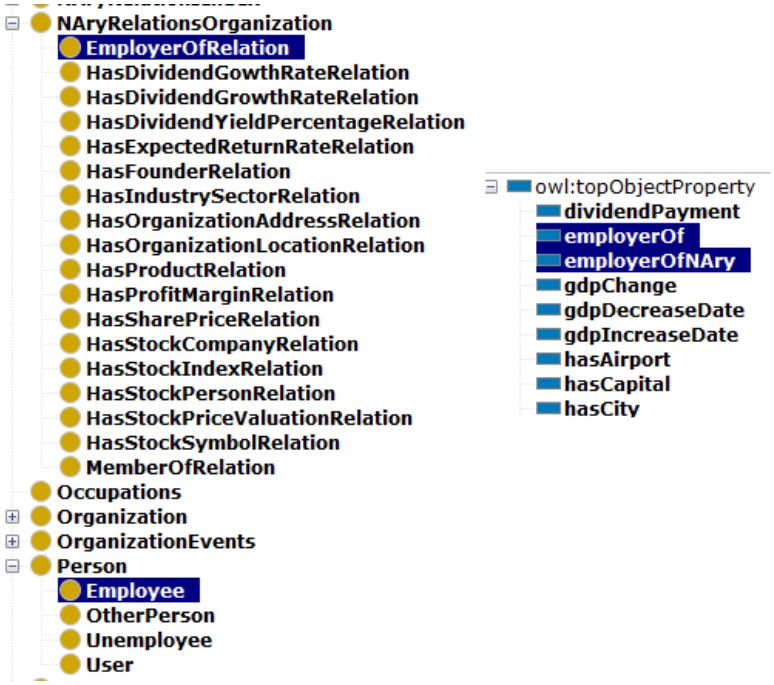

Figure 9.5: The Intermediate Classes and N-ary relation properties in the ontology

If we would like to classify the organisations which have employees as employers, we should consider these resources. For example, we can apply the OWL existential restrictions to describe the set of individuals that have at least one (kbfwo:employerOfNAry) relationship to intermediate individuals in the intermediate class (kbfwo:EmployerOfRelation). In the meanwhile, this intermediate individuals have relationships to individuals in the (kbfwo:Employee). This axiom can be written in Manchester syntax as below:

Class: kbfwo:Employer **EquivalentTo:** kbfwo:employerOfNAry **some**
(kbfwo:employerOf **some** kbfwo:Employee)

Figure 9.6 shows an axiom to classify organisations as an employers by using OWL existential restrictions in an ontology.

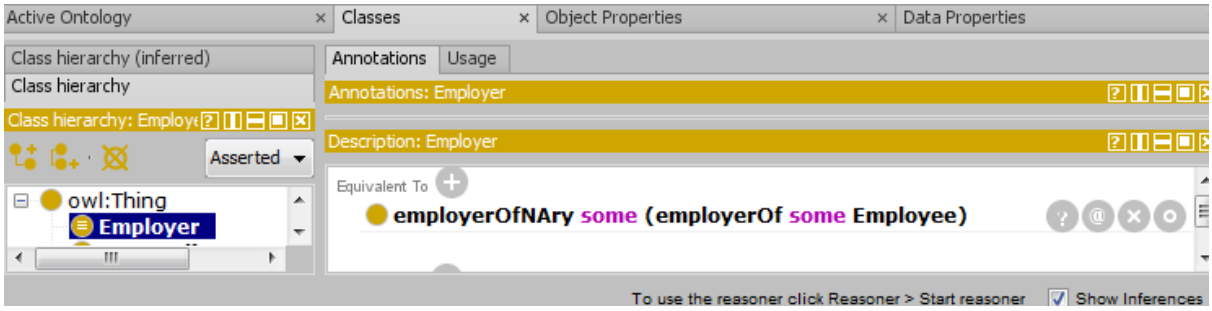

Figure 9.6: An axiom to classify organisations as an employers by using OWL existential restrictions in an ontology (This is a screenshot of Portege)

After applying the reasoner, it would derive the following statement:

kbfwo:microsoft → rdf:type → kbfwo:Employer

In addition, the intermediate resources should be considered when we would like to query N-ary relations in the semantic knowledgebase. For example, the SPARQL below:

```
SELECT DISTINCT     (str(?ms ) as ?Organisation)
                    (str(?Nary) as ?EmployerOfRelationNary)
                    (str(?Empel) as ?Employee)
WHERE {
?Org kbfwo:employerOfNAry ?Nary.
?Nary kbfwo:employerOf ?Empe.
?Org rdfs:label ?ms .
?Empe rdfs:label ?Empel.
FILTER (regex(?ms, "Microsoft" , "i"))
FILTER (regex(?Empel, "David Pann" , "i"))
}
```

Where ?Nary variable represents the intermediate resources. After executing this query by using SPARQL engine, it would retrieve the following results:

```
-------------------------------------------------------------
| Organisation | EmployerOfRelationNary      | Employee     |
=============================================================
| "Microsoft"  | "kbfwo:employerofrelation39" | "David Pann" |
-------------------------------------------------------------
```

9.5 Semantic Knowledgebase Accessibility

Semantic Web Technologies offer an opportunity for heterogeneous objects to exchange data and information in an interoperable way to make data and services machine-accessible and machine-processable. The functionalities of a Semantic Web application encompass the query activities for the extraction of data semantics from the semantic knowledgebase. Querying is a form of information discovery that allows for complex, explicit, and structured questions to be posed, and the resulting information either succeeds or fails to answer those questions. Queries offer formal interrogation of the Semantic Web data and are based on formal syntax and semantics in the semantic knowledgebase. As aforementioned, we employed Jena SPARQL engine to query the semantic knowledgebase. Jena contains a SPARQL query processor to translate SPARQL queries to a result set or graphs.

To clearly understand knowledge user needs and to acquire knowledge on a target application domain, we have conducted an interactive process for knowledge acquisition. In fact, we composed a question to capture these needs and verify that required knowledge to answer this question is represented and expressed in the knowledgebase and the required techniques to access and query this knowledge are implemented in the application. The question is:

"What are the highest five stock prices of companies which are under valuated, are there any events related to these companies in the online news and what are the situations of their counters' economies?"

We believe that the answer of this question consists of all information that supposed to be available on the sample knowledgebase. This information includes inferred and classified knowledge. The inferred knowledge is about the best companies to invest in the stock market and the situations of their counters' economies the classified knowledge is about their events that are mentioned in the online news. This question can be modelled into a SPARQL query by using different techniques. They could be:

1- Manually. Converting the natural language question into query by hand. This technique is used for general queries. It requires a SPARQL language expertise.
2- Automatically. Converting the natural language question into query by machine. This technique is used for general queries. It does not require a SPARQL language expertise.

3- Semi-automatically. Converting the question into query by machines; however, this techniques is using a specific user interface that contain a limited options to be queried. This technique is used for a specific queries. It does not require a SPARQL language expertise.

For the purpose of this research, we manually converted the natural language question above to SPARQL query.

The exemplified question is asking about two main entities, companies, and their countries. The required pieces of information about the companies are stock prices and their intrinsic value with the condition of that the stock prices are under valuated (stock price less than the intrinsic value) that supports the decision of buying the stock. Also, there another piece of information in the question that are required to be retrieved from the semantic knowledgebase, which are the classified events related to the companies in the online news. The required pieces of information about the counties are their economic indicators which are available in the semantic knowledgebase, which are GDP, Unemployment and Inflation rates. The SPARQL query code of the question is below:

```
SELECT DISTINCT
        (str(?comN) as ?CompanyName)
        (str(?price) as ?StockPrice)
        (str(?PriceDate) as ?StockPriceDate)
        (str(?value) as ?IntrinsicValue)
        (str(?con) as ?CountryName)
        (str(?GR) as ?GDPRate)
        (str(?UR) as ?UnemploymentRate)
        (str(?IR) as ?InflationRate)
        (str(?GY) as ?GDPYear)
        (str(?UY) as ?UnemploymentYear)
        (str(?IY) as ?InflationYear)
        (str(?fpctd) as ?ProfitDecreasedByPercent)
        (str(?EPDDate) as ?ProfitDecreaseDate)
        (str(?DSTnp) as ?DataSourceTitleOfProfitDecreaseNews)
        (str(?DSUnp) as ?DataSourceURLOfProfitDecreaseNews)
        (str(?dsdv) as ?DataSourceDate)
WHERE {
        ?com kbfwo:hasSharePriceNAry ?suri .
        ?com rdfs:label ?comN .
        ?suri kbfwo:hasSharePrice ?price .
        ?suri kbfwo:hasSharePriceDate ?PriceDateResource.
        ?PriceDateResource kbfwo:hasDateValue ?PriceDate.
        com kbfwo:hasStockPriceValuationNAry ?vuri .
        ?vuri kbfwo:hasStockPriceValuationValue ?value.
        ?com kbfwo:hasOrganizationLocationNAry ?olNAry.
        ?olNAry kbfwo:hasCountry ?coni.
        ?coni rdfs:label ?con.
        ?coni kbfwo:hasGDPRateNAry ?gdpNAry.
        ?gdpNAry kbfwo:hasGDPRate ?GR.
        ?gdpNAry kbfwo:hasDate ?gdpdateResource.
        ?gdpdateResource kbfwo:hasExtractedDateValue ?GY.
```

```
        ?coni kbfwo:hasUnemploymentRateNAry ?uneNAry.
        ?uneNAry kbfwo:hasUnemploymentRate ?UR.
        ?uneNAry kbfwo:hasDate ?unedateResource.
        ?unedateResource kbfwo:hasExtractedDateValue ?UY.
        ?coni kbfwo:hasInflationRateNAry ?infNAry.
        ?infNAry kbfwo:hasInflationRate ?IR.
        ?infNAry kbfwo:hasDate ?infdateResource.
        ?infdateResource kbfwo:hasExtractedDateValue ?IY.
OPTIONAL {
              ?EPD a kbfwo:OrganizationNegativeEvents.
              ?com kbfwo:profitMarginChange ?EPD.
              ?EPD kbfwo:profitDecreasedBy ?fpctd.
             ?EPD kbfwo:profitDecreaseDate ?EPDD.
             ?EPDD kbfwo:hasDateValue ?EPDDate.
             ?EPD kbfwo:hasDataSource ?datasourcenp.
             ?datasourcenp kbfwo:hasTitle ?DSTnp.
             ?datasourcenp kbfwo:hasURL ?DSUnp.
             ?datasourcenp kbfwo:hasURL ?DSUnp.
             ?datasourcenp kbfwo:hasDate ?dsd.
              ?dsd kbfwo:hasDateValue ?dsdv.
          }
FILTER (?price < ?value )
} LIMIT 1
```

Note: we limit the query to only one result and only the negative organizations' events because of the limited space and they can be done in like manner.

There are two kinds of user interfaces in the developed Semantic Web application, SPARQL endpoint by using Fuseki sever and the graphic user interface by using the tools of JAVA SDK tools and techniques. The SPARQL endpoint is not part of the application; however, it can be called from inside the application. It only accepts SPARQL query to be executed on the semantic knowledgebase in TDB store. The Results of the endpoint can be in different formats such as Text, Comma-separated values (CSV), Tab Separated Values (TSV), Extensible Mark-up Language (XML) and JavaScript Object Notation (JSON) (JSON 1999). The results of these formats will be processed separately. An example of the endpoint interface is shown in Figure 9.7 below.

