3.1 INTRODUCTION

In this era, search engines are acting as a vital tool for users to retrieve the necessary information in web searches. Web is huge but it is not intelligent enough to understand the queries made by the user and relate them real or abstract entities in the world. It is a collection of text documents and other resources, linked by hyperlinks and URLs. The retrieval of web page results are based on page ranking algorithms working in the search engines. It also uses the statistical based search techniques or content based information extraction from each web pages. But from the analysis of web retrieval results of Google like search engines, it is still hard for the user to understand the inner details of each retrieved web page contents unless otherwise the user open it separately to view the web content.

This key point motivated to propose and display an ontology based O–A–V (Object–Attribute–Value) information extraction for each web pages retrieved and that will impart knowledge for the user to take the correct decision. Also, it helps as a user dictionary to refine his search key based on O–A–V triplet that may improve the retrieval result for the next attempt. Also, the proposed system parses the users natural language sentence given as a search key into O–A–V triplets and converts it as a semantically analyzed O–A–V using the inferred ontology. This conversion procedure involves various proposed algorithms and each algorithm aims to help in building the taxonomy. The ontology graph has also been displayed to the user to know the dependencies of each axiom given in his search key. The web document retrieval performance based on this proposed
method is evaluated using the appropriate metrics.

3.1.1 SEMANTIC WEB

Semantic web is the next level of web which treats it as a knowledge graph rather than a collection of web resources interconnected with hyperlinks and URLs. It also aims at adding semantic content to web pages and providing machine processable semantics. With semantically annotated web resources, the machine interpretable semantic web agents will be able to perform complex operations on behalf of the user using this semantically interconnected web of resources. Semantic Web is about common formats for integration and combination of data drawn from diverse sources and how the data relates to real world objects. It provides a common framework that allows data to be shared and reused across applications, enterprise and community boundaries (Berners-Lee et al., 2001).

Heath and Bizer (2011) describes a method of publishing structured data so that it can be interlinked and become more useful. Rather than using web technologies to serve web pages for human readers, it uses them to share information in a way that can be read automatically by computers enabling data from different sources to be connected and queried (Bizer et al., 2009).

Reasoning is the capacity for consciously making sense of things, applying logic, for establishing and verifying facts, and changing or justifying practices, institutions, and beliefs based on new or existing information (Kompridis, 2000). With semantic web intelligent, web agents will be able to reason the content on web and draw inferences based on the relations between various web resources.

3.1.2 ONTOLOGY

Ontology is the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations. Any entity whether real or abstract have certain characteristics which relate to certain entities in the real world and interact with them. Ontologies deal with the existence of entities, organizing them into groups based on their similarity, developing hi-
erarchies and studying relations among them which allows us to draw inferences based on their classification. It also deals with a study of how entities interact with other discrete entities in the world and finally develop our own ontologies based on our domain of interest. In computer science and information science, an ontology formally represents knowledge as a set of concepts within a domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts (Gruber, 1993).

Ontologies act as the building blocks for the infrastructure of semantic web. They will allow to transform the existing web of data into web of knowledge. They will allow for knowledge sharing among the various web applications and enable intelligent web services.

3.1.3 KNOWLEDGE REPRESENTATION

Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain (Sowa, 2000). Knowledge representation and reasoning are the backbones of semantic web. There is no absolute knowledge representation methodology, it solely depends on the type application and how it uses the acquired knowledge.

- The computer system uses knowledge stored in Knowledge Base (KB) which is represented in a formal sensible way.

- The following sections describes the various basic knowledge representation models.

3.1.4 LOGIC

Logic is a method of reasoning that helps to prove an argument. Logic works by evaluating the exactness of a gathering of statements. For Example,

Socrates is a man.
All men are mortal.
Therefore, Socrates is mortal.
3.1.4.1 PROPOSITIONAL LOGIC

The true or false propositions which are framed by other propositions with the utilization of legitimate connectives are such as, and, or, not, and if. The following is an example of primitive inference within the scope of propositional logic:

Premise 1: If it’s raining then it’s cloudy.
Premise 2: It’s raining.
Conclusion: It’s cloudy.

