Semantic Agent Based Knowledge Discovery from Big Data on the Web

Raniah Bafaqeer, Fathy Eassa and Kamal Jambi

Computer Science Department, Faculty of Computing and Information Technology, King AbdulAziz University, Jeddah, Saudi Arabia
rbafaqeer@kau.edu.sa

Abstract. The emergence of Big Data has begun from the fast-rapid growth of the Web resource on the internet. The size of the Web becomes too massive, heterogeneous, and updated day by day, impacted by the networking's fast development, data collection capacity and data storage. Discovering unknown knowledge is a big challenge due to the lack of relationships between data sets, making traditional web mining results almost unsatisfactory when the user have insufficient data parameters to search. In this paper, we have suggested a unified semantic agent program to improve the knowledge discovery process from Big Data on the web using ontology. The model extracts information from heterogeneous data sources, then defines a unified structure to store them that make it easily searchable, maximizing discovery and reporting for users. The evaluation of the approach shows higher precision and recall, using ontology technique instead of a keyword based in the mining process.

Keywords: Big Data, Semantic Web, Knowledge Discovery, Ontology, Semantic Scrapper, RDF.

1. Introduction

The high spectrum of data being generated and collected today is growing exponentially which will make it approach the Zettabyte/year range within a few years^[1]. Telecommunications (Mobile phones, GPS), online activities (social networks, websites, electronic communication, online shopping), scientific activities (simulations, experiments, environmental sensors) all generate streams of data as a by-product of their usual operations. Besides the fact that organizations are generating and storing more data than ever before such website access logs are no longer being deleted – they are being archived to discover valuable insights.

As clear, many of the most important sources of Big Data are relatively new and separate. A study stated that “A full 90% of all

the data in the world has been generated over the last two years”^[2]. By 2012, about 2.5 Exabytes of data are made every day, and every 40 months or thereabouts, that number is multiplying at constant intervals. According to^[3], “Every second, more data cross the internet than were stored in the entire internet just 20 years ago”.

The overflow of data is becoming a reality. About 500 million tweets daily or several terabytes of data are generated by Twitter alone. CISCO estimates that the world will produce 1.3 Zettabyte of information annually by 2016. According to^[4], “Annual global IP traffic will surpass the zettabyte threshold in 2016, and will reach 2.3 zettabyte by 2020”. That means, throughout the following 5 years it will increment about triple, and will have expanded almost a hundredfold from 2005 to 2020. The growth of

data is estimated by the Internet Data Center (IDC) to be of a factor of 300 between 2005 and 2020, and is expected to raise from 130 Exabytes to 20,000 Exabytes^[5].

To describe the Big Data, many definitions of Big Data can be found. According to^[6], Big Data is defined as “the representation of the progress of human cognitive processes, which generally includes data sets with sizes beyond the capacity of current technology, methods and theories to capture, manage and process the data within a specified time”. Additionally, Gartner^[7] defines Big Data as the high volume, high speed and/or high variety of information that require new ways of processing to allow better decision making, new knowledge discovery and process optimization.

This large variety of definitions in fact gives the impression that Big Data is an indistinct term, likely in light of the fact that Big Data is not tangible nor a fixed property. It is harder to detect, define and recognize Big Data analytics from traditional data analytics, because Big Data itself is not a disruptive advancement but rather a sustaining innovation. What is called “Big Data” now is not the result of a specific invention of a product or service but rather some revolutionary and evolutionary changes in different fields.

A large amount of information resources distributed on the web of high value-added to organizations can be presented due to the massive increase in the production and dissemination of data on the Internet. Impacted by the fast development of networking, data storage and the data collection capacity, the size of the web becomes too massive, heterogeneous and updated day by day. Therefore, knowledge discovery from Big Data enhance the organization's perspectives, allow them to have deeper insights and look at

the bigger picture in their business. Big Data analytics can be seen as both the successor and enhancer of traditional data analytics .

New data warehousing and database technologies have been introduced to address the problem of data becoming at extreme scale and continuing to do so. Cloud computing plays a significant role as it gives organizations the ability to analyze data often in a more economical way as it presents computing resources on demand. The outcome of the large volume of unstructured data is the development of new NoSQL databases as it is capable of storing data having an unknown or dynamic structure. The same applies to Hadoop^[8], often identified as a synonym for Big Data due to its capabilities to store and handle huge amounts of data in an economical way by effectively making use of parallel and distributed computing. These previous technologies have been an enormous help in dealing with the first two dimensions of Big Data – volume and velocity. But the large variety of data still remains an issue that is much more difficult to resolve programmatically and attracted more focus from research community over the last years^[7,9].

How to effectively mine the data and put the data into the right context by making sense and adding context to the diverse data types and sources is a large part of the challenge in variety dimension. Domain expertise is needed to make sense of the context, to understand the context of the data, and Big Data skills, to understand how to use the data, which cost higher. Thus, knowing the exact semantics of the data is required for eliminating the inconsistency of data and terms as well as finding and meaningfully combining information in the vast majority of data^[10]. This is where the "Semantic Web technology" steps in, so it can provide the opportunity for reducing cost and complexity of data integration. Semantic web technologies are a

good fit for the variety of Big Data and are more powerful ways to find and explore information.

The semantic web-based approach can be sufficiently used to improve the quality of knowledge discovery and the heterogeneity boundary break down, makes machines or software agents capable of seamlessly accessing the cross-organizational resource sharing and increase the level of automation which reduce the need of manual intervention. Many different roles for knowledge management can be played by the core Semantic Web technologies such as ontology and semantic reasoning ^[11]. The Semantic Web technology's core is ontology, which is also an important method to conceptualize domain knowledge and modeling that can be used to describe the semantics of the data. Domain ontology using Web Ontology Language (OWL), Resource Description Framework (RDF) and Resource Description Framework Schema (RDFS) ^[12,13]. While RDFS is used to represent the conceptual hierarchy of classes and relationships, RDF represents the information as a triple (a subject, a predicate, and an object).