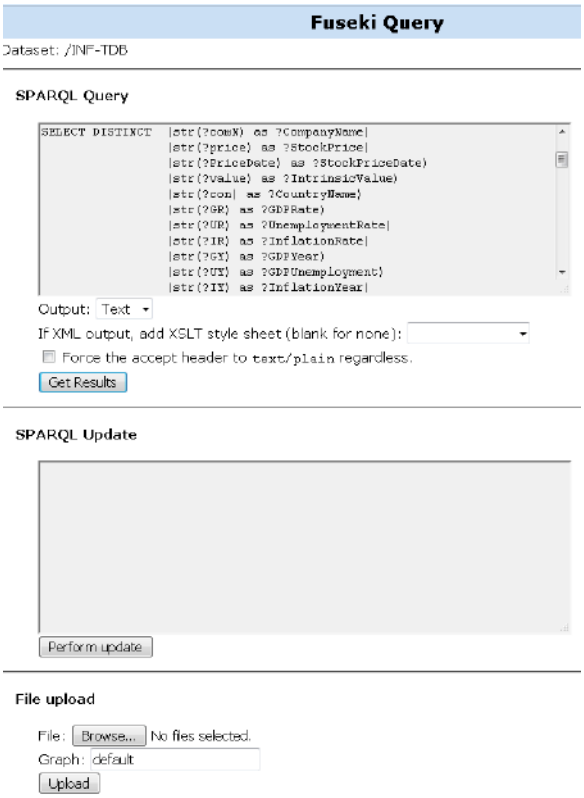

Figure 9.7: Example of Fuseki endpoint Interface

The graphic user interface is part of the application and can accept, besides the SPARQL query, any other information such as the user details. The Results of the query can processed directly by the application and presented to the user in different styles. Figure 9.8 shows an example of these interfaces.

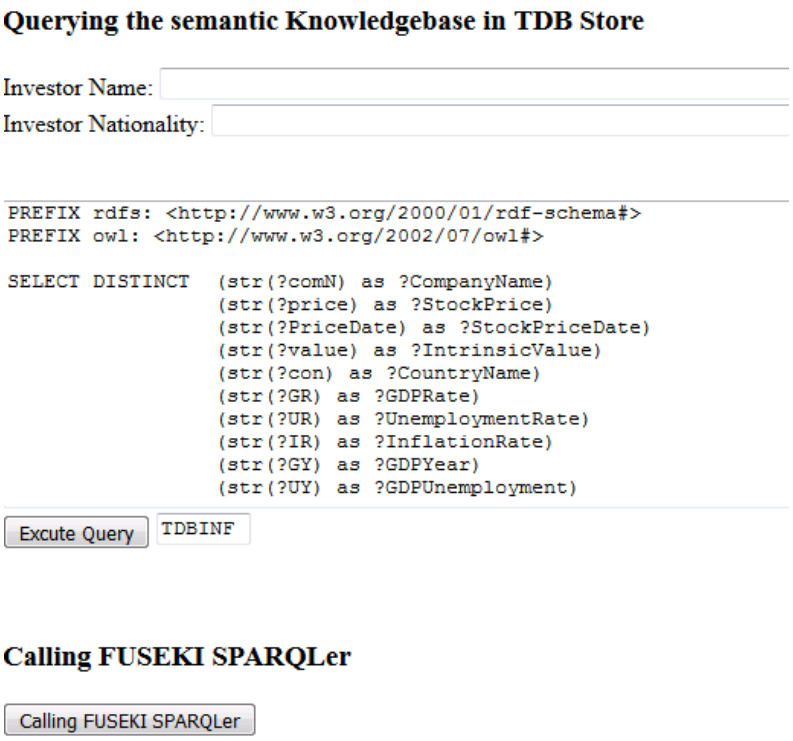

Figure 9.8: Example of SPARQL Query Entering and Executing Intercace

After executing the above query on the semantic knowledgebase in TDB store, the results will be presented according to the technique employed to operate the query. The results details of the query should be as below:

CompanyName: "Microsoft"
StockPrice: "65.22"
StockPriceDate: "2017-3-29"
IntrinsicValue: "70.01118"
CountryName: | "United States"
GDPRate: "2.59614804050973"
UnemploymentRate: "6.19999980926514"
InflationRate: "0.118627135552317"
GDPYear: "2015"
UnemploymentYear: "2014"
InflationYear: "2015"
ProfitDecreasedByPercent: "1.0"
ProfitDecreaseDate: "2016-8-16"
DataSourceTitleOfProfitDecreaseNews:"Wall Street rises with tech stocks; one eye on Fed"
DataSourceURLOfProfitDecreaseNews: "http://uk.reuters.com/article/us-usa-stocks-idUKKCN10Y1GE"
DataSourceDate: "23 Aug 2016"

Jena SPARQL engine API has different tools which can be utilised to manipulate the results and presented to user interface in appropriate style. For example,

Figure 9.9 below shows one way of presenting the details of the query results by the application.

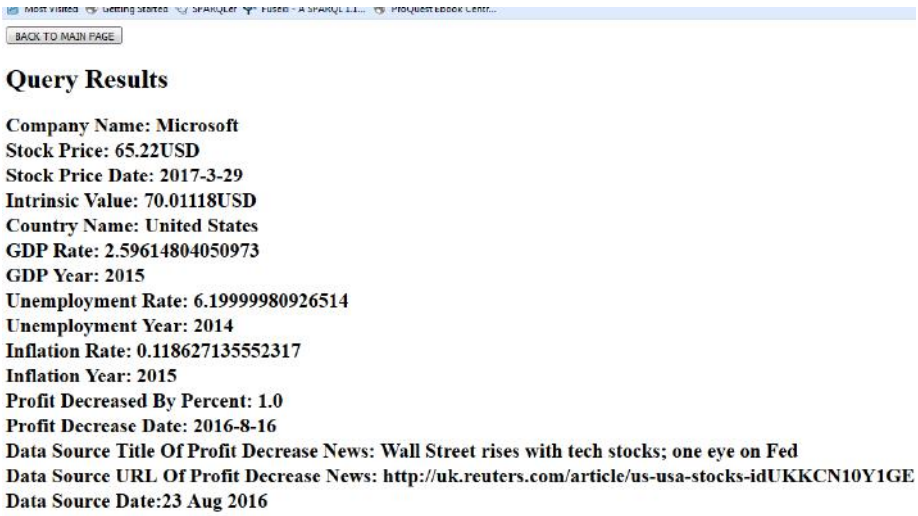

Figure 9.9: Example of displaying Query results by using Jena API

Figure 9.10 below shows the query results by using Fuseki endpoint which is presented in JSON format.

```
{
  "head": {
    "vars": [ "CompanyName" , "StockPrice" , "StockPriceDate" , "IntrinsicValue" , "CountryName" , "GDPRate" ,
"UnemploymentRate" , "InflationRate" , "GDPYear" , "UnemploymentYear" , "InflationYear" , "ProfitDecreasedByPercent" ,
"ProfitDecreaseDate" , "DataSourceTitleOfProfitDecreaseNews" , "DataSourceURLOfProfitDecreaseNews" , "DataSourceDate"
]
  } ,
  "results": {
    "bindings": [
      {
        "CompanyName": { "type": "literal" , "value": "Microsoft" } ,
        "StockPrice": { "type": "literal" , "value": "65.22" } ,
        "StockPriceDate": { "type": "literal" , "value": "2017-3-29" } ,
        "IntrinsicValue": { "type": "literal" , "value": "70.01118" } ,
        "CountryName": { "type": "literal" , "value": "United States" } ,
        "GDPRate": { "type": "literal" , "value": "2.59614804050973" } ,
        "UnemploymentRate": { "type": "literal" , "value": "6.19999980926514" } ,
        "InflationRate": { "type": "literal" , "value": "0.118627135552317" } ,
        "GDPYear": { "type": "literal" , "value": "2015" } ,
        "UnemploymentYear": { "type": "literal" , "value": "2014" } ,
        "InflationYear": { "type": "literal" , "value": "2015" } ,
        "ProfitDecreasedByPercent": { "type": "literal" , "value": "1.0" } ,
        "ProfitDecreaseDate": { "type": "literal" , "value": "2016-8-16" } ,
        "DataSourceTitleOfProfitDecreaseNews": { "type": "literal" , "value": "Wall Street rises with tech stocks; one
eye on Fed" } ,
        "DataSourceURLOfProfitDecreaseNews": { "type": "literal" , "value": "http://uk.reuters.com/article/us-usa-
stocks-idUKKCN10Y1GE" } ,
        "DataSourceDate": { "type": "literal" , "value": "23 Aug 2016" }
      }
    ]
  }
}
```


Figure 9.10: Example of Query results by using Fuseki endpoint which is presented in JSON presented in JSON format.

9.6 Semantic Knowledgebase Sources Quality

Because the success of the semantic Web based applications crucially depends on the availability of machine-understandable knowledge, Semantic Web engineering should consider modelling, extracting, maintaining and mapping the information from a diversity of sources. Not only the availability of the required information sources in the semantic knowledgebase for supporting a decision-making process can be a serious issue, but also the quality of that information.

The semantic knowledgebase in the proposed knowledge-based framework should be constructed by integrating information from different data sources, unstructured, semi-structured and structured. The availability of these data sources is an important attribute of the relevance and applicability of this framework to other problem domains. For the motivation use-case scenario of this research, the sources are, unstructured data sources which is the online news articles, structured data source which is Crunchbase dataset and semi-structured data sources which are Yahoo Finance API and Worldbank API.

The sources of the unstructured data are the relevant events published in online news articles. They contain domain-specific online economic and finance news. The investigation and experiments of improving the quality of extracting information form these online unstructured sources is presented in details in chapters 5, 6 and 7 above. Online data can

be exploited to inform data analytics and decision support systems for a variety of applications such as those belonging to the financial services domain. The structured and semi-structured data sources cover the required information for stock investment decision-making process including companies' details, real time stock prices and rates and countries' economic indicators.

We believe that the quality and the coverage of information in the constructed semantic knowledgebase from these sources is sufficient to apply and test the proposed knowledge-based framework to support the modelling, integration, navigation and presentation to guide the selection of the wrapping technologies in in the context of supporting the decision-making process. The adopted knowledge-based approach in constructing the semantic knowledgebase depends on analysing domain-specific knowledge to understand its syntactic and semantic characteristics. These characteristics aid Information Extraction process and knowledge representation activities such as reasoning about objects related to that domain.

9.7 Summary

In this chapter, we summarised our experience on addressing the challenges of implementing the proposed knowledge-based framework for constructing and exploiting a semantic knowledgebase. These challenges could be considered by domain experts and knowledge engineers as a roadmap for employing the Semantic Web Technologies for the knowledge user to intelligently exploit knowledge in similar problem domains.

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 Natural Language Processing in information extraction to ontology engineering and intelligent inferencing in knowledge representation. 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.