3.1.5 PREDICATE LOGIC

For symbolic formal frameworks, the predicate logic is the nonexclusive term in mathematical logic. It is described using two common quantifiers, they are

- Universal ∀ “for all"
- Existential ∃ “there exists”

Consider the following example,

Suppose we have a set

p(x) = "x is an even number"

3.1.5.1 UNIVERSAL QUANTIFIER ∀

It communicates that a propositional function can easily fulfilled by each individual of a domain of discourse. In another terms, it is the predication of a characteristic or relation to each individual of the domain.

\[ \forall(x)P(x) \]  
\[ (3.1) \]

- For all \( x \) \( P(x) \)
- \( P(x) \) is true for all possible \( x \) in universal set.
• If the set $X$ contains only odd numbers, $X = 1, 3, 5, 7$ then the universal quantifier can be written as $\forall x(x \in X \rightarrow P(x))$

### 3.1.5.2 EXISTENTIAL QUANTIFIER $\exists$

Existential quantifier is distinct from universal quantifier ("for all"), which states that the property or relation holds for all individuals in the domain.

$$\exists(x)P(x) \quad (3.2)$$

• There exist an $x$ such that $P(x)$

• $P(x)$ is true for at least one possible $x$ in universal set.

• If the set $X$ contains some odd and even numbers, $X = 1, 2, 3, 4, 5$ then the existential quantifier can be written as $\exists x(x \in X \land P(x))$

### 3.1.6 RULE BASED SYSTEMS

Rule-based frameworks comprise of a set of rules or standards, a working memory and an inference engine. The standards encode domain knowledge as direct condition-action pairs. These frameworks normally utilize a working memory that initially contains the input data for a specific run, and an inference engine to discover relevant rules and apply them.

#### 3.1.6.1 INFERENCE ENGINE

Defines how existing knowledge may be used to derive new knowledge.

Rule-based systems are also called condition-action rules. These components of a rule-based system have the form:
if <condition> then <conclusion>
or
if <condition> then <action>

Example-1: Condition-Action Rule

IF GMAT score >= 600
THEN
Admit student to MBA program
ELSE
do not admit student.

Example-2: Conjunctive Condition Clauses

IF sky is clear AND temperature is low
THEN
chance of frost is high

The rules can be evaluated by:

- backward chaining
- forward chaining

3.1.6.2 BACKWARD CHAINING

- To figure out whether a decision ought to be made, work in reverse searching for justifications for the choice. It begins with a set of goals.
- It is obvious that the facts are used to justify decision.
3.1.6.3 FORWARD CHAINING

- The taken facts flow forward in inference net based on decisions.
- Finds what conclusions can be obtained from information.

3.1.7 OBJECT–ATTRIBUTE–VALUE REPRESENTATION

- Object-Attribute-Value Triplets used to represent facts about objects and their attributes.
• Uncertain facts represent uncertainty about facts in O–A–V Triplets.

• Fuzzy facts represent uncertainty about facts using terms from natural language.

• Semantic networks represents semantic relations between concepts. They are designed after the psychological model of the human associative memory and attempt to reflect cognition.

3.1.8 WORDNET

WordNet (Fellbaum, 2010; Miller et al., 1990) is a large lexical database of English Language. It groups closely related words into unordered set called Synsets which are interlinked by means of conceptual-semantic and lexical relations. While considered an upper ontology by some, is not strictly an ontology. However, it has been employed as a linguistic tool for learning domain ontologies. Also, Wordnet is a system of words connected by lexical and semantic relations.
Composing lexical data as far as word significance, as opposed to word structure, so it can likewise be utilized as a thesaurus.

The primary Wordnet in the globe was for English developed at Princeton over 15 years. The EuroWordNet - connected structure of European language wordnets was created in 1998 as a funded project for more than 3 years. Wordnets for Hindi and Marathi being worked at IIT Bombay are amongst the main Indian language wordnets. All these are proposed to be connected into the IndoWordnet which in the end will be connected to the English and the Euro wordnets. The Indian linguistic Wordnet structure is shown in Fig. 3.5.