The difficulties in processing the heterogeneity of Big Data is addressed by the existing researchers as well as leveraging them for further analysis and query response. The major challenge includes feature extraction from the structured, semi-structured and unstructured data, as well as deciding which mining techniques can and should be used in which contexts. Furthermore, enabling interoperability across heterogeneous semantic web applications and extracting an efficient result by understanding the users' contextual intent.

The key to solving these problems is in establishing a unified data model that either develops new algorithms that cope with heterogeneous data between distributed sites

or new models for preprocessing data to make them structured to fit existing algorithms .

To our knowledge, there is no unified semantic agent ontology-based cloud computing system that consider all the Big Data characteristics, not only variety, but also volume and velocity for discovering knowledge dependent on user's requirement.

Our idea is to create a unified model that aims to leverage the semantics available in ontologies in the Knowledge Discovery from Data (KDD) process. The model's purpose is to improve the knowledge discovery process from big data on the web, leveraging ontology to address ambiguity in the textual sources and to give the user more efficient and effective results by comprehending the users' intent and the contextual meaning of terms as they show in the searchable data space. The suggested model should take into consideration all the Big Data characteristics, not only variety, but also volume and velocity which were confronted when processing huge heterogeneous data amount and a quickness of data processing, data loading, and data storing.

The following sections of the paper are organized as follows. Section II presents the related work of the study, while section III, introduces our proposed model from the architectural design and implementation perspectives. Section IV shows the experiments process, with methods of training and test set construction. discusses the tools and techniques used to implement and test the proposed system. Section V shows the results and the discussion, while section VI illustrates the conclusion and future work.

2. Related Work

2.1 Knowledge Discovery from Big Data on the Web

Over the last years, Big Data's knowledge discovery has grown vastly,

attracted much more focus, and strategically became the next big trend in analytics for large business enterprises, government organizations, and research institutions. However, effectively producing knowledge from massive Big Data that were distributed and often had heterogeneous architecture, using data sources and data mining algorithms spread over thousands of nodes is as challenging as it is a complex task. Formalizing and organizing the process of Big Data analysis based on a controlled and measurable factor will become essential.

E. Begoli and J. Horey in ^[14], suggested three design principles that can guide organizations to develop an effective, useful and flexible knowledge discovery architectures over Big Data. These principles indicate that Big Data is reshaping the way knowledge discovery is, and how it will be done. Nevertheless, this paradigm shift brings new challenges, more opportunities that surpass traditional relational databases to rely on less structured data.

The issue of heterogeneity in Big Data which is the result of unstructured data and how we can ensure “veracity” in such data has revolted great concern to researchers all over the world. Since the ever-growing data come in multiple formats, the challenge that we face is how to effectively mine the data. This challenge is prevalent because existing data mining techniques have been designed for schema-oriented and structured data storages, and they are no longer capable of solving the problem of Big Data. Traditional compute and storage systems reach their limits, as they are incapable of handling the volumes, velocities or varieties brought by Big Data.

In Ref. ^[15], authors presented a HACE theorem that characterizes the features of the Big Data revolution, discusses the critical and most challenging issues for discovering useful

knowledge from the Big Data that lie in three tiers : data, model and systems levels. The challenges include not just the obvious issues of scale, but also heterogeneity, lack of structure, error-handling, privacy, timeliness, provenance, and visualization, at all stages of the analysis pipeline, from data acquisition to result interpretation. These technical challenges are common among a large variety of application domains, and therefore will demand transformative solutions from basic science to industrial application, and will not be addressed naturally by the next generation of industrial products.

Research in the area of the semantic web has created a quite grown standard for modeling and regulating the knowledge of the domain. Today, semantic web ontologies became a key technology for intelligent knowledge processing, providing a framework for sharing conceptual models about a domain ^[16].

The introduction of ontology-based data mining to integrate thinking, intelligent discovery of Big Data, can offer a new idea for Big Data research and application and seems to be one of most promising approaches. At present, the research by combining the ontology with the Big Data application is still rare and in its early stage ^[17, 18, 19]. Many research efforts have pointed out that ontology can play an important role in all stages of data mining including data transformation, feature reduction, algorithm selection, post-processing, model interpretation and so forth. In Ref. ^[16], authors investigated the potential of ontology to bring rich semantics into all stages of the data mining process which called it a semantic data mining. It summarized three main advantages that ontologies have brought to semantic data mining: First, bridge semantic gaps between the data, applications, data mining algorithms, and data mining results in all stages of data

mining including preprocessing, algorithm execution, and result generation. Second, provide the data mining algorithm with prior knowledge which either acts as a guide to the mining process or reduces the search space. Third, provide a formal way for representing the data mining flow, from data preprocessing to mining results.

In Ref. ^[20], L. Kang et al proposed a new Big Data model based on ontology that accommodate with the MapReduce framework for solving the issue of structural and semantic heterogeneity of multiple data sources known as Key/Value data model. The Key/Value data model is a map in which each specific thing in the ontology that is defined by URI will be stored as a Row-key, while properties and individuals will be stored as values as a collection of columns. Therefore, the data are consistent with RDF semantic and managed by HBase NoSQL databases based on cloud computing. The model has a simple and flexible scalability based on HBase implementation and verification, which is easy for updating data dynamically and meeting the needs of high concurrent data processing in a Big Data environment.