We confirm that Semantic Web, as an alliance between Semantic Web languages and Semantic Web applications, is powerful paradigm to represent and share Knowledge from a diversity of data sources. The architecture of Semantic Web provides the efficient foundation for developing Semantic Web applications by describing a knowledge-based framework of utilising the existing types of technology and their functionalities.

The knowledge accessibility by utilising Semantic Web Technologies 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. Investigating the tasks of reasoning, accessing and querying the semantic knowledgebase has evidenced that Semantic Web Technologies can perform an accurate and complex

knowledge representation to improve the decision-making process and the intelligent exploration of the semantic knowledgebase.

Semantic Web Technologies are very useful when working in semantic interoperability settings in discovering semantics in contents retrieved from different sources. This can be attributed to that Semantic Web technologies enable the joint exploitation of heterogeneous, distributed content by means of ontologies. For these reasons, we can prominently anticipate that the proposed framework can encourage the developers of Semantic Web based applications for any domain.

10 Conclusions and Future Work

This chapter presents an overview of the main work, the outcome contributions of the work, the PhD research limitations and proposes some ideas for further work.

10.1 Overview of the work

An 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 such as those belonging to the financial services domain. However, this online data is diverse in terms of volume and complexity, 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 techniques are needed in order to extract useful information to be represented in a machine understandable format.

The Knowledge-based to Information Extraction is based on analysing and understanding the syntactic and semantic characteristics of problem domain knowledge. These characteristics include the key concepts and their interrelations of the problem domain, the grammar and the meaning of words in the context of sentence structure or the style of the documentation language. They inform Information Extraction process by assisting in generating the linguistic and structural features to recognise Named Entitles and extracting the relations between named entities. Also, they inform knowledge representation activities by considering the sentence structure to model and reason about objects related to that domain.

In this research, the semantic and syntactic characteristics of domain knowledge were exploited in improving Natural Language Processing tasks associated with the instances labelling and feature generation processes in our implementation of Machine Learning 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.

Knowledge representation allows the structuring of information extracted from a targeted problem domain so that it can be interpreted and reasoned upon and interpreted by machines. There are a diversity of Knowledge Representation approaches; however,

Semantic Web Technologies were adopted as a knowledge representation approach. It is because they are a powerful paradigm to access, use and share information such as inference and validation. These technologies include: semantic Web models (ontologies), Semantic Web Languages (RDF, RDFS, OWL) and Semantic Web Query Languages (SPARQL). Ontology is a formal explicit description of the targeted domain knowledge and it plays a key role in Semantic Web knowledge representation. Ontologies have well-defined syntax, that offer expressive power and convenience of expression, and support advanced reasoning methods. The resultant structured data can be reasoned upon to deliver intelligent query methods against the information and the underlying metadata.

A comprehensive knowledge-based framework were presented for exploiting domain knowledge in constructing a semantic knowledgebase for a target problem domain. The semantic knowledgebase allows for the intelligent inference and advanced interrogation of information from the target domain. The proposed framework has a roadmap of integrating several components of different techniques and tools. It focuses on providing reusable and configurable data and application templates, which allow the users to apply it in a diversity of domains. It, also, allows the application developers to focus on domain problems rather than the tools and techniques of the application. In addition, it covers a diversity of disciplines and techniques that include the automatic Information Extraction from unstructured data, constructing a semantic knowledgebase from different sources, enriching the resultant semantic knowledgebase by sourcing appropriate semi-structured and structured datasets, and consuming the resultant semantic knowledgebase by intelligent exploration and support decision-making. For the purpose of implementing and evaluating the proposed framework, stock investment activities in the financial domain were employed as a use-case scenario to investigate extracting and exploring information.

In the initial phase of the proposed framework, how Semantic Web Technologies were utilised to model the targeted domain knowledge was described. ontology engineering was utilised to describe and combine the corresponding relation between the concepts' instances from different sources and infer new information about these concepts in different contexts and enable the sharing and reuse of domain knowledge. Ontology building is a process that comprises of a number of stages including specification, conceptualisation, formalisation and implementation. However, the targeted problem domain in this research is heavily represented by non-binary relations because it is characteristically represented by facts that involve more than two entities, usually called N-ary relations. As a result, a relation centred or a relation-as-class pattern was adopted to represent these domain-specific non-binary relations or N-ary relations.

In the second stage of the proposed framework, the information is extracted from unstructured data of online news. Extracting information from unstructured data requires applying Natural Language pre-Processing tasks in order to obtain the appropriate linguistic

features to be used to extract valuable information from natural language texts. Information Extraction is a pipeline process. This process is started in recognising the named entities; then, identifying identity relation between named entities, which is co-references resolution; lastly, extracting the relation between the named entities in a certain event. For Natural Language pre-Processing and Named Entity Recognition tasks, the Rule-based ANNIE pipeline system in the GATE NLP engine was utilised. For relation extraction, a hybrid approach of integrating Rule-based and Machine Learning based techniques was adopted. Our approach that relies on Rule-Based techniques for extracting relation instances and generating features vectors from the input unstructured data; subsequently, supervised Machine Learning techniques are utilised for relation classification based on named entities' relation instances and their feature vectors. With respect to relation classification tasks, three ML classifiers were implemented, configured and evaluated. They are commonly adopted for relation extraction from unstructured text, which are SVM, PAUM and KNN.

The relation classification models were further boosted by our implementation of GAs as wrapper approach to reduce the feature space. The configuration parameters of GAs require tuning to find the best fit for a specific optimisation problem. the optimum values were heuristically established for the GA's initial population size, the number of generations, crossover rate and mutation rate that represent the best fit for our features selection problem for relation classification. Our implementation of GAs has resulted in significant improvement in the accuracy of the ML based Relation Extraction process. Furthermore, GAs were compared against a space search algorithm that has similar operational dynamics, Random Mutation Hill-Climbing (RMHC) to verify that GAs are an appropriate choice for optimising the process of features selection for the relation classification problem. In order to further examine any significant difference in the performance of our implementation of GAs and Random Mutation Hill-Climbing algorithm, a non-parametric statistical procedure, the Wilcoxon test, was used to detect if there is a significant difference among the behaviour of the sample runs of our algorithms' implementations.

In the third stage of the proposed framework, the semantic knowledgebase was constructed. The process of constructing the knowledgebase was implemented in three stages, Information Extraction, ontology population and knowledgebase enrichment. After building the relation classification models by using the configured training datasets and the best selected features vectors on SVM, these models were applied onto the pre-processed unlabelled online financial news documents to extract and annotate new relations between the targeted annotated entities. The annotated entities and their interrelations are related to domain concepts and properties in the semantic model (ontology), to construct a semantic knowledgebase. The resultant semantic knowledgebase is further enriched by utilising a diversity of structured and semi-structured data sources. I believe that the formalism in modelling the semantic knowledgebase provides a great opportunity to leverage domain-relevant facts that are published in structured and semi-structured data sets.

In the fourth 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. Semantic Web Technologies were adopted to interrogate the resultant semantic knowledgebase by implementing the proposed knowledge-based framework as an application. This application integrates the semantic knowledgebase exploration and Decision Support System 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 explore the semantic knowledgebase to make the decision by themselves. The application receives and processes the user's request, delivering the request answer. The request answer could be producing the recommended decision or exploring the semantic Knowledgebase.

As a part of the framework, OWL reasoning have been applied to achieve many tasks such as automatic class subsumption and automatic Individuals classification. Moreover, rule-based reasoning was utilised to develop decision-rules based on the ontology and executed them to derive stock investment specific recommendations. In addition, SPARQL was employed to explore the semantic knowledgebase because it is capable to support advanced exploration scenarios and solve complex information needs such as supporting the decision-making process by allowing users to express complex queries across diverse RDF data sources. Moreover, in consideration of adopting N-ary relations patterns requirements to represent non-binary relations in the problem domain, the reasoning axioms and SPARQL queries were adapted to fit the intermediate resources in the N-ary relations requirements.

Implementing phases of the proposed framework requires the development and integration of processes that utilise a number of constantly evolving technologies ranging from using Natural Language Processing in information extraction to ontology engineering and intelligent inferencing in knowledge representation. 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. Our experience on addressing the challenges of implementing the proposed knowledge-based framework for constructing and exploiting a semantic knowledgebase could be considered by domain experts and knowledge engineers as a roadmap for employing Semantic Web Technologies for the knowledge user to intelligently exploit knowledge in similar problem domains. In the process of evaluating the knowledge representation and knowledge accessibility, they have been assessed if they meet the knowledge users need in a specific use case scenario. It has been confirmed that Semantic Web, as an alliance between Semantic Web languages and Semantic Web applications, is powerful enough to represent and share Knowledge from a diversity of data sources.

Next subsection will present the thesis contributions after the investigation of the challenges in the implementation the knowledge-based framework.

10.2 Thesis Contributions To Knowledge

The main novel outcome of this thesis 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 Semantic Web Technologies to intelligently explore knowledge in other and similar problem domains. The roadmap integrates contributions at the algorithmic and implementation level to different disciplines including Information Extraction, Machine Learning, Evolutionary Algorithms and Knowledge representation to allow the application developers emphasising their efforts on domain problems.

Also, in the course of this thesis, other valuable contributions to knowledge were produced. The following list presents recaps the most important ones:

1) Employing Knowledge-based approach in Information Extraction and Knowledge Representation (Research Question 1 and Research Question 3).

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 problem domain knowledge in solving these challenges. Analysing and understanding the syntactic and semantic characteristics of the problem domain knowledge did benefit Information Extraction and knowledge representation in this research. These characteristics aid in Natural Language Processing tasks associated with automating or semi-automating instance labelling process. For instance, in our implementation of Machine Learning based relations classification, domain-specific knowledge is used to compile some of our 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. In addition, the syntactic and semantic characteristics of the problem domain knowledge aid in the process of semantically modelling this domain knowledge by capturing the key concepts and their interrelations that are related to the problem domain and to understand how they have been used in the domain knowledge to be transferred into ontology.

Furthermore, they aid knowledge representation activities by considering the sentence structure to reason about objects related to that domain.

2) Adopting an N-Ary Relation pattern for representing Non-Binary relations (Research Question 6).

Relation-centred or relation-as-class pattern were adopted to represent domain-specific N-ary relations or non-binary relations in the resultant semantic knowledgebase. In this pattern solution, the N-ary relation is transferred into multi-binary relations. This pattern uses an intermediate resource to represent the main N-ary predicate as an individual member of an intermediate class with "N" properties that provides additional information about the relation instances. The intermediate resources are flexible in describing the relationships between resources. In this pattern, any number of additional properties may be used to describe the relation, whether this relation is between two resources or between several resources.

Our findings revealed that the N-ary relation patterns are a very important for non-binary relations in a variety of domains and that the existing Semantic Web Technologies and languages can be employed to represent those patterns. Representing N-ary relation patterns in semantic knowledge bases is clearly domain independent and can be applied across multiple application domains. However, there are some considerations that should be taken when introducing a new intermediate class for a relation. The first consideration is that meaningful names should be given to instances of properties or to the classes used to represent instances of N-ary relations. The second consideration is that the inverse N-ary relations requires defining inverse properties for all properties that are involved in the N-ary relations. The last consideration is that expressing the N-ary relation in terms of OWL axioms should consider the intermediate resources. The novelty in the approach of adopting relation-as-class pattern to represent N-ary relations in this research is derived from not using the intermediate resources as blank nodes; alternatively, they have been identified by URI references to avoid the negative impact of blank nodes on the representation of the Semantic Web data (See section 1.3.1.1 and section 4.5.2.1).