![Indian Linked Wordnets](image)

Fig. 3.5: Indian Linked Wordnets

By writing a Python script one can easily retrieve the meaning of a particular word from Wordnet easily as shown in Fig. 3.6.
3.1.9 RESOURCE DESCRIPTION FRAMEWORK (RDF)

Resource Description Framework (McBride, 2004) is an official W3C Recommendation for semantic web data models. RDF and RDFS can be used to design an efficient framework to describe various resources on the web in such a way that they are machine understandable. A resource description in RDF is a list of statements (triplets), each expressed in terms of a web resource (an object), one of its properties (attributes), and the value of the property. RDF is used to describe instances of ontologies, whereas RDF schema encodes ontologies providing the semantics, vocabulary and various relationships in the domain. A sample RDF created for faculty information database is shown in Fig. 3.7.

3.1.10 SPARQL

The name is a recursive acronym for SPARQL Protocol and RDF Query Language (SPARQL), which is described by a set of specifications from the W3C. The W3C, or World Wide Web Consortium, is the same standards body responsible for HTML, XML, and CSS. SPARQL is designed to query RDF, but you’re not limited to querying data stored in one of the RDF formats.

```
from nltk.corpus import wordnet as wn

#Get the Search keyword from the user
key = raw_input("Enter the keyword to Search: ")

#Find Alternate and Matching words of user input
synonym = []
for i in wn.synsets('love'):
    for lemma in i.lemmas:
        print lemma.name
        synonym.append(lemma.name)

#Call Search Method With Simple Query
result = api.GetSearch(term=key, per_page=count)
for i in result:
    print i.text
```
SPARQL is performing a pattern matching operation against RDF graphs. By first defining a triple pattern, one can easily define graph patterns. It is similar to a RDF triple, but with the choice of a variable instead of RDF terms (i.e., literals, IRIs or blank nodes) present in the subject, predicate or object places. The various building blocks of group graph patterns are basic graph patterns, filter conditions, optional graph patterns, alternative graph patterns and patterns on named graphs.

A basic graph pattern (BGP) is an arrangement of triple patterns (separated by a period if required). A basic graph pattern (BGP) is an arrangement of triple patterns (separated by a period if required). A BGP ought to be comprehended as the conjunction of its triple elements. An example of BGP is shown in Fig. 3.8.

\[
?x \text{ foaf:name } ?\text{name} . ?x \text{ foaf:mbox } ?\text{mbox}
\]

Fig. 3.8: An Example of Basic Graph Pattern

A group graph pattern is a collection of basic graph patterns delimited with braces \{ \}. An example of group graph pattern is shown in Fig. 3.9.
The group graph patterns can include different builds to be characterized underneath. By introducing certain keywords onto these constructs will help to impose constraints on SPARQL query. The conjunctive triples or BGPs are basically compared and afterward encased in { and } to frame a group graph pattern. A structure of a SPARQL query is shown in Fig. 3.10. Using this query, one can easily retrieve data which is stored in RDF format on the Web servers.

```
{ ?x foaf:name ?name . ?x foaf:mbox ?mbox }
{ ?x foaf:name ?name . ?x foaf:mbox ?mbox . }
{ { ?x foaf:name ?name . }
{ ?x foaf:mbox ?mbox . } }
```

Fig. 3.9: An Example of Grouph Graph Pattern

3.1.11 RDF Turtle Format

Information will be displayed utilizing Turtle. The Turtle structure is used in SPARQL so it is helpful to know it well. SELECT and WHERE provisions are similar to in SQL. But be cautious with the fact that the SPARQL and SQL constructs
are entirely different in their form. Variables are similar to in Prolog or Datalog. Variables can likewise be composed as $x$ rather than ?x. We can compose SELECT * like in SQL. The structure of a query is a collection of variables showing up in the SELECT clause. A sample RDF Turtle is shown in Fig. 3.11.

