Vijayarajan *et al. Hum. Cent. Comput. Inf. Sci.* (2016) 6:18
DOI 10.1186/s13673-016-0074-1

Human-centric Computing and Information Sciences

RESEARCH Open Access

A generic framework for ontology-based information retrieval and image retrieval in web data

V. Vijayarajan[1*], M. Dinakaran[2], Priyam Tejaswin[1] and Mayank Lohani[1]

*Correspondence:
virtual.viji@gmail.com
[1] School of Computing Science and Engineering, VIT University, Vellore 632014, Tamilnadu, India
Full list of author information is available at the end of the article

Abstract

In the internet era, search engines play a vital role in information retrieval from web pages. Search engines arrange the retrieved results using various ranking algorithms. Additionally, retrieval is based on statistical searching techniques or content-based information extraction methods. It is still difficult for the user to understand the abstract details of every web page unless the user opens it separately to view the web content. This key point provided the motivation to propose and display an ontology-based object-attribute-value (O-A-V) information extraction system as a web model that acts as a user dictionary to refine the search keywords in the query for subsequent attempts. This first model is evaluated using various natural language processing (NLP) queries given as English sentences. Additionally, image search engines, such as Google Images, use content-based image information extraction and retrieval of web pages against the user query. To minimize the semantic gap between the image retrieval results and the expected user results, the domain ontology is built using image descriptions. The second proposed model initially examines natural language user queries using an NLP parser algorithm that will identify the subject-predicate-object (S-P-O) for the query. S-P-O extraction is an extended idea from the ontology-based O-A-V web model. Using this S-P-O extraction and considering the complex nature of writing SPARQL protocol and RDF query language (SPARQL) from the user point of view, the SPARQL auto query generation module is proposed, and it will auto generate the SPARQL query. Then, the query is deployed on the ontology, and images are retrieved based on the auto-generated SPARQL query. With the proposed methodology above, this paper seeks answers to following two questions. First, how to combine the use of domain ontology and semantics to improve information retrieval and user experience? Second, does this new unified framework improve the standard information retrieval systems? To answer these questions, a document retrieval system and an image retrieval system were built to test our proposed framework. The web document retrieval was tested against three key-words/bag-of-words models and a semantic ontology model. Image retrieval was tested on IAPR TC-12 benchmark dataset. The precision, recall and accuracy results were then compared against standard information retrieval systems using TREC_EVAL. The results indicated improvements over the standard systems. A controlled experiment was performed by test subjects querying the retrieval system in the absence and presence of our proposed framework. The queries were measured using two metrics, time and click-count. Comparisons were made

on the retrieval performed with and without our proposed framework. The results were encouraging.

Keywords: Information retrieval, Ontology, Image retrieval, Natural language processing, SPARQL query

Background

The web is vast, but it is not intelligent enough to recognize the queries made by users and relate them to real or abstract entities in the world. It is a collection of unstructured documents and other resources, which are linked by hyperlinks and URLs. The 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 is all about common formats for incorporation and amalgamation of data drawn from miscellaneous sources and how the data relates to real world objects. It provides a common structure that allows data to be shared and reused across applications, enterprise and community boundaries [1]. The linked data [2, 3], refers to a method of publishing structured data so that it can be interlinked and made more useful.

Rather than using Web Technologies to serve web pages for human renders, it uses these web technologies to share information in a way that can be read automatically by computers, enabling data from different sources to be connected and queried [4]. The reasoning is the capacity for consciously making sense of things, applying logic, establishing and verifying facts, and changing or justifying practices, institutions, and beliefs based on new or existing information [5]. Using the intelligent Semantic Web, the web agents will be able to identify the content on the web and draw inferences based on the relationships between various web resources. Ontology is the metaphysical study of the nature of being, becoming, presence, or truth, as well as the basic groups of being and their relations. Any entity, whether real or abstract, has firm characteristics, which relate to firm entities in the real world and interactions among them.

Ontologies address the existence of entities, organize them into groups based on their similarity, develop hierarchies and study the relationships among them, which allows for the drawing of inferences based on their classification, studying how they interact with other distinct entities in the real world and, finally, helps in the development of domain ontologies. In the Semantic Web, 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 [6, 7].

Additionally, in the Semantic Web, the ontologies act as the building blocks for the infrastructure of the semantic web. They transform the existing web data into the web of knowledge, share the knowledge among various web applications, and enable intelligent web services. Knowledge representation is the application of logic and ontology to build computable models for various domains [8]. The pillars of the Semantic Web are knowledge representation and reasoning. There is no absolute knowledge representation methodology, and it depends only on the type of application and how it uses the acquired knowledge. The WordNet [9, 10] is a large lexical database of the English Language. It groups closely related words into unordered set called Synsets, which are interlinked via conceptual-semantic and lexical relations. It is considered an upper ontology

by some, but it is not strictly an ontology. However, it has been used as a linguistic tool for learning domain ontologies. The Resource Description Framework (RDF) [11] is an official W3C Recommendation for Semantic Web data models. In this way, the RDF and RDFs can be used to design an efficient framework to describe various resources on the web so 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. The RDF schema encodes ontologies, providing the semantics, vocabulary and various relationships in the domain. A semantic RDF alignment based information retrieval system is found in [12]. Due to the growth of multimedia technologies, hardware improvements and low-cost storage devices, the number of digital images on the web is increasing dramatically.

For the past two decades, a considerable amount of research has been performed in Image Retrieval (IR). In traditional text-based image annotations, the images are manually annotated by humans, and the annotations are used as an index for image retrieval [13, 14]. The second well-known approach in image retrieval is Content-Based Image Retrieval (CBIR) where the low-level image features, such as color, texture and shape are used as an index for image retrieval [15–17]. The third approach is Automatic Image Annotation (AIA) where the system learns semantic information from image concepts and uses that knowledge to label a new image [18, 19]. There are some benchmark image datasets available, such as IAPR TC-12, that have proper image content descriptions. ImageCLEF [20] can be used for ad-hoc image retrieval tasks via text and/or content-based image retrieval of CLEF from 2006 onwards [21]. To query results from ontology, SPARQL [22, 23] is used as the query language using Jena Fuseki [24], which is a server that stores all RDFs. However, the image retrieval result is accurate if the annotations are perfect.

Related work

Google's knowledge graph [25, 26] is a knowledge base used by Google to enhance its search engine's search results with semantic-search information gathered from a wide variety of sources. There are some challenges to be considered while constructing a knowledge graph discussed in [27]. It works at the outer level, drawing semantic relationships among various resources, and provides us with the best web results. In contrast to this behavior, our web model works at the inner level, drawing semantic relationships inside each web document and providing meaningful insight to the content available with each web link further improving the user's web search experience. MagPie [28] allows for semantic interpretation of web pages. It comes as a plugin to web browsers. It decides the user domain of search by asking him to select an ontology and concepts to confine his search. Based on these parameters, it relates web pages and highlights the various concepts on the web pages. It also allows the user service to determine the type of content the user searches for and develops a profile to enhance the search results. The DBpedia [29] is a project aimed at extracting structured content from the information created as part of the Wikipedia project. It allows users