3) Configuring the ML algorithms for Relation Classification Problem (Research Question 2).

- For the purpose of ML algorithms' parameter optimisation and improving the ML classifiers' performance, a grid-based manual search approach was adopted to perform parameter tuning, which proved sufficient to satisfy the requirements of the deployed ML techniques (SVM, PAUM, KNN); a grid-based search is simple to implement compared to the computationally expensive automatic optimisation techniques. Adapting ML algorithms' parameters is a critical task in tuning general-purpose algorithms to solve different domain-specific problems. The parameters' values, which are selected by grid search, proved favourable to the traditionally

accepted default values for the SVM, PAUM and KNN algorithms to classify relations in unstructured data.

- In order to further enhance the accuracy of the relation classification models, by means of experimentation, the best probability threshold values were heuristically determined for all classification models on all training datasets by drawing on the correlation between the threshold probability classification and F1-measure. Experimental results showed that the empirically selected values deliver better classification accuracy compared to the default threshold value. Hence, I believe that the the probability threshold should be investigated when creating classification models, in particular for the relation classification problem.
- Macro-averaging was considered more appropriate for evaluating the classification accuracy for the problem domain since the sourced financial news articles represent independent documents. Precision, recall and F1-measure were computed for individual documents and then averaged for the entire corpus.

4) Reducing the training datasets' imbalances (Research Question 2).

The utilisation of distant supervision for the compilation of the training data ground facts can result in incorrectly labelling a considerable number of relations as negative instances thus disrupting the balance between True Positive and True Negative instances of the classes in the training datasets. Hence, a number of experiments were conducted to heuristically reduce the number of resulting negative instances; also, some negative relation instances in the training datasets of one relation class were explicitly introduced in order to decrease the true positive rate while maintaining a low false positive rate. The experimental results evidenced that our approach has a positive impact on the models' accuracy.

5) Fitting the GAs' operations and parameters to the relation classifiers' features selection problem (Research Question 4 and Research Question 5).

- GAs as wrapper approach were utilised to optimise the ML features selection and the experimental results proved that all of the studied relation classifiers perform significantly better in the reduced feature space.
- The configuration parameters of GAs require tuning to find the best fit for a specific optimisation problem. By means of experimentation, the optimum values were heuristically established for the GA's initial population size, the number of generations, crossover rate and mutation rate that represent the best fit for our features selection problem for relation classification.
- In terms of selecting the best features for relation classification, the research findings indicate that the models that are created using the Named Entity category combined with lexical and/or syntactic features, exhibit better accuracy. The exception for our target domain is the Stock Symbol and Organization relation as it is characterised with short relation mentions (instances) in terms of the number of words.

- Due to the modest search space of the target domain and the predominantly linguistic characteristic of the features, our attempt to further improve the performance of the GA by reducing their search space through features grouping did not result in a significant improvement. However, I believe that exploring the similarities and interrelations between features could yield better results for other domains with larger search space and different feature types.
- The finding provides evidence that GAs are an appropriate choice for optimising the process of features selection for the relation classification problem. The implementation of GA in this research were compared against Random Mutation Hill-Climbing (RMHC) by using a non-parametric statistical procedure, Wilcoxon test. Our findings demonstrated that our implementation of GAs for feature selection outperforms the Random Mutation Hill-Climbing algorithm in terms of improving relation classifiers accuracy.

6) Utilising the domain-relevant public online datasets to aid Information Extraction and Enriching the resulting semantic knowledgebase (Research Question 7).

Semantic Web Technologies were adopted because there are public datasets available online (see section 4.2.2 and section 8.2.3) that adopt the same Semantic Web standards. These datasets are relevant to various problem domains and their contents can be used to aid Information Extraction and to enrich the resulting knowledgebase.

Public LOD datasets (DBpedia and Freebase) have been employed as distant supervision sources to our ML algorithms as these datasets are similar to our knowledge modelling approach, these datasets use the same standardised semantic formalism to publish ground facts that are relevant to our problem domain. The ground facts were used to compile training datasets for relation classification. Also, they are used to collect gazetteer lists to aid Named Entity Recognition.

The resulting sematic knowledgebase 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. These public datasets are leverage the domain-relevant facts and improve the intelligent exploration to support decision-making process.

In summary, the above contributions are produced during the implementation of the proposed framework. Most of the problems and challenges of implementing the framework were investigated, which are modelling the problem domain knowledge, the problem of optimising the relation classifiers, the issue of representing the non-binary relations by using Semantic Web languages and reasoning, accessing and querying the resultant semantic knowledgebase. The investigation of these problems was undertaken by using stock investment decision-making process as use-case scenario. The results of these investigations are an evidence for that Semantic Web Technologies can perform an

accurate and complex knowledge representation to improve the decision-making process and the intelligent exploration of the semantic knowledgebase. The formalism in modelling the semantic knowledgebase provides a great opportunity to leverage domain-relevant facts that are published in structured and semi-structured data sets. The architecture of the Semantic Web provides the efficient foundation for developing semantic Web applications by describing a knowledge-based framework of utilising the existing types of technology and their functionalities. The evaluation of knowledge accessibility by utilising Semantic Web Technologies 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.

10.3 The PhD Research Limitation and Plans for Further Work

In this section, some of limitations to this study have been acknowledged and further work based on these limitations have been suggested. Fundamentally, this thesis 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 were developed for investigating the implementation challenges of that framework. Naturally, I did not intend to produce a complete commercial application; nevertheless, the framework attempts to cover most issues and techniques required to implement the knowledge-based framework. As a result, it is recommended that further research might be conducted to explore the following areas that this research effort does not engage with. Below is a list of these suggested further work:

1. Considering the problem of inaccurate annotation:

Extracting information from natural language texts is a very complex task because real-world data is noisy and often suffers from corruptions that may impact data understanding and modelling. This can cause inaccurate labelling and inaccurate classification. Another factor that can cause the same is the accuracy of Information Extraction techniques and tools. The accuracy of the techniques and tools can vary depending on the considered type of entity, class of text and the complexity of the targeted problem domain. Alleviating this problem needs data cleaning by catching and fixing corruptions in the data before applying the Information Extraction tools or by cleaning the extracted information after populating it into the semantic knowledge (Tang 2015, Hu, et al. 2012, Feilmayr 2011).

Although this issue is beyond the scope of this study, it is worth to mention that the inaccurate extracted information in the resultant semantic knowledgebase should be considered specifically in Decision Support Systems.

2. Investigating issues related to the dynamic update of the Knowledgebase:

Online data in a majority of domains is often subjected to change and evolves over time due to the dynamic nature of knowledge. For example, new facts are becoming known while some of the older ones need to be revised and/or retracted at the same time. For example, the head manager of a company is changed or died. This evolution should be addressed by adding new facts to the knowledgebase (Toledo, Chiotti and Galli 2012, Nováček, et al. 2008). Although this issue is beyond the scope of this study, it is worth to mention that the Information Extraction component of the proposed Semantic Web application should be designed to be dynamic in terms of the ability of making the semantic Knowledgebase constantly up-to-date. This update can be achieved periodically or on demand regarding the application consumers requests and the targeted domain knowledge. For example, the economic and finance domain is a dynamic domain and causes a rapid information influx; as a result, the process of decision-making support and knowledge exploration should be achieved regarding a newly extracted information (Li, et al. 2014a).

3. Sentiment analysis for supporting the decision-making process

According to Wu, et al. in (Wu, Zheng and Olson 2014), there are several studies showing that the sentiment information contained in the published financial online data could be used to make beneficial stock investment decisions. Therefore, Opinion Mining and Sentiment Analysis could be integrated in the recommended systems for processing the automatic collected information to support decision-making process. This thesis does not engage with a full discussion of Opinion Mining and Sentiment Analysis because they lie beyond the scope of this study. However, further work investigating the integration of Opinion Mining and Sentiment Analysis in Decision Support Systems would be very interesting.

4. Fuzzy RDF in the Semantic Web.

Another possible research direction is the inclusion of fuzziness and probabilities for facts in the semantic Knowledgebase. There is a growing interest in a very common requirement in real world applications that in the development of knowledge representation formalisms able to deal with uncertainty (Rodríguez, et al. 2014). For example, stock investment decision-making is a complex process that is influenced by several interrelated factors, characterised by inherent nonlinearities. Stock investment decisions are preferably made with a certain level of a truth rather than crisp investment decision. As a result, I believe that stock investment Decision-Making processes can be successful only with the use of tools and techniques that can overcome the problem of uncertainty, noise and nonlinearity of

data (Chourmouziadis and Chatzoglou 2016). Since fuzzy set theory and fuzzy logic are suitable formalisms to handle these types of knowledge, further research might explore the integration of them into Semantic Web Technologies to allow expressing fuzzy concepts and axioms in the ontology.

5. The Exploration and Visualisation of the semantic knowledgebase

In the previous suggested further works, several areas have been covered; for example, Information Extraction, knowledge updating and decision-making support. However, a further study with more focus on knowledge exploration and visualisation is suggested in this section because the question of how typical end-users can access this body of knowledge becomes of crucial importance. The efforts should facilitate end-users interaction with the semantic knowledge to assist them to learn and make sense of complex and heterogeneous data. The user interfaces techniques should hide the complexity of formal query languages for regular end-users in the meanwhile those end-users take advantage of using exploration and visualisation techniques. In addition, this further research should investigate how to support end-users in situations where the knowledge has complex elements that require user interpretation during the exploration process. For example, how to support the end-users' search task when they are not familiar with the search domain or they do not have sufficient knowledge about the domain to make a query and how to support the navigation in large knowledge bases (Thakker, Yang-Turner and Despotakis 2016, Fafalios and Tzitzikas 2013). I believe that this suggested further work should focus on how semantic knowledgebase is explored by end-users and how the result of the exploration is visualised to them.

References

Acampora, G., Pedrycz, W. and Vitiello, A., 2015. A Competent Memetic Algorithm for Learning Fuzzy Cognitive Maps. *Fuzzy Systems, IEEE Transactions On,* 23 (6), 2397-2411.

Agrawal, A., Viktor, H.L. and Paquet, E., 2015. SCUT: Multi-Class Imbalanced Data Classification using SMOTE and Cluster-based Undersampling. *In: The 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Lisbon-Portugal, November 12-14, 2015.* SCITEPRESS–Science and Technology Publications, pp. 226-234.

Agrawal, J., Chourasia, V. and Mittra, A., 2013. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* 2 (4), 1360-1366.

Al-Ajlan, A., 2015. The Comparison between Forward and Backward Chaining. *International Journal of Machine Learning and Computing,* 5 (2), 106.

Alatrish, E., 2013. Comparison Some of Ontology. *Journal of Management Information Systems,* 8 (2), 018-024.