![Sample RDF Turtle Model](image)

Fig. 3.11: Sample RDF Turtle Model

### 3.1.11.1 SPARQL Queries for Turtle Model

The way of writing a sample SPARQL query for Turtle model is shown in Fig. 3.12 and 3.13. For an example, let’s say someone called my phone from the number 88766454542 and I didn’t answer. I want to know who tried to call me? by using the SPARQL query shown in Fig. 3.12.

```sparql
PREFIX ad: <http://www.anthoniraj.com/ns/addressbook#>
PREFIX id: <http://www.anthoniraj.com/ns/data#>
SELECT ?id ?name
WHERE
{ ?id ad:mobile "88766454542".
  ?id ad:firstName ?name.
}
```

Fig. 3.12: SPARQL Query-1: RDF Turtle Model

An another example, let’s say someone called my phone from the number 88766454542 and I didn’t answer. I want to know who tried to call me (ALL Details)? by using the SPARQL query shown in Fig. 3.13. It is obvious that the responses to a SPARQL query is a formalization of set of mappings i.e., partial
functions from the collection of variables in the SPARQL query to the collection of RDF terms (literals, URIs and blank nodes). We can assess SPARQL queries by making an interpretation of them into the algebraic language discussed in (Perez et al., 2006). The RDF triples used by SPARQL has many advantages such as schema flexibility, more modern, standardisation, expressivity and provenance over SQL used in relational databases.

### 3.2 PROPOSED O–A–V ARCHITECTURE

A typical sentence structure clause consists of a subject and a predicate, where the predicate is typically a verb phrase a verb together with any objects and other modifiers. A parse tree developed from a simple clause is shown in Fig. 3.14.

![Parse Tree Developed from a Simple Clause](image)

The breakdown of the clause based on the parse tree is shown in Fig. 3.15.
When passing the text through the proposed model shown in Fig. 3.16, it is broken down into clauses, which are then tokenized and passed through the WordNet analyzer. The WordNet analyzer provides characteristic properties for each lemma, such as the part of speech (POS), synonyms, hypernyms, hyponyms, etc. Later, an object is created for each of these individuals and is added to the ontology. When passing the clause through the triplet extractor, it continuously searches for nested and direct relationships using the existing ontology.

The extracted O–A–V triplets are then passed through a semantic analyzer, which determines the true form of the various objects in the O–A–V triplet based on the context where it has been used. These triplets and updated individuals are added to the ontology along with the generation of a taxonomy. At the end of all of these processes, a well-defined semantic network is developed, which can then be used to enhance search engine web results, providing the user with a completely reformed search experience.

3.2.1 ALGORITHM DESIGN

This architecture uses the following four algorithms. The algorithm-1 is used to develop an ontology from the content in a web document, the algorithm-2 is used to extract compound entities from NP and represents Object–Attribute–Value triplet. The algorithm-3 represents semantic analyzation of direct relations and the algorithm 4 is used to develop a taxonomy.
Algorithm 1 Developing an Ontology from the Content in a Web Document

1: extract clauses

2: while no more clause left do

3: analyze the clause and obtain NP and the VP;

4: obtain the last occurring V from the VP;

5: extract compound entities from the NP and the VP;

6: create O–A–V triplets between subjects and objects;

7: semantically analyze the extracted O–A–V triplets;

8: create a semantic network by adding the triplets and individuals to the ontology;

9: develop a taxonomy;

10: end while

For extracting nested relations, such as X’s Y’s Z, the triplet extractor continuously checks for relationships and creates empty individuals, which can later be updated based on their future occurrence. The individuals are then classified
based on the context where they are used, e.g., “Tommy” will represent a dog based on the relationship “Sam’s dog Tommy” but not on the convention that we have always used the name “Tommy” to refer to a dog.

Algorithm 2 Extracting Compound Entities from NP, O–A–V represents Object–Attribute–Value Triplet

1: while not end of NP do
2:     if next token $\notin N$ then
3:         create the current token as individual in ontology;
4:     else
5:         create O–A–V triplet between current token and next token with V as a combination of both;
6:         update current token with value of V;
7:         set class of V with the value of class of A;
8:     end if
9: end while

To analyze direct relations, such as X is Y, the semantic analyzer determines the group that both individuals belong to, compares them, and accordingly updates the O–A–V triplet based on previous occurrences of both the object and its value, as shown in Fig. 3.19.