Furthermore, authors in ^[21] introduced a framework for a semantic tool that extracts relevant information from unstructured data and presents to the user a dynamic and interactive graph that allows visual exploration of the domain entities and the relationships between them based on the input search string. This research use semantic and NLP technologies to build mechanism for semantically processing domain specific data and provides a meaningful and relevant results from a well-structured ontology file that is stored in the RDF files as collection of triplets. Then visualize the extracted structured data through a graph visualization technique called spring graph. Spring graph is knowledge representation that shows the underlying

relationships among different entities and visualize them using interactive and dynamic graph. This system uses different open source tools. It is mostly dependent on available GATE tool's plug-in and Stanford parser. The suggested implementation of the spring graph component forms a robust foundation for information retrieval and visualization.

In Ref. ^[22], the authors presented a framework to automate multi-layered classification algorithms to facilitate connecting, capturing, and accessing relevant information from large datasets of sensory data as an application of ontology-based classifiers. They developed a new classifier to classify data from various domains of knowledge semantically, as shown in fig.3. The purpose of this research was to simplify integrating different heterogeneous sources of knowledge into a single one, by using ontological patterns.

Abirami and Askarunisa ^[23] developed a model for converting the unstructured or semi-structured text in an organized and structured manner into XML/RDF format to achieve easy machine processing and discover new knowledge. It uses RDF to store the content that was extracted from HTML pages, SPARQL and Jena for querying RDF content, Ontologies for context based search combined with machine learning techniques (in particular, wrapper induction based components) and cosine similarity to measure the similarity between extracted data. The model includes three main steps: pre-processing phase to make the document syntactically clean by HTMLTidy parser, structured data extraction phase which uses wrapper induction techniques to induce extraction rules on web pages which are semantically analyzed and RDF generation phase. This approach leads to quick inferences, increased relevancy and new knowledge discovery from the set of data in minimal time compared to manual efforts when semantic web technologies are used alone to develop the system.

2.2 Semantic Agent Technology and Knowledge Discovery

In Ref. ^[24], Scheutz and Mayer believed that combining Big Data methods with agent-based modeling is very important and worthwhile. It showed invaluable results to support urban designers and architects. It discusses that combining Big Data methods with those of agent-based simulations to support architectural and urban designs has a great potential.

Many authors have worked on techniques for extracting information from the web. In Ref. ^[25], the authors implement an architecture of Information Retrieval, that uses a semantic extraction agent for the Web environment, allowing information finding, storage, processing and retrieval from the Big Data context using the domain of higher education institutions. In this context, the scrapper semantic search participates as a tooling strategy for searching and finding the information that adds actual value to the decision-making process.

In Ref. ^[26], the authors developed an agent based knowledge management application framework. They used a certain type of ontology to facilitate semantic web service search and automatic composition. The framework is a general development environment for both business and GIS agent based. It used many technologies which are: Semantic web and ontology, multi agent systems and semantic web services. The authors used the proposed framework in developing agent based supply chain application in order to validate the framework. The supply chain application is implemented in a distributed manner that allows agents running on different platforms to communicate on the internet. The communication between agents is dependent on OWL ACL messages compliant with FIPA standards. However, the authors didn't discuss

the accuracy of their results and compare it with other works.

As a result, new ideas, an effective approaches and technologies are needed alongside powerful computer platforms to support traditional data analytics, to achieve fast response and real-time discovery and unprecedented advances in Big Data application ^[27]. Adrian & Chamberlin ^[28] mentioned that: "Big data can enable richer reporting and analysis, complementing, rather than replacing, traditional data warehouses".

3. System Architecture and Design

The proposed model called the Semantic Agent Based Knowledge Discovery from Big Data on the Web. Figure 1 shows our model's high-level architecture that consists of two main modules: the knowledge Discovering and Retrieving Module and Semantic Query Analysis module.

The knowledge Discovering and Retrieving Module receives the request through Web application & APIs layer to crawl the external data source, then transforms, discovers the key features and knowledge from each document. Finally, stores these data as RDF in the System Repository.

Meanwhile, the Semantic Query Analysis module is responsible for receiving, processing, displaying the result of user search query through the Web application and APIs layer. It analysis the semantic relationships between the query words and the semantic meaning which the users wants to express, next step is to optimize the query with the help of semantic relationship in ontology base. After that, the query will be transformed into the query language of ontology, i.e., the SPARQL language. The SPARQL query will scavenge the data from System Repository. Finally, the discovered knowledge and information will be put on display to the user.

Fig. 1. The high-level architecture.

Fig. 2. The Knowledge Discovery and Retrieving Module.

Both modules interact with the user through Web application & APIs layer and integrate with highly specialized Big Data analytics tools (e.g. Spark streaming, Apache Hadoop, MongoDB and Kafka)^[29-31].

3.1 Knowledge Discovery and Retrieving Module

The module is created out of many components that integrated in order to find the required knowledge from the required data source and store the result in Apache Jena^[32] as RDF triple as shown in Fig 2.

3.1.1 Semantic Search Engine

The whole process in our model is managed by controller agent. It manage the data transmission procedure in different stages of data life cycle as follows: data collection from heterogeneous sources for storage, data integration of multiple data sources, classification of the retrieved documents, data management for transferring the integrated data to processing platforms and finally, data analysis for moving data from storage to analyzing host.

Web crawling is the main and central function of the semantic search engine. It is scalable and runs multiple processes in parallel to reach the maximum download rate while keeping the overhead from parallelization at minimum and to avoid repeated downloads of the same page. The agent is used for web scraping and collecting data from various heterogeneous sources recursively based on the source structure.

Our web crawler has the ability to scrape all pages that are: unstructured, semi-structured and structured such as HTML pages along with PDF and Doc files. For every crawled URL, we save two kinds of data:

- *URL name:* It saves in MongoDB to prevent duplicate processing of the same URL.