Aljamel, A., Osman, T. and Acampora, G., 2015. Domain-specific Relation Extraction: using distant supervision Machine Learning. *In: The 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Lisbon, Portugal, November 12-14, 2015.* Lisbon, Portugal: SCITEPRESS, pp. 92-103.

Allami, R., Stranieri, A., Balasubramanian, V. and Jelinek, H.F., 2016. A genetic algorithm-neural network wrapper approach for bundle branch block detection. *In: Computing in Cardiology Conference (CinC), Vancouver, BC, Canada, 11-14 Sept. 2016.* Vancouver, BC, Canada: IEEE, pp. 461-464.

Allemang, D., and Hendler, J., 2011. *Semantic web for the working ontologist: effective modeling in RDFS and OWL.* Second Edition ed. Elsevier.

Aly, M., 2005. Survey on multiclass classification methods. *Neural Netw,* , 1-9.

Amardeilh, F., Kraaij, W., Spitters, M., Versloot, C. and Yurtsever, S., 2013. Semi-automatic ontology maintenance in the virtuoso news monitoring system. *In: Intelligence and Security Informatics Conference (EISIC), 2013 European,* IEEE, pp. 135-138.

Ameen, A., Khan, K.U.R. and Rani, B.P., 2014a. Extracting knowledge from ontology using Jena for semantic web. *In: Convergence of Technology (I2CT), 2014 International Conference for,* IEEE, pp. 1-5.

Ameen, A., Khan, K.U.R. and Rani, B.P., 2014b. Reasoning in Semantic Web Using Jena. *Computer Engineering and Intelligent Systems,* 5 (4), 39-47.

Ameen, A., Khan, K.U.R. and Rani, B.P., 2012. Creation of ontology in education domain. *In: Technology for Education (T4E), 2012 IEEE Fourth International Conference on,* IEEE, pp. 237-238.

Amiri, A., Ravanpaknodezh, H. and Jelodari, A., 2016. Comparison of stock valuation models with their intrinsic value in Tehran Stock Exchange. *Marketing and Branding Research,* 3 (1), 24.

Anbarasi, M., Anupriya, E. and Iyengar, N., 2010. Enhanced prediction of heart disease with feature subset selection using genetic algorithm. *International Journal of Engineering Science and Technology,* 2 (10), 5370-5376.

Andrew, G., and Gao, J., 2007. Scalable training of L 1-regularized log-linear models. *In: Proceedings of the 24th international conference on Machine learning,* ACM, pp. 33-40.

Anu, 2013. Improved Performance of Replacement Strategies in GA. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3 (9), 344-346.

Appelt, D.E., 1999. Introduction to information extraction. *Ai Communications,* 12 (3), 161-172.

Aranguren, M.E., Antezana, E., Kuiper, M. and Stevens, R., 2008. Ontology Design Patterns for bio-ontologies: a case study on the Cell Cycle Ontology. *BMC Bioinformatics,* 9 (5), 1.

Archibald, R., and Fann, G., 2007. Feature selection and classification of hyperspectral images with support vector machines. *IEEE Geoscience and Remote Sensing Letters,* 4 (4), 674-677.

Arenas, M., Gottlob, G. and Pieris, A., 2014. Expressive languages for querying the semantic web. *In: Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems,* ACM, pp. 14-26.

Atdağ, S., and Labatut, V., 2013. A comparison of named entity recognition tools applied to biographical texts. *In: Systems and Computer Science (ICSCS), 2013 2nd International Conference on,* IEEE, pp. 228-233.

Bagherzadeh-Khiabani, F., Ramezankhani, A., Azizi, F., Hadaegh, F., Steyerberg, E.W. and Khalili, D., 2016. A tutorial on variable selection for clinical prediction models: feature selection methods in data mining could improve the results. *Journal of Clinical Epidemiology,* 71, 76-85.

BBC, C., 2018. *BBC Stock Market news RSS Link* [online]. BBC Co. Available at: http://feeds.bbci.co.uk/news/business/rss.xml [Accessed 03/02 2014].

Beck, H., and Pinto, H.S., 2002. Overview of approach, methodologies, standards, and tools for ontologies. *Draft Paper, the Agricultural Ontology Service, UN FAO,* .

Benetka, J.R., Balog, K. and Nørvåg, K., 2017. Towards Building a Knowledge Base of Monetary Transactions from a News Collection.

Bergstra, J., and Bengio, Y., 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research,* 13, 281-305.

Bhavsar, H., and Ganatra, A., 2012. A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE),* 2 (4), 2231-2307.

Blomqvist, E., 2014. The use of Semantic Web technologies for decision support–a survey. *Semantic Web,* 5 (3), 177-201.

Bock, J., Haase, P., Ji, Q. and Volz, R., 2008. Benchmarking OWL reasoners. *In: ARea2008-Workshop on Advancing Reasoning on the Web: Scalability and Commonsense,* Tenerife, .

boilerpipe, 2014. *boilerpipe* [online]. Google. Available at: https://code.google.com/p/boilerpipe [Accessed 5/20 2014].

Booth, D., 2013. Well Behaved RDF: A Straw-Man Proposal for Taming Blank Nodes.

Brester, C., Kauhanen, J., Tuomainen, T., Semenkin, E. and Kolehmainen, M., 2016. Comparison of Two-Criterion Evolutionary Filtering Techniques in Cardiovascular Predictive Modelling. *In: Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics - (Volume 1), Lisbon, Portugal, July 29-31, 2016.* Portugal: SCITEPRESS Digital Library, pp. 140-145.

Bunakov, V., 2015. Use Cases for Triple Stores and Graph Databases in Scalable Data Infrastructures. *In: DAMDID/RCDL,* pp. 37-40.

Buranarach, M., Supnithi, T., Thein, Y.M., Ruangrajitpakorn, T., Rattanasawad, T., Wongpatikaseree, K., Lim, A.O., Tan, Y. and 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 (01), 115-145.

Buzdalov, M., Yakupov, I. and Stankevich, A., 2015. Fast implementation of the steady-state NSGA-II algorithm for two dimensions based on incremental non-dominated sorting. *In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation,* ACM, pp. 647-654.

Cao, Y., Wang, X., Zhang, F. and Yang, W., 2012. Ontology-Based Domain Knowledge Acquisition Technology. *In: Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on,* IEEE, pp. 487-490.

Capadisli, S., 2014. *SPARQLer Linked Data endpoint of World Bank Data - General purpose processor* [online]. The World Bank Group. Available at: http://worldbank.270a.info/sparql [Accessed 11/2014 2015].

Cashell, B., 2006. Economic Growth, Inflation and Unemployment: Limits to Economic Policy. *In: CRS Report for Congress, .*

Castells, P., Foncillas, B., Lara, R., Rico, M. and Alonso, J.L., 2004. Semantic web technologies for economic and financial information management. *In: European Semantic Web Symposium,* Springer, pp. 473-487.

Castro, A.G., Rocca-Serra, P., Stevens, R., Taylor, C., Nashar, K., Ragan, M.A. and Sansone, S., 2006. The use of concept maps during knowledge elicitation in ontology development processes–the nutrigenomics use case. *BMC Bioinformatics,* 7 (1), 267.

Chandrashekar, G., and Sahin, F., 2014. A survey on feature selection methods. *Computers & Electrical Engineering,* 40 (1), 16-28.

Chen, K., Yin, J. and Pang, S., 2017. A design for a common-sense knowledge-enhanced decision-support system: Integration of high-frequency market data and real-time news. *Expert Systems,* 34 (3).

Chen, L., Zhang, H., Chen, Y. and Guo, W., 2012. Blank Nodes in RDF. *Jsw,* 7 (9), 1993-1999.

Chourmouziadis, K., and Chatzoglou, P.D., 2016. An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Systems with Applications,* 43, 298-311.

Clark, J.H., and González-Brenes, J.P., 2008. Coreference resolution: Current trends and future directions. *Language and Statistics II Literature Review,* , 1-14.

CNNMoney, R., 2018. *CNN Money Stock Market news RSS Link* [online]. CNN Co. Available at: http://rss.cnn.com/rss/money_markets.rss [Accessed 03/2 2014].

Corcho, O., Fernández-López, M. and Gómez-Pérez, A., 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering,* 46 (1), 41-64.

Corsar, D., and Sleeman, D.H., 2008. Developing Knowledge-Based Systems using the Semantic Web. *In: BCS Int. Acad. Conf.* pp. 29-40.

Costantino, M., Morgan, R.G., Collingham, R.J. and Carigliano, R., 1997. Natural language processing and information extraction: Qualitative analysis of financial news articles. *In: Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997,* IEEE, pp. 116-122.

Crunchbase, C., 2018. *Crunchbase Marketplace API* [online]. Crunchbase. Available at: https://www.crunchbase.com/#/home/index [Accessed 01/10 2016].

Cunningham, H., 2005. Information extraction, automatic. *Encyclopedia of Language and Linguistics,* , 665-677.

Cunningham, H., Maynard, D. and Bontcheva, K., 2014. *Developing Language Processing Components with GATE Version 8, University of Sheffield Department of Computer Science.* 8th ed. Sheffield, UK: Gateway Press CA.

Daelemans, W., and Hoste, V., 2002. Evaluation of machine learning methods for natural language processing tasks. *In: 3rd International conference on Language Resources and Evaluation (LREC 2002),* European Language Resources Association (ELRA), .

Damodaran, A., 2012. *Investment valuation: Tools and techniques for determining the value of any asset.* Third Edition ed. John Wiley & Sons.

Davies, J., Studer, R. and Warren, P., 2006. *Semantic Web technologies: trends and research in ontology-based systems.* John Wiley & Sons.

Davis, J., and Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves. *In: Proceedings of the 23rd international conference on Machine learning,* ACM, pp. 233-240.

DBpedia Team, 2015. ***The DBpedia Knowledge Base*** [online]. Open Community Project. Available at: www.dbpedia.org [Accessed 2/1 2014].

De Marneffe, M., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J. and Manning, C.D., 2014. Universal Stanford dependencies: A cross-linguistic typology. *In: LREC 2014, Ninth International Conference on Language Resources and Evaluation, Reykjavik, Iceland, May 26-31, 2014.* pp. 4585-4592.

de Marneffe, M., and Manning, C.D., 2014. Stanford typed dependencies manual [EB/OL].

Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S. and Weischedel, R.M., 2004. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. *In: LREC,* pp. 1.

Dodds, L., and Davis, I., 2012. *Linked Data Patterns: A pattern catalogue for modelling, publishing, and consuming Linked Data.* http://patterns.dataincubator.org: .

Dombeu, J.V.F., and Huisman, M., 2011. Combining ontology development methodologies and semantic web platforms for e-government domain ontology development. *arXiv Preprint arXiv:1104.4966,* .

Du, J., and Zhou, L., 2012. Improving financial data quality using ontologies. *Decision Support Systems,* 54 (1), 76-86.

Fadyart, C., 2013. *Finance Ontology* [online]. Fadyart Co. Available at: http://www.fadyart.com/en/ [Accessed 12/15 2014].

Fafalios, P., and Tzitzikas, Y., 2013. X-ENS: semantic enrichment of web search results at real-time. *In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval,* ACM, pp. 1089-1090.