Algorithm 3 Semantic Analysis of Direct Relations

1: if O $\in$ Ontology and O $\notin$ class of V then
2:     V represents a property or a characteristic of O rather than him;
3: else
4:     set class of O with the value of class of V of A;
5: end if

To develop a hierarchy among the various identified groups, hypernyms of all of the groups are acquired using WordNet (based on their usage) and common ancestors are determined for each entity going up the hierarchy level.
Algorithm 4 Developing a Taxonomy

1: while no Individual left do
2:   extract hypernyms for each individuals;
3:   arrange the individuals in order of appearance in their hierarchies;
4:   find common ancestors between the individuals up their hierarchies;
5:   add individuals to a common class having common ancestors;
6:   remove these individuals and add the ancestor as another individual in the given set;
7: end while

This process is continued until we reach the top-level entity (Thing). With all of the individuals classified into groups along with their relationships and a hierarchy, a taxonomy is develop, as shown in Fig. 3.23, 3.24, 3.25 and 3.26.

The doctor and doctor’s companion live in London. Xion is a country. Neo is a hacker. Neo lives in Xion. Neo is a bull.

Fig. 3.17: A Sample Query for Analysis

For parsing the sentences taken in Fig. 3.17 using the proposed algorithm, it generates the semantic networks shown in Fig. 3.18. The semantic analysis of direct relationships is shown in Fig. 3.19.
The OWL (Web Ontology Language) representation for the above semantic network is shown in Fig. 3.20. The entity recognition for unknown entities and known entities during the semantic analysis are shown in Fig. 3.21 and 3.22.

After analyzing the clause “Neo is a bull”, it determines the group to which Neo belongs using its previous occurrences and compares it with the group bull belongs to. After analyzing the sentence, the proposed algorithm determines a conflict and infers that bull represents certain characteristic of Neo and does not imply that Neo is actually a bull.
Fig. 3.19: Semantic Analysis of Direct Relationships

Fig. 3.21: Named Entity Recognition for Unknown Entities
Fig. 3.20: An owl Representation for the Semantic Network
Fig. 3.22: Named Entity Recognition for Known Entities based on Context

Fig. 3.23: Initial State while Developing the Hierarchy
Fig. 3.24: Intermediate State while Developing the Hierarchy

Fig. 3.25: Intermediate State while Developing the Hierarchy Contd

Fig. 3.26: Developed Hierarchy
3.2.2 A LIGHT-WEIGHT ONTOLOGY BASED SEARCH ENGINE DESIGN

The content in a web page is unstructured. A browser can recognize the type of content in a web page using the meta-data provided but has no means of understanding it. A sentence, such as “Karen is a cow”, is just another piece of text it has to render, but actually, it might be expressing Karen’s behavior or simply implying that Karen is a cow. A browser has no means to infer such interpretations by just reading the plain unstructured text available in a web page. An ontological representation of the web page is a possible solution to this dilemma. Ontologies can act as computational models and provide us with certain type of automated reasoning. They will enable semantic analysis and processing of the content in the web page. The following Fig. 3.27 will show the results of Google Search Engine for the keyword “Neo”.

The currently available functional search engines provide the best available web results based on various ranking algorithms but do not provide us with meaningful insight into the content of the web page. The information available with each web link is not sufficient to help the user select the most apt web page. To obtain detailed information, it creates the user tendency of blindly going to

Fig. 3.28: Enhanced Architecture of the Ontology-based Search Engine
Fig. 3.27: The Results Obtained when Querying “Neo” on the Web using the Google Search Engine
Wikipedia without even checking the other web results provided by the search engine. In a way, we are bound to various websites based on their reputation and neglect valuable information that might be available with other web pages.

Fig. 3.29: Search Results Obtained when Querying using Semantically Extracted Information as O–A–V Triplets

The user should be made aware of the contents of the webpages before he selects a link. This approach will enable the user to make a more informed choice and streamline the web surfing experience. To fill these gaps, the proposed architecture of the ontology-based search engine is given in Fig. 3.28.
Representing information with each web link in the form of O–A–V triplets provides the user with insight into the content on a web page. Because this information is extracted semantically using ontologies, it also allows the user to understand the type of content available on the web and is shown in Fig. 3.29, 3.30 and 3.31.