- *Content:* Based on the meta-tag ‘content-type’, we use multiple support agents to read the content, then store it in the Hadoop Distributed File System (HDFS).

In our case, the MongoDB is utilized as the “operational” real-time data store and Hadoop is utilized for offline batch data processing and analysis.

3.1.2 Data Normalization and Transformation

The size of the content is huge depending on the depth of the site structure. Additionally, these data are from multiple heterogeneous data sources, the formats are inconsistent and less structured. Thus, the first step of analysis is developing a repository for the data, that’s capable of storing high-throughput any type of data. This is where HDFS come in. HDFS acts as a centralized, unified repository for data before the start of analysis and allows distributed processing of large data sets across clusters of computers. Therefore, the extracted content from each URL is stored into HDFS by connecting to Hadoop cluster.

3.1.3 Multiagent Layer

This layer consists of three support agents with a specific task based on the content type as follows:

- *Mobile Agent for Unstructured Data:* it is responsible for extracting the content of HTML, PDF and Word Processing Files.
- *Mobile Agent for Semi-Structured Data:* It is responsible for the XML documents and NoSQL database content extraction.
- *Mobile Agent for Structure Data:* It is responsible for the content extraction from relational databases.

3.1.4 Data Integration

Our semantic platform provides a real time processing of crawling pages. To build a large-scale data pipeline, we need to combine Spark streaming with another tool that addresses data integration challenges which is Apache Kafka. It is scalable, fault-tolerant and partition support that could read multiple Kafka events queue concurrently from multiple partitions. It acts as a message queue which follows producer-consumer pattern (or publish/subscribe) for processing messages. Kafka behaves as a form of universal pipeline for data and becomes the backbone for data flow between systems of all kinds, the core pipeline for Hadoop data, and the hub for stream processing and real-time applications. Once the process of saving content to HDFS file gets completed, crawler agent will be integrated with Apache Kafka and writing events to Kafka connect cluster which is stored as Kafka topic that contains the following information:

- The URL we scraped.
- URL HDFS's content location.
- University name.
- HDFS URI (Hadoop cluster remote location- host name and port number)
- Timestamp (current time).

This information is used to re-read the data from HDFS and continue processing by Spark Streaming in the next stage.

3.1.5 Data Cleansing and Preprocessing

In this stage, we use a high performance platform, distributed data processing engine in which we can define our Kafka Consumer to read messages from Kafka topics that is Spark Streaming. Once a Spark Streaming has received its input data from Kafka topic, the content of the crawled page will be read from the Kafka

event which contains the HDFS location of the URL. Thus, the Spark streaming will do the following:

- *Data Cleaning:* The goal of this step is to remove the noise, irrelevant, and unnecessary parts of the document. Thus, all Hypertext links, images, scripting code, HTML comments and unwanted lines are removed.
- *Data Pre-processing:* In this step, Spark streaming starts the natural processing using machine learning algorithms to process the stream of the data in a predefined order as follows: (1) Tokenization, (2) Sentence Splitting, (3) Syntactic analysis (Part of Speech Tagger), (4) Morphological analysis, (5) Stop-word removal.

3.1.6 Filtering and Categorization

Filtering and Categorization step is the important part that further analysis is relying on. It depends on the combination of using Natural Language Processing techniques (NLP), Python Machine Learning API's, statistical algorithms, and classification algorithms to assign one or more predefined categories to documents. N-gram model, TF-IDF (Term Frequency – Inverse Document Frequency) and Multinomial Naive Bayes (MNV) classification algorithms are used.

The document classification is used as a document filtering based on specific features. Then, it identifies the main topics that the document covers for document categorization. We build our classification model to train the features based on the supervised learning algorithm which assorted the documents based on two levels:

- *Level 1 (Domain related features classification model):* The classification rules are generated based on the corpus created which contains few expected

things in a faculty profile. These features include: Personal information, Academic Experience, Publication, Email and Educational qualification, etc. We assign to each feature a score. Thus, if the document has scores of 7 or more it is classified as faculty profile.

- *Level 2 (Research Area classification model):* There is a chance that the number of research areas of faculty work may be more than one. Thus, top three research areas are extracted based on our classification models.

3.1.7 Feature Extraction

Feature Extraction is the process of discovering important features from documents for further automatic querying, processing to make predictions and decisions and data storage. The goal of this step is to find the desired pieces of information, such as concepts, entities (name, numeric expressions, date) in natural language texts, then prepare these features to store them in a form that is suitable for the automatic querying and processing.

3.1.8 Data Storage and Management

The Data Storage and Management stage, is dedicated to integrating multiple features together from multiple sites to arrive at a unified view and a standard format that is suitable for the automatic querying, data analysis, exchange of data and evaluation. Thus, all discovered knowledge from the previous steps includes: faculty name, Email address, top three research areas, publication titles and other key phrases are converting to the standardized data structure by W3C called RDF. Then, Spark streaming will store the RDF triples in Apache Jena using Jena API. Apache Jena is a Java open source framework for building Semantic Web and

Linked data applications. Finally, Spark streaming call the REST API to update the retrieval status of the document to true in MongoDB that indicate the source is a computer science document for statistical purposes.

3.2 Semantic Query Analysis Module

This module is responsible for receiving user requests, generating, and constructing a query in ontology base for presenting the discovered knowledges according to user query in an effective and efficient way. The goal of the semantic engine layer is to pass the request through a series of reviews that will improve the quality of the data mining result. Thus, the results obtained will be accurate enough to satisfy the request made by the user. It has multiple subcomponents to complete its mission. Figure 3 shows an overview of the processes used in this module.