Färber, M., Menne, C. and Harth, A., 2017. A Linked Data wrapper for CrunchBase. *Semantic Web,* (Preprint), 1-11.

Faria, C., Girardi, R. and Novais, P., 2012. Using domain specific generated rules for automatic ontology population. *In: Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on,* IEEE, pp. 297-302.

Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C.D. and Stamatopoulos, P., 2000. Rule-based named entity recognition for Greek financial texts. *In: Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000),* Citeseer, pp. 75-78.

Feilmayr, C., 2011. Text mining-supported information extraction: an extended methodology for developing information extraction systems. *In: Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on,* IEEE, pp. 217-221.

Finkel, J.R., Grenager, T. and Manning, C., 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *In: Proceedings of the 43rd annual meeting on association for computational linguistics,* Association for Computational Linguistics, pp. 363-370.

FIRST, C., 2013. *Large scale information extraction and integration infrastructure for supporting financial decision making. Final Version of Integrated Financial Market Information System.* D7.1 ed. The European Commission within the Seventh Framework Programme (2007-2013).

Freebase Metaweb, 2014. *Freebase Dataset* [online]. Metaweb Technologies (Google). Available at: http://www.freebase.com [Accessed 9/1 2014].

Freeman, E.A., and Moisen, G.G., 2008. A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecological Modelling,* 217 (1), 48-58.

Fundel, K., Kuffner, R. and Zimmer, R., 2007. RelEx--relation extraction using dependency parse trees. *Bioinformatics (Oxford, England),* 23 (3), 365-371.

Gao, X., and Zhang, M., 2003. Learning knowledge bases for information extraction from multiple text based Web sites. *In: Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on,* IEEE, pp. 119-125.

Garcia, M., and Gamallo, P., 2011. A Weakly-Supervised Rule-Based Approach for Relation Extraction. *In: XIV Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2011),* pp. 07-2011.

García, S., Molina, D., Lozano, M. and Herrera, F., 2009. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics,* 15 (6), 617-644.

GATE, P.T., 2018. *General Architectures for Text Engineering* [online]. The University of Sheffield. Available at: https://gate.ac.uk/ [Accessed 02/15 2014].

Gokal, V., and Hanif, S., 2004. *Relationship between inflation and economic growth.* Economics Department, Reserve Bank of Fiji.

Goldberg, D.E., and Holland, J.H., 1988. Genetic algorithms and machine learning. *Machine Learning,* 3 (2), 95-99.

Goncalves, R., Josef, H., Horridge, M., Musen, M., Nyulas, C., Tu, S. and Tudorache, T., 2015. *Protégé project* [online]. National Institute of General Medical Sciences of the United States National Institutes of Health. Available at: http://protege.stanford.edu [Accessed 6/1 2010].

Grimm, S., Hitzler, P. and Abecker, A., 2007. *Knowledge representation and ontologies. Logic, Ontologies and SemanticWeb Languages.* Springer.

Grishman, R., 2012. Information extraction: Capabilities and challenges. *Lecture Notes].Retrieved from Http://Cs.Nyu.Edu/Grishman/Tarragona.Pdf,* .

Han, J., Kamber, M. and Pei, J., 2011. *Data mining: concepts and techniques: concepts and techniques.* Elsevier.

Harris, S., Seaborne, A. and Prud'hommeaux, E., 2013. SPARQL 1.1 query language. *W3C Recommendation,* 21.

Hasanuzzaman, M., Saha, S. and Ekbal, A., 2011. Feature Subset Selection Using Genetic Algorithm for Named Entity Recognition. *In: PACLIC 24, The 24th Pacific Asia Conference On Language, Information and Computation, Tohoku University, Sendai, Japan, November, 4-7 2010.* Japan: Institute of Digital Enhancement of Cognitive Processing, Waseda University, pp. 153-162.

Hattori, L., Guerrero, D., Figueiredo, J., Brunet, J. and Damásio, J., 2008. On the precision and accuracy of impact analysis techniques. *In: Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on,* IEEE, pp. 513-518.

Heath, T., and Bizer, C., 2011. *Linked data: Evolving the web into a global data space.* First Edition ed. USA: Morgan & Claypool Publishers.

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

Hegazy, A., Sakre, M. and Khater, E., 2015. Arabic Ontology Model for Financial Accounting. *Procedia Computer Science,* (62), 513-520.

Henson, C.A., 2013. *A Semantics-based Approach to Machine Perception.* Doctor of Philosophy., Wright State University.

Hitzler, P., Krotzsch, M. and Rudolph, S., 2009. *Knowledge representation for the semantic web.* Paderborn, Germany: KI.

Hmeidi, I., Hawashin, B. and El-Qawasmeh, E., 2008. Performance of KNN and SVM classifiers on full word Arabic articles. *Advanced Engineering Informatics,* 22 (1), 106-111.

Hoekstra, R., 2009. *Ontology Representation: Design Patterns and Ontologies that Make Sense.* ProQuest Ebook Central: Ios Press.

Hogan, A., Arenas, M., Mallea, A. and Polleres, A., 2014. Everything you always wanted to know about blank nodes. *Web Semantics: Science, Services and Agents on the World Wide Web,* 27, 42-69.

Hong, G., 2005. Relation extraction using support vector machine. *In:* Relation extraction using support vector machine. *Natural Language Processing–IJCNLP 2005.* Springer, 2005, pp. 366-377.

Horridge, M., Drummond, N., Goodwin, J., Rector, A.L., Stevens, R. and Wang, H., 2006. The Manchester OWL Syntax. *In: OWLed,* .

Horrocks, I., and Patel-Schneider, P.F., 2011. Knowledge Representation and Reasoning on the Semantic Web: OWL. *In:* J. Domingue, D. Fensel and J.A. Hendler, eds., *Handbook of Semantic Web Technologies .* Domingue, J.; Fensel, D.; Hendler, J. A.; ed. Springer, 2011, pp. 365-398.

Hsu, C., Chang, C. and Lin, C., 2003. *A Practical Guide to Support Vector Classification,* .

Hu, Y., De, S., Chen, Y. and Kambhampati, S., 2012. Bayesian data cleaning for Web data. *arXiv Preprint arXiv:1204.3677,* .

Huang, M., Zhu, X. and Li, M., 2006. A hybrid method for relation extraction from biomedical literature. *International Journal of Medical Informatics,* 75 (6), 443-455.

Hunjra, A.I., Chani, M.I., Javed, S., Naeem, S. and Ijaz, M.S., 2014. Impact of Micro Economic Variables on Firms Performance.

Imandoust, S.B., and Bolandraftar, M., 2013. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications,* 3 (5), 605-610.

Isiaq, S.O., and Osman, T., 2012. Multi-phase reasoning model for temporal semantic knowledgebase. *In: Proceedings of the International Workshop on Intelligent Exploration of Semantic Data (IESD),* .

Isiaq, S., and Osman, T., 2014. Ontology modelling methodology for temporal and interdependent applications. *In: Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on,* IEEE, pp. 434-439.

JENA Apache, 2015. *Apache JENA. a Java Framework for Building Semantic Web Applications* [online]. Apache Software Foundation. Available at: http://jena.apache.org/index.html [Accessed 02/01 2014].

Jiang, J., and Zhai, C., 2007. A Systematic Exploration of the Feature Space for Relation Extraction. *In: HLT-NAACL,* pp. 113-120.

Jiang, X., Huang, Y., Nickel, M. and Tresp, V., 2012. Combining information extraction, deductive reasoning and machine learning for relation prediction. *In: The Semantic Web: Research and Applications, ESWC 2012: Extended Semantic Web Conference. 9th Extended Semantic Web Conference, Heraklion, Crete, Greece, May 27-31, 2012.* Germany: Springer, pp. 164-178.

Jovic, A., Prcela, M. and Gamberger, D., 2007. Ontologies in medical knowledge representation. *In: Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on,* IEEE, pp. 535-540.

JSON, L., 1999. *JavaScript Object Notation. ECMA-404 The JSON Data Interchange Standard.* [online]. . Available at: http://www.json.org/ [Accessed 16/15 2015].

Kapoor, B., and Sharma, S., 2010. A comparative study ontology building tools for semantic web applications. *International Journal of Web & Semantic Technology (IJWesT),* 1 (3), 1-13.

Karegowda, A.G., Jayaram, M. and Manjunath, A., 2010. Feature subset selection problem using wrapper approach in supervised learning. *International Journal of Computer Applications,* 1 (7), 13-17.

Karkaletsis, V., Fragkou, P., Petasis, G. and Iosif, E., 2011. Ontology based information extraction from text. *In:* Ontology based information extraction from text. *Knowledge-driven multimedia information extraction and ontology evolution.* Springer, 2011, pp. 89-109.

Kazimipour, B., Li, X. and Qin, A., 2014. A review of population initialization techniques for evolutionary algorithms. *In: IEEE 2014, Congress on Evolutionary Computation (CEC), Beijing, China, 6-11 July 2014.* IEEE, pp. 2585-2592.

Khan, S., 2018. *Invest Excel* [online]. investexcel.net. Available at: http://investexcel.net/ [Accessed 07/15 2015].

Khondoker, M.R., and Mueller, P., 2010. Comparing ontology development tools based on an online survey. *In: Proceedings of the World Congress on Engineering,* pp. 2010.

Kim, M.K., and Burnie, D.A., 2002. The firm size effect and the economic cycle. *Journal of Financial Research,* 25 (1), 111-124.

Knublauch, H., Fergerson, R.W., Noy, N.F. and Musen, M.A., 2004. The Protégé OWL plugin: An open development environment for semantic web applications. *In: International Semantic Web Conference,* Springer, pp. 229-243.

Kohlschütter, C., Fankhauser, P. and Nejdl, W., 2010. Boilerplate detection using shallow text features. *In: Proceedings of the third ACM international conference on Web search and data mining,* ACM, pp. 441-450.

Kolovson, E., 2014. *What is the relationship between growth, inflation, and unemployment?* [online]. Quora. Available at: https://www.quora.com/What-is-the-relationship-between-growth-inflation-and-unemployment [Accessed 10/April 2017].

Konstantinova, N., 2014. Review of Relation Extraction Methods: What Is New Out There? *In: AIST 2014: The Third International Conference on Analysis of Images, Social Networks and Texts, Yekaterinburg, Russia, April 10-12, 2014.* Germany: Springer, pp. 15-28.

Kostylev, E.V., and Grau, B.C., 2014. On the semantics of SPARQL queries with optional matching under entailment regimes. *In: International Semantic Web Conference,* Springer, pp. 374-389.

Kovačević, A., Dehghan, A., Filannino, M., Keane, J.A. and Nenadic, G., 2013. Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives. *Journal of the American Medical Informatics Association,* 20 (5), 859-866.

Krieger, H., and Declerck, T., 2015. An OWL Ontology for Biographical Knowledge. Representing Time-Dependent Factual Knowledge. *In: BD,* pp. 101-110.

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

Kumar, B.S., and Ravi, V., 2016. A survey of the applications of text mining in financial domain. *Knowledge-Based Systems,* 114, 128-147.

Kumar, V., and Minz, S., 2014. Feature Selection: A literature Review. *Smart Computing Review,* 4 (3), 211-229.

Kumari, B., and Swarnkar, T., 2011. Filter versus wrapper feature subset selection in large dimensionality micro array: A review. *International Journal of Computer Science and Information Technologies,* 2 (3), 1048-1053.

Lantzaki, C., Yannakis, T., Tzitzikas, Y. and Analyti, A., 2014. Generating synthetic RDF data with connected blank nodes for benchmarking. *In: European Semantic Web Conference,* Springer, pp. 192-207.

Lawrynowicz, A., and Tresp, V., 2014. Introducing Machine Learning. *In:* J. Lehmann, and J. Völker, eds., *Perspectives on Ontology Learning.* Germany: IOS Press, 2014, pp. 35-50.

Levine, L., 2012. Economic growth and the unemployment rate.

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

Lewis, D.D., 1995. Evaluating and optimizing autonomous text classification systems. *In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval,* ACM, pp. 246-254.

Li, Q., Wang, T., Li, P., Liu, L., Gong, Q. and Chen, Y., 2014a. The effect of news and public mood on stock movements. *Information Sciences,* 278, 826-840.

Li, X., Xie, H., Chen, L., Wang, J. and Deng, X., 2014b. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems,* 69, 14-23.

Li, Y., Bontcheva, K. and Cunningham, H., 2009. Adapting SVM for data sparseness and imbalance: a case study in information extraction. *Natural Language Engineering,* 15 (02), 241-271.

Li, Y., Bontcheva, K. and Cunningham, H., 2005. SVM based learning system for information extraction. *In: First International Workshop in Deterministic and statistical methods in machine learning, Sheffield, UK, September 7-10, 2004.* UK: Springer, pp. 319-339.

Li, Y., Miao, C., Bontcheva, K. and Cunningham, H., 2005. Perceptron Learning for Chinese Word Segmentation. *In: Proceedings of Fourth SIGHAN Workshop on Chinese Language processing (Sighan-05),* pp. 154-157.

Li, Y., and Shawe-Taylor, J., 2003. The SVM with uneven margins and Chinese document categorization. *In: Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17),* pp. 216-227.

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J. and Kandola, J., 2002. The perceptron algorithm with uneven margins. *In: ICML 2002, The Nineteenth International Conference on Machine Learning, The University of New South Wales, Sydney, Australia, 8-12 July 2002.* pp. 379-386.

Liu, B., 2011. Supervised learning. *In:* Supervised learning. *Web Data Mining.* Springer, 2011, pp. 63-132.

Liu, L., Ren, X., Zhu, Q., Zhi, S., Gui, H., Ji, H. and Han, J., 2017. Heterogeneous Supervision for Relation Extraction: A Representation Learning Approach. *arXiv Preprint arXiv:1707.00166,* .

Lloret, E., Gutiérrez, Y. and Gómez, J.M., 2015. Developing an Ontology to Capture Documents' Semantics. *In: KEOD,* pp. 155-162.

LOD, c., 2018. *Linked Open Data Clould* [online]. Linked Data community. Available at: http://linkeddata.org/ [Accessed 03/15 2014].

Lord, P., 2010. ***Components of an Ontology*** [online]. . Available at: http://ontogenesis.knowledgeblog.org/514 [Accessed 6/29 2015].

Lorena, A.C., and De Carvalho, A.C., 2008. Evolutionary tuning of SVM parameter values in multiclass problems. *Neurocomputing,* 71 (16), 3326-3334.

Lozano, M., Herrera, F. and Cano, J.R., 2005. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *In:* F. Hoffmann, M. Köppen, F. Klawonn and R. Roy, eds., *Soft Computing: Methodologies and Applications.* Netherlands: Springer Science & Business Media, 2005, pp. 85-96.

Lupiani-Ruiz, E., GarcíA-Manotas, I., Valencia-GarcíA, R., GarcíA-SáNchez, F., Castellanos-Nieves, D., FernáNdez-Breis, J.T. and CamóN-Herrero, J.B., 2011. Financial news semantic search engine. *Expert Systems with Applications,* 38 (12), 15565-15572.

MacFarlane, A., Secker, A., May, P. and Timmis, J., 2010. An experimental comparison of a genetic algorithm and a hill-climber for term selection. *Journal of Documentation,* 66 (4), 513-531.

Mallea, A., Arenas, M., Hogan, A. and Polleres, A., 2011. On blank nodes. *In: International Semantic Web Conference,* Springer, pp. 421-437.

Marie, N., and Gandon, F., 2014. Survey of linked data based exploration systems. *In: Proceedings of the 3rd International Conference on Intelligent Exploration of Semantic Data-Volume 1279,* CEUR-WS. org, pp. 66-77.

Maynard, D., Li, Y. and Peters, W., 2008. Nlp techniques for term extraction and ontology population. *In: Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge,* pp. 107-127.

Mendes, P.N., Jakob, M. and Bizer, C., 2012. DBpedia: A Multilingual Cross-domain Knowledge Base. *In: LREC,* pp. 1813-1817.

Mian, G.M., and Sankaraguruswamy, S., 2012. Investor sentiment and stock market response to earnings news. *The Accounting Review,* 87 (4), 1357-1384.

Mills, K.L., Filliben, J.J. and Haines, A., 2015. Determining relative importance and effective settings for genetic algorithm control parameters. *Evolutionary Computation,* 23 (2), 309-342.

Minard, A.L., Ligozat, A.L., Ben Abacha, A., Bernhard, D., Cartoni, B., Deleger, L., Grau, B., Rosset, S., Zweigenbaum, P. and Grouin, C., 2011. Hybrid methods for improving information access in clinical documents: concept, assertion, and relation identification. *Journal of the American Medical Informatics Association : JAMIA,* 18 (5), 588-593.

Minkov, E., Wang, R.C., Tomasic, A. and Cohen, W.W., 2006. NER systems that suit user's preferences: adjusting the recall-precision trade-off for entity Extraction. *In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers,* Association for Computational Linguistics, pp. 93-96.

Mintz, M., Bills, S., Snow, R. and Jurafsky, D., 2009. Distant supervision for relation extraction without labeled data. *In: ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume(2), Suntec, Singapore, August 02 - 07, 2009.* Association for Computational Linguistics, pp. 1003-1011.

Mitchell, A., Strassel, S., Przybocki, M., Davis, J., Doddington, G., Grishman, R., Meyers, A., Brunstein, A., Ferro, L. and Sundheim, B., 2003. *ACE-2 Version 1.0 LDC2003T11. Web Download*
[online]. Linguistic Data Consortium. Philadelphia, USA. Available at: https://catalog.ldc.upenn.edu/docs/LDC2003T11/ [Accessed 05/25 2015].

Mitchell, M., Holland, J. and Stephanie, F., 1994. When will a genetic algorithm outperform hill-climbing? *In:* J.D. Cowan, G. Tesauro and J. Alspector, eds., *Advances in Neural Information Processing Systems. Volume(6), In Proceedings of the annual Conferences on Advances in Neural Information Processing Systems 1993.* Morgan Kaufmann, 1994, pp. 51-85.

Mohamed, R., El-Makky, N.M. and Nagi, K., 2015. ArabRelat: Arabic Relation Extraction using Distant Supervision. *In: The 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Lisbon-Portugal, November 12-14, 2015.* SCITEPRESS–Science and Technology Publications, pp. 410-417.

Nagypál, G., 2005. *WP3: Service Ontologies and Service Description, D3.11. Methodology for building SWS ontologies in DIPFP6–507483.* D3.11 ed. Germany: Data, Information and Process (DIP) Integration with Semantic Web Services, FP6–507483.

Nagypál, G., Deswarte, R. and Oosthoek, J., 2005. Applying the semantic web: The VICODI experience in creating visual contextualization for history. *Literary and Linguistic Computing,* 20 (3), 327-349.

Nofsinger, J.R., 2001. The impact of public information on investors. *Journal of Banking & Finance,* 25 (7), 1339-1366.

Nováček, V., Laera, L., Handschuh, S. and Davis, B., 2008. Infrastructure for dynamic knowledge integration—Automated biomedical ontology extension using textual resources. *Journal of Biomedical Informatics,* 41 (5), 816-828.

Noy, N.F., and McGuinness, D.L., 2001. Ontology development 101: A guide to creating your first ontology.

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

ODP, o., 2018. *Ontology Design Patterns* [online]. NeOn projec. Available at: http://ontologydesignpatterns.org/wiki/Main_Page [Accessed 12/10 2016].

Osman, T., Lotfi, A., Langensiepen, C., Chernbumroong, S. and Saeed, M., 2014. Semantic-based decision support for remote care of Dementia patients. *In: Intelligent Agents (IA), 2014 IEEE Symposium on,* IEEE, pp. 89-96.

Panchenko, A., Adeykin, S., Romanov, P. and Romanov, A., 2012. Extraction of semantic relations between concepts with knn algorithms on wikipedia. *In: Concept Discovery in Unstructured Data Workshop (CDUD) of International Conference On Formal Concept Analysis, Belgium,* Citeseer, pp. 78-88.

Perera, S., Henson, C., Thirunarayan, K., Sheth, A. and Nair, S., 2012. Data driven knowledge acquisition method for domain knowledge enrichment in the healthcare. *In: Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on,* IEEE, pp. 1-8.

Petrillo, M., and Baycroft, J., 2010. Introduction to Manual Annotation.

Piskorski, J., and Yangarber, R., 2013. Information Extraction: Past, Present and Future. *In:* T. Poibeau, H. Saggion, J. Piskorski and R. Yangarber, eds., *Multi-source, Multilingual Information Extraction and Summarization.* Berlin, Heidelberg: Springer-Verlag, 2013, pp. 23-49.

Pohja, M., 2011. *Web application user interface technologies.* Doctor of Science., Aalto University.

Polleres, A., Hogan, A., Delbru, R. and Umbrich, J., 2013. RDFS and OWL reasoning for linked data. *In:* RDFS and OWL reasoning for linked data. *Reasoning Web. Semantic Technologies for Intelligent Data Access.* Springer, 2013, pp. 91-149.

Pollin, R., and Zhu, A., 2006. Inflation and economic growth: A cross-country nonlinear analysis. *Journal of Post Keynesian Economics,* 28 (4), 593-614.

Protege, S., 2018. *Developing and Maintaining Ontologies* [online]. Stanford Center for Biomedical Informatics Research (BMIR). Available at: https://protege.stanford.edu [Accessed 02/02 2014].

Prud, E., and Seaborne, A., 2006. SPARQL query language for RDF.

Pundir, P., Gomanse, V. and Krishnamacharya, N., 2013. Classification and Prediction techniques using Machine Learning for Anomaly Detection. *International Journal of Engineering Research and Applications (IJERA),* .

Pundir, P., Gomanse, V. and Krishnamacharya, N., 2011. Classification and Prediction techniques using Machine Learning for Anomaly Detection. *International Journal of Engineering Research and Applications (IJERA),* 1 (4), 1716-1722.

R, P., 2018. *The R Project for Statistical Computing* [online]. R Project. Available at: https://www.r-project.org/ [Accessed 11/15 2016].