3.2.1 Query Analysis

It is responsible for receiving user requests and displaying the system outputs. In this phase, the analytical operation of the query is performed to determine what terms it should search for. It includes a series of operation to identify the subject, verb, noun, synonym, phrases and adjectives from the question. Tokens are separated and the meaning is analyzed. Then, a list of terms that includes the spell-corrected root forms of the original search terms and any synonyms for these terms are sent to the next stage.

3.2.2 Query Optimization

In this phase, we use ontology that contains all possible nodes related to user request terms in a specific domain to infer those data with implied semantic association and get all data with related relationships. Thus, it will increase search space and get more accurate results.

Fig. 3. The Semantic Query Analysis Module.

When the query terms get mapped to a concept in the ontology, the query is expanded in terms of its definition, which captures the necessary and sufficient classes and its inferred subclasses based on the relations and hierarchy of concepts defined in the ontology. Then, create a vocabulary list for the further query processing step.

For example, if the user inputs a sentence: “machine learning and data analytics”, the analytical result will be as follows: First, remove stop words from which the query will give us four words = ["machine", "learning", "data", "analytics"]. Second, we are creating combination of two words in contiguous locations to compare it with our research areas in ontology such as ["machine_learning", "learning_data", "data_analytics"]. Third, combining ["machine", "learning", "data", "analytics"] and ["machine_learning", "learning_data", "data_analytics"] into a single list and then compare it with research areas stored in the ontology and if it matches with any research

area then it will find all subclasses of that research area in recursive method. Expanding the concept in terms of other inferred knowledge (i.e., properties, subclasses, instances) ensures that the documents are retrieved in the right context.

Following the example above, the expanded query now retrieves the documents which speak of term such as “computer vision”, “Natural Language Processing”, “Robotics” and “Pattern Recognition” etc., even in their absence in the query terms. Otherwise, If the terms in the query are not related by any of the relationships discussed above, the keyword based search is performed which can be treated as the default case.

3.2.3 Query Processing

This subcomponent will receive a list of vocabulary obtained from the previous steps, construct the SPARQL Query [33], processes it against the RDF triples in data storage and then returns the answer to the users. Apache Jena was used for executing SPARQL queries. The Query Processor fetches RDF data using the Jena API which is used for mapping

SPARQL with RDF database and retrieving the relevant information, performing a semantic search into the database.

3.2.4 Query Result Ranking

For each query:

1. Fetching the RDF's based on the (user search query and expanded query).
2. Scoring the words in the RDF based on TF-IDF Vectorizer.
3. Get the words from user query and expanded query.
4. Give the scores to user query words and expanded query (based on ontology relations) words as 1 and 0.5.
5. Multiply the Query word scores with the document word scores. Both the words are same, but scores would be different.
6. Same process for expanded query words.
7. Sum up both to give a final score for a document.

Normalization is not done in the process above since the TF-IDF Vectorizer gives the normalized score for words in the RDF document we have considered.

4. Implementation

4.1 Tools

Our proposed system is implemented by using multiple open source tools and web service APIs. We build the system as a web application. Table 1 displays the technical infrastructure that is used to implement our system.

Table. List of Technical Infrastructure.

Development environment	Eclipse for Java Development. PyCharm for developing Python, Machine Learning Algorithms
-------------------------	--

Software	MongoDB 3.2.9 JDK 1.8 (oracle) Maven 3.3.9 Hadoop (version: - 2.6) Spark 1.6.0 Python 3.4.3 Apache Kafka 0.8.2.1-Scala-2.11 Zookeeper Ruby 2.2.3 Apache Jena Fuseki Server 2.5.0
Hardware	Hadoop 2 Nodes: a) Master Node (Name Node) - 16 GB RAM with 4 CPU/VCores b) Slave Node (Data Node) - 16 GB RAM with 4 CPU/VCores with 50GB Hard disk Spark One master node and 4 workers Master Node: 16 GB RAM with 4 CPU/VCores 4 Workers are having 16 GB RAM with 4 CPU/VCores Kafka Single Node Cluster 4 GB RAM with 8 Partitions and single topic. Web Crawler Service API Running on 8 GB Heap Space, RAM and 4 CPU/VCores
Infrastructure	Cloud-based

4.2 Ontology Development

We used Protégé to build the research area domain ontology for computer science as shown in Fig. 4^[34]. Ontology is created using OWL. We consider the 20 research areas and 317 sub-area which were considered during training the model to extract research area and the related concepts. The ontology stored in Apache Jena, where the RDF's were stored. An ontology consists of key elements:

- *Class and Class Hierarchy*: Classes are the main concepts and focus of most ontology. It represents the main research area in the computer science. Each class has sub-class that represents a more specific research area than the super-class.

- *Relationship*: We define multiple kinds of relationships between classes (or sub-

classes) that are done by relationship rules such as: Is- a, Equivalent, Disjoint, use.

4.3 Training Data

1) *Training Data for Faculty Profiles:* 16 departments and 20 profiles from each department are considered for creating training data from Pondicherry University, Harvard University and King Abdulaziz University.

Before modeling the data, we split the data set for training and testing based on the department tags with training and test set split being 50% each. The accuracy of the model was 85%.

2) *Training Data for Research Area:* In this section, we discuss how to create the research area corpus in computer science. Since the faculty usually does his research in different research area, we will build a classifier to extract only the top three research areas. The research areas corpus is created for supervised learning considering 20 research areas that were collected from research on IEEE explore, science-direct and computer Journal.

Finally, we have modeled the data to extract top three research areas for a profile. We apply the similar steps for processing and defining the training data in the previous section to extract research areas too. The accuracy of the test data was 78%.

4.4 Feature Extraction

1) Named Entity Feature

- Stanford CoreNLP library (NER) ^[35]
- A Ruby Anystyle - parser to extract all author names from the publication section, and then compute the frequency of each one of them in the document ^[36].

2) *Email:* We use regular expression for representing patterns that match Email

expression to discover from the document as follows:

$(r'[\backslash w\backslash.-]+@[\backslash w\backslash.-])$

3) *Research Area:* Based on the classification model, we extract the top 3 research areas

4) *Research Publication:* Ruby Anystyle-Parser is used to extract bibliographic information such as: author names, title, location, publisher, edition, year, etc.

5. Result and Discussion

The evaluation was conducted in terms of measuring the query matching result quality and measuring the impact of performance on request processing response time.

5.1 The Performance of the Model

1) Web Crawler Speed

As shown in Tables 2, we found that the external website's servers affect the web crawler speed. We notice that some website servers have very slow response such as <http://tce.edu/cse/people>. This may be due to increased load on the server that is slowing down response time due to maxed out resources (processors, memory, disk space, bandwidth). Thus, web crawler performance depends upon the web server of university from which we are extracting faculty profile and the current load on their server while we are extracting pages from their websites. By observing the log files of the crawler job, we get the following:

- 230 pages per minute.
- 80 pages per minute.
- 130 pages per minute.

The average will be about 100 pages per minute with 8 GB and 4 CPU core processor. It can be up to (1000-5000) pages per minute

in case of the websites having quick response time.

Table 2. Web crawler speed.

ID	URL	URL Scrapped	URL Retrieved	Time Taken (min)
1	http://demba-ba.org/	32	1	2 Sec
2	https://www.cs.cornell.edu/people/faculty	111599	202	60
3	http://cs.stanford.edu/directory/faculty	11170	105	35
4	http://www.angio.net/personal/	317	3	4
5	http://eecs.oregonstate.edu/people/faculty-directory	1594	15	10
6	http://tce.edu/cse/people	6260	212	39

2) Spark Streaming Performance

This paper deals with the deployment of Spark cluster as a cloud service on the Amazon Web Service (AWS). The operation that is done by Spark Streaming Job as follows:

- Reads the Kafka Message after storing document content in HDFS.
- Acquisition and preprocessing by reading the data from HDFS based on Kafka Topic.
- In Filtering and Categorization step for classification algorithm based on two levels:
 - Level 1: Domain related features classification model.
 - Level 2: Research area classification model.
- Data Extraction.
- RDF is created from the data above.
- RDF stored to the Apache Jena.

- Update status of URL to “isRetrieved=true” in MongoDB.

We measure the performance of Spark job with different numbers of workers to study the effect on the time and speed of the processing steps as shown in Table 3.

Table 3. Spark streaming performance

Spark Mode	No. of Workers / Executor	TABLE 1 No. OF PAGES PROCESSED PER MINUTE
The standalone mode cluster will run in FIFO (first-in-first-out) order	1 Worker / 4 Executors (4 GB, RAM each and 1 CPU per executor)	30 pages
Cluster	1 worker / 1 Executor (8GB RAM and 4 CPU each)	30 pages
Cluster	2 Worker / 2 executors (8 GB RAM and 4 CPU each)	80-100 pages
Cluster	3 workers / 3 Executors (8GB RAM and 4 CPU each)	100-200 pages
Cluster	4 workers / 4 Executors (8 GB RAM and 4 CPU each)	150-300 pages

The results illustrate that there is a positive relationship between the number of Spark workers and the processing page. Increasing the number of workers leads consequently to the parallelization of Spark jobs, reduce the time, and speed up the processing step as shown in Fig 5. As seen in Table 3, we haven't used multiple executors per worker. Because we have a lot of processes for one Kafka messages as we mentioned above in the spark job flow. Thus, we focused on giving more memory per executor instead of creating multiple executors having less RAM.

5.2 The Impact of the Ontology

In the following experiment, we examine the effect of using the ontology to improve the accuracy and expand the search space. As

shown in Table 5, the ontology-based semantic discovery can detect more results related to the user's query that in some cases users do not know the sub area of a specific domain. In this case, the ontology-based semantic discovery can return more results and expand the probability of information that the user needs. Furthermore, the retrieval effectiveness is tested based on standard information retrieval metrics^[38], i.e., Precision and Recall.

Precision is the part of a search result that is retrieved and relevant to a query. Its calculation needs knowledge of the relevant and non-relevant documents in the evaluated set of documents^[39]. That's why it is possible to calculate absolute precision of search engines which provide an indication of the relevance of the system as follows:

$$Precision = \frac{\text{number of relevant document}}{\text{number of all retrieved document}} \times 100\% \quad (1)$$

To define the relevance of each page, a four-point criteria was used which enabled us to calculate precision. The criteria employed for the purpose are as follows:

- A page representing the full text of faculty profile is considered.
- A page representing a list of faculty profiles are considered.
- A page representing anything other than the faculty profile (i.e. Course syllabus, publication, research paper, project page, seminar/conference, etc.) is not considered.
- A page occurring more than once under different URL structure is not considered.

On the other hand, the recall is the ability of a retrieval system to gain all or most of the relevant documents in the collection. Recall is the number of relevant documents divided by the number of all the needed

documents from the database. Thus, it requires the entirety of the needed knowledge from the database not just of the relevant and retrieved. There is no one appropriate method of determining the absolute recall of search engines as it is impossible and difficult to know the total number of relevant pages in huge databases. In our experiment, we adopted the classic recall measurement for use in the Web environment by giving it a relative value. The relative recall value is thus defined as:

$$Recall = \frac{\text{number of relevant document}}{\text{number of all needed documents}} \times 100\% \quad (2)$$

Figures 6 and 7 show the comparison of Precision and Recall between ontology-based semantic discovery and keyword.

Based on the approach proposed in this paper, it shows that the ontology-based semantic mechanism (including the ontology construction, annotation, semantic retrieval, and the framework) can give back more useful results related to the needs of the users. In particular, when users' query conditions are not the precise and single word, the recall of the ontology-based semantic retrieval will rise and satisfy the users' needs.

6. Conclusion

In this paper, we created a unified semantic based approach to improve the knowledge discovery process from Big Data on the web. The model transformed the extracted information from the heterogeneous data sources (unstructured, semi-structured or structured) into RDF format and stored them in a unified structure to enable easy machine processing data from separate sources and make it easily searchable, maximizing discovery, analytics, and reporting to the users. The model uses domain ontology to get efficient and effective results by understanding the users' objective and the contextual meaning of terms as they show in the searchable data space using ontology. The model used a variety

of methodologies, techniques, and tools to ease the transformation of data into value. This approach leads to quick conclusions and new knowledge discovery from the set of data.

The system is currently capable of handling simple queries with any syntax because it is partially implemented. Thus, further development would be needed to answer all possible types of queries. Additionally, there are many areas for further

exploration. First, extending an existing ontology with relevant properties, classes, and relationships. Second improvement is extending the train set size to optimize the classification models and get more accurate results. Finally, the crawler agent needed to learn a new technique to extract and automatically update the content from those web pages periodically.

Fig. 4. The class and subclass on ontology.

Fig. 5. Spark Streaming performance.

Table 5. Impact of ontology.

Search query	Keyword result	The correct Result	Ontology-based result	The correct results based ontology
Artificial intelligence and machine learning	35	8	63	20
Networking	55	31	268	165
Computer graphics and visualization	45	22	327	176
Computer architecture and design	105	74	246	183

Fig. 6. Comparison of Precision.

Fig. 7. Comparison of Recall.

References

- [1] **Kaisler, S., Armour, F., Espinosa, J. A. and Money, W.** (2013) "Big data: Issues and challenges moving forward," in: *System sciences (HICSS), 2013 46th Hawaii international conference on 2013*, pp: 995-1004.
- [2] **Sagiroglu, S. and Sinanc, D.** (2013) "Big data: A review," in: *Collaboration Technologies and Systems (CTS), 2013 International Conference on, 2013*, pp: 42-47.
- [3] **McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. and Barton, D.** (2012) "Big data," The management revolution. *Harvard Bus Rev*, **90**: 61-67.
- [4] **Cisco, I.** (2012) "Cisco visual networking index: Forecast and methodology, 2011–2016," *CISCO White paper*, pp. 2011-2016,.
- [5] **Gantz, J. and Reinsel, D.** (2012) "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the Future*, vol. 2007, pp:1-16.
- [6] **Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, and M. M. Teixeira,** *New Advances in Information Systems and Technologies* vol. 444: Springer, 2016.
- [7] **Beyer, M. A. and Laney, D.** (2012) "*The importance of 'big data': A definition*," Stamford, CT: Gartner, pp. 2014-2018.
- [8] **What Is Apache Hadoop?** (2016) [Online]. Available: <http://hadoop.apache.org/> [Accessed: 23- Mar- 2017].
- [9] **Hitzler, P. and Janowicz, K.** (2013) "Linked Data, Big Data, and the 4th Paradigm," *Semantic Web*, **4**: 233-235.
- [10] **Monticolo, D., Lahoud, I. and Bonjour, E.** (2012) "Distributed knowledge extracted by a MAS using ontology alignment methods", In: *Computer & Information Science (ICCIS), International Conference on, 2012*, pp: 386-391.
- [11] **Chen, L., Shadbolt, N. R. and Goble, C. A.** (2007) "A semantic web-based approach to knowledge management for grid applications," *IEEE Transactions on Knowledge and data Engineering*, **19**.
- [12] **OWL: Web Ontology Language.** (2017) [Online]. Available: <https://www.w3.org/TR/owl-ref/> [Accessed: 3-Jan- 2017].
- [13] **RDF.** (2017) [Online]. Available: <https://www.w3.org/RDF/> [Accessed: 23- Jan- 2017].
- [14] **Begoli, E. and Horey, J.** (2012) "Design principles for effective knowledge discovery from big data", In: *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 joint working IEEE/IFIP conference*, pp: 215-218.
- [15] **Wu, X., Wu, X. G. Q. and Ding, W.** (2014) "Data mining with big data", *IEEE Transactions on Knowledge and Data Engineering*, **26**: 97-107.
- [16] **Dou, D., Wang, H. and Liu, H.** (2015) "Semantic data mining: A survey of ontology-based approaches", In: *Semantic Computing (ICSC), IEEE International Conference on, 2015*, pp: 244-251.

- [17] **Jirkovský, V.** and **Obitko, M.** (2014) "Semantic Heterogeneity Reduction for Big Data in Industrial Automation," in: *ITAT, 2014*.
- [18] **Thirunarayan, K.** and **Sheth, A.** (2013) "Semantics-empowered approaches to big data processing for physical-cyber-social applications," in: *Semantics for Big Data: Papers from the AAAI Symposium. AAAI Technical Report FS-13-04, 2013*, pp: 68-75.
- [19] **Gundla, N. K.** and **Chen, Z.** (2016) "Creating NoSQL Biological Databases with Ontologies for Query Relaxation", *Procedia Computer Science*, **91**: 460-469.
- [20] **Kang, L., Yi, L.** and **Dong, L.** (2014) "Research on construction methods of big data semantic model," in: *Proceedings of the World Congress on Engineering, 2014*.
- [21] **Jadhao, H., Aghav, D. J.** and **Vegiraju, A.** (2012) "Semantic tool for analysing unstructured data", *International Journal of Scientific & Engineering Research*, **3**.
- [22] **Al Zamil, M. G.** and **Samarah, S.** (2014) "The application of semantic-based classification on big data," in: *Information and Communication Systems (ICICS), 2014 5th International Conference on, 2014*, pp: 1-5.
- [23] **Abirami, A.** and **Askarunisa, A.** (2013) "A semantic Based Approach for Knowledge Discovery and Acquisition from Multiple Web Pages Using Ontologies," *International Journal of Web & Semantic Technology*, **4**: 57.
- [24] **Scheutz, M.** and **Mayer, T.** (2016) "Combining Agent-Based Modeling with Big Data Methods to Support Architectural and Urban Design", pp. 15-31,.
- [25] **Coneglian, C. S.** and **Fusco, E.** (2014) "Semantic Agent of Informational Extraction on Big Data Ontological Context", *The Sixth International Conference on Information, Process, and Knowledge Management*, ISBN: 978-1-61208-329-2
- [26] **Diosteanu, A.** and **Cotfas, L.** (2009) "Agent based knowledge management solution using ontology, semantic web services and gis", *Informatica Economica*, **13**: 90.
- [27] **Hu, H, Wen, Y., Chua, T. S.** and **Li, X.** (2014) "Toward scalable systems for big data analytics: A technology tutorial", *IEEE Access*, **2**: 652-687.
- [28] **Adrian, M.** and **Chamberlin, D.** (2012) "Orbitz Worldwide Uses Hadoop to Unlock the Business Value of "Big Data." Gartner".
- [29] *Apache Spark.* (2017) [Online]. Available: <https://spark.apache.org> [Accessed: 2- Feb- 2017].
- [30] *What Is MongoDB?*, ()2017. [Online]. Available: <https://www.mongodb.com/what-is-mongodb> [Accessed: 4- Feb- 2017].
- [31] *Apache Kafka.* (2017) [Online]. Available: <https://kafka.apache.org/> [Accessed: 23- Feb- 2017].
- [32] *Apache Jena.* (2017) [Online]. Available: <https://jena.apache.org/> [Accessed: 23- Jan- 2017].
- [33] *SPARQL 1.1 Query Language.* (2017) [Online]. Available: <https://www.w3.org/TR/sparql11-query/> [Accessed: 15- Jan- 2017].
- [34] *Protégé.* (2017) [Online]. Available: <http://protege.stanford.edu/> [Accessed: 23- Jan- 2017].
- [35] *Stanford Named Entity Recognizer (NER).* (2017) [Online]. Available: <https://nlp.stanford.edu/software/CRF-NER.shtml> [Accessed: 19- Feb- 2017].
- [36] *Ruby AnyStyle Parser.* (2017) [Online]. Available: <https://anystyle.io/> [Accessed: 20- Feb- 2017].
- [37] **Van, C. J.** (1986) "A new theoretical frame-work for information retrieval", *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy*, pp: 194-200.
- [38] **Chang, Y. S., Juang, T. Y., Chang, C. H.** and **Yen, J. S.** (2012) "Integrating intelligent agent and ontology for services discovery on cloud environment", *IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, 2012*, pp: 3215-3220.

اكتشاف المعلومات من البيانات الكبيرة على الإنترنت بواسطة تقنية الوكيل الدلالي

رانية بافقيير، و فتحي البرعي، و كمال جمبي

قسم علوم الحاسبات، كلية الحاسبات وتقنية المعلومات، جامعة الملك عبدالعزيز، ص. ب. ٨٠٢٠٤ جدة،

٢١٥٨٩، المملكة العربية السعودية

rbafaqeer@kau.edu.sa

المستخلص. نشأ ظهور البيانات الضخمة نتيجة النمو السريع لمصادر الويب على شبكة الإنترنت. حيث أصبحت البيانات غير متجانسة وحجمها ضخم جداً، نتيجة التحديث المستمر والتأثر من التطور السريع للشبكات والقدرة على جمع البيانات وتخزينها. أصبحت نظم المعلومات التقليدية غير قادرة على التعامل بكفاءة مع جميع مصادر البيانات الضخمة وسياقات المعلومات المتعددة التي تعتمد على الإنترنت كمنصة أساسية لها. حيث تواجه هذه النظم التقليدية العديد من الصعوبات والتحديات في استرجاع وتوحيد وتخزين ومعالجة واستخدام المعلومات الناتجة عن مصادر المعلومات المختلفة وغير المتجانسة والتي هي أساس العديد من المنظمات في عملية اتخاذ القرار. في هذا البحث، قمنا بتطوير نهج الوكيل الدلالي الموحد القائم على علم الأنماط و(ontology) لتعزيز عملية اكتشاف المعرفة من البيانات الضخمة على شبكة الإنترنت. حيث يستخرج النموذج المعلومات من مصادر البيانات المختلفة المنظمة وغير المنظمة ويعمل على تخزينها في قواعد بيانات موحدة مما يسهل عملية البحث والتحليل واعداد التقارير. كما يساعد على استخراج نتائج أكثر كفاءة وفعالية من خلال فهم طلب المستخدمين والمعنى السياقي للمصطلحات باستخدام ontology. ولتقييم هذا العمل المقترح، أجريت العديد من التجارب لتقييم الأداء، ودقة النظام باستخدام مقاييس الأداء المختلفة. ولقد أظهرت النتائج أن استخدام تقنية ontology حسن من نسبة كفاءة الأداء ودقة النتائج بشكل أفضل من طريقة اكتشاف المعلومات التي تستند إلى البحث عن الكلمات الرئيسية.

الكلمات المفتاحية: البيانات الضخمة، الويب الدلالي، اكتشاف المعرفة، علم الوجود، المزاح الدلالي.