Rattanasawad, T., Buranarach, M., Thein, Y.M., Supnithi, T. and Saikaew, K.R., 2014. Design and Implementation of a Rule-based Recommender Application Framework for the Semantic Web Data. *In: 2013 Linked Data in Practice Workshop (LDPW2013),* pp. 54.

Reilly, F.K., and Brown, K.C., 2011. *Investment analysis and portfolio management.* 10th ed. USA: Cengage Learning.

Rekha, R., and Syamili, C., 2017. Ontology Engineering Methodologies: An Analytical Study. *In: 11th International CALIBER2017, Gujarat, India, 2-4, August, 2017.* INFLIBNET Centre, pp. 193-199.

Remya, K.R., and Rama, J.S., 2014. A Survey of Machine Learning Approaches for Relation Classification from Biomedical Texts . *International Journal of Emerging Technology and Advanced Engineering,* 4 (3), 143-148.

Reuters, C., 2018. *Reuters Stock Market news RSS Link* [online]. Reuters Co. Available at: http://feeds.reuters.com/reuters/UKPersonalFinanceNews [Accessed 03/02 2014].

Rifkin, R., and Klautau, A., 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research,* 5 (Jan), 101-141.

Rinne, M., 2012. SPARQL update for complex event processing. *The Semantic Web–ISWC 2012,* , 453-456.

Rizzo, G., and Troncy, R., 2012. NERD: a framework for unifying named entity recognition and disambiguation extraction tools. *In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics,* Association for Computational Linguistics, pp. 73-76.

Rodríguez, N.D., Cuéllar, M.P., Lilius, J. and Calvo-Flores, M.D., 2014. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems,* 66, 46-60.

Rospocher, M., and Serafini, L., 2012. An ontological framework for decision support. *In: Joint International Semantic Technology Conference,* Springer, pp. 239-254.

Roussey, C., Pinet, F., Kang, M.A. and Corcho, O., 2011. An introduction to ontologies and ontology engineering. *In:* An introduction to ontologies and ontology engineering. *Ontologies in Urban Development Projects.* Springer, 2011, pp. 9-38.

Ruiz-Martínez, J.M., Valencia-García, R. and García-Sánchez, F., 2012. Semantic-Based Sentiment analysis in financial news. *In: Proceedings of the 1st International Workshop on Finance and Economics on the Semantic Web,* pp. 38-51.

Saidi, I., Amer-Yahia, S. and Bahloul, S.N., 2014. An Approach to Diversify Entity Search Results. *In: ICAASE,* pp. 44-51.

Saif, H., Fernandez, M., He, Y. and Alani, H., 2014. SentiCircles for Contextual and Conceptual Semantic Sentiment Analysis of Twitter. *In:* SentiCircles for Contextual and Conceptual Semantic Sentiment Analysis of Twitter. *The Semantic Web: Trends and Challenges.* Springer, 2014, pp. 83-98.

Sakamoto, S., Kulla, E., Oda, T., Ikeda, M., Barolli, L. and Xhafa, F., 2014. A comparison study of hill climbing, simulated annealing and genetic algorithm for node placement problem in WMNs. *Journal of High Speed Networks,* 20 (1), 55-66.

Salguero, A., Delgado, C. and Araque, F., 2009. Easing the Definition of N–Ary Relations for Supporting Spatio–Temporal Models in OWL. *Computer Aided Systems Theory-EUROCAST 2009,* , 271-278.

Sastry, K., Goldberg, D.E. and Kendall, G., 2014. Genetic Algorithms. *In:* E.K. Burke, and G. Kendall, eds., *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques.* Second Edition ed. New York, USA: Springer Science and Business Media, 2014, pp. 93-117.

Schumaker, R.P., and Chen, H., 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS),* 27 (2), 12.

Segaran, T., Evans, C. and Taylor, J., 2009. *Programming the semantic web.* " O'Reilly Media, Inc.".

Shalev-Shwartz, S., and Ben-David, S., 2014. *Understanding machine learning: From theory to algorithms.* Cambridge University Press.

Simeonov, B., Alexiev, V., Liparas, D., Puigbo, M., Vrochidis, S., Jamin, E. and Kompatsiaris, I., 2016. Semantic integration of web data for international investment decision support. *In: International Conference on Internet Science,* Springer, pp. 205-217.

Sinha, A., and Couderc, P., 2012. Using owl ontologies for selective waste sorting and recycling. *In: OWLED-2012,* .

Slimani, T., 2015. Ontology development: A comparing study on tools, languages and formalisms. *Indian Journal of Science and Technology,* 8 (24).

Song, Y., and Roth, D., 2017. Machine Learning with World Knowledge: The Position and Survey. *CoRR,* arXiv:1705.02908v1 [cs.AI].

Songsangyos, P., and Iamamporn, S., 2014. DECISION SUPPORT SYSTEM FOR PERSONAL FINANCIAL ANALYSIS. *In: The International Academic Conference Proceedings , Bali, Indonesia, 2014.* The West East Institute, pp. 90-94.

Spaulding, W., C., 2017. *Dividend Discount Model (DDM), Money Tutorials* [online]. thismatter.com. Available at: http://thismatter.com/money/stocks/valuation/dividend-discount-model.htm [Accessed March/15 2017].

Szeredi, P., Lukácsy, G. and Benkő, T., 2014. *The Semantic Web Explained, The Technology and Mathematics behind Web 3.0.* 2nd ed. Cambridge, UK: Cambridge University Press.

Tan, F., 2007. *Improving feature selection techniques for machine learning.* Doctor of Philosophy (PhD)., Georgia State University.

Tang, N., 2015. Big RDF data cleaning. *In: Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on,* IEEE, pp. 77-79.

Taye, M.M., 2010. Understanding semantic web and ontologies: Theory and applications. *arXiv Preprint arXiv:1006.4567,* .

Thakker, D., Osman, T. and Lakin, P., 2009. Gate jape grammar tutorial. *Nottingham Trent University, UK, Phil Lakin, UK, Version,* 1.

Thakker, D., Yang-Turner, F. and Despotakis, D., 2016. User interaction with linked data: An exploratory search approach.

Thakker, D., Dimitrova, V., Cohn, A.G. and Valdes, J., 2015. PADTUN-using semantic technologies in tunnel diagnosis and maintenance domain. *In: European Semantic Web Conference, Portoroz, Slovenia, 31 May - 04 Jun 2015.* Springer, pp. 683-698.

Toledo, C.M., Chiotti, O. and Galli, M.R., 2012. An ontology evolution approach for information retrieval strategies with compound terms. *In: Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En,* IEEE, pp. 1-10.

Tomai, E., and Spanaki, M., 2005. From ontology design to ontology implementation: A web tool for building geographic ontologies.

Tzitzikas, Y., Lantzaki, C. and Zeginis, D., 2012. Blank node matching and RDF/S comparison functions. *In: International Semantic Web Conference,* Springer, pp. 591-607.

Valstar, M.F., Mehu, M., Jiang, B., Pantic, M. and Scherer, K., 2012. Meta-analysis of the first facial expression recognition challenge. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* 42 (4), 966-979.

Van Heijst, G., Schreiber, A.T. and Wielinga, B.J., 1997. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies,* 46 (2-3), 183-292.

Vinu, P., Sherimon, P., KRISHNAN, R. and SAAD TAKRONI, Y., 2014. PATTERN REPRESENTATION MODEL FOR N-ARY RELATIONS IN ONTOLOGY. *Journal of Theoretical & Applied Information Technology,* 60 (2).

W3C, S., 2018. *The World Wide Web Consortium (W3C)* [online]. The World Wide Web Consortium. Available at: https://www.w3.org/standards/semanticweb/ [Accessed 02/01 2014].

Wang, T., Li, Y., Bontcheva, K., Cunningham, H. and Wang, J., 2006. Automatic extraction of hierarchical relations from text. *In: ESWC 2006, 3rd European Semantic Web Conference, Budva, Montenegro, June 11-14, 2006.* Springer, pp. 215-229.

Wang, X.H., Zhang, D.Q., Gu, T. and Pung, H.K., 2004. Ontology based context modeling and reasoning using OWL. *In: Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on,* Ieee, pp. 18-22.

Wanner, L., Rospocher, M., Vrochidis, S., Johansson, L., Bouayad-Agha, N., Casamayor, G., Karppinen, A., Kompatsiaris, I., Mille, S. and Moumtzidou, A., 2015. Ontology-centered environmental information delivery for personalized decision support. *Expert Systems with Applications,* 42 (12), 5032-5046.

Weikum, G., and Theobald, M., 2010. From information to knowledge: harvesting entities and relationships from web sources. *In: Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems,* ACM, pp. 65-76.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V., 2001. Feature selection for SVMs. *In: Advances in neural information processing systems,* pp. 668-674.

Witten, I.H., and Frank, E., 2005. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

WorldBankData, A., 2018. *Developer Information for Accessing the World Bank API* [online]. The World Bank Group. Available at: https://datahelpdesk.worldbank.org/knowledgebase/topics/125589-developer-information [Accessed 11/10 2015].

Wu, D.D., Zheng, L. and Olson, D.L., 2014. A decision support approach for online stock forum sentiment analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 44 (8), 1077-1087.

Xue, B., Zhang, M. and Browne, W., N., 2015. A comprehensive comparison on evolutionary feature selection approaches to classification. *International Journal of Computational Intelligence and Applications,* 14 (2), 22-49.

Yahoo, C., 2018. *Yahoo Finance Stock Market news RSS Link* [online]. Yahoo Co. Available at: https://uk.finance.yahoo.com/news/provider-yahoofinance [Accessed 03/02 2014].

YahooFinance, A., 2018. *Yahoo Finance API to access Stock Market data* [online]. Yahoo Co. Available at: http://finance.yahoo.com/lookup?s=API [Accessed 02/10 2014].

Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z. and Ishizuka, M., 2009. Unsupervised relation extraction by mining Wikipedia texts using information from the web. *In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2,* Association for Computational Linguistics, pp. 1021-1029.

Yang-Turner, F., Lau, L., Dimitrova, V. and Thakker, D., 2013. Profiling Exploratory Browsing Behaviour with a Semantic Data Browser.

Ye, J., Dasiopoulou, S., Stevenson, G., Meditskos, G., Kontopoulos, E., Kompatsiaris, I. and Dobson, S., 2015. Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing,* 23, 1-25.

Yelwa, M., David, O.O. and Awe, E.O., 2015. Analysis of the Relationship between Inflation, Unemployment and Economic Growth in Nigeria: 1987-2012. *Applied Economics and Finance,* 2 (3), 102-109.

Yong, C.C., and Taib, S.M., 2009. Designing a decision support system model for stock investment strategy. *In: Proceedings of the World Congress on Engineering and Computer Science 2009 I,* .

Yong, Z., and Sannomiya, N., 2001. An Improvement of Genetic Algorithms by Search Space Reductions in Solving Large-Scale Flowshop Problems. *The Transactions of the Institute of Electrical Engineers of Japan.C, A Publication of Electronics, Information and System Society,* 121 (6), 1010-1015.

Yoo, D., and No, S., 2014. Ontology-based economics knowledge sharing system. *Expert Systems with Applications,* 41 (4), 1331-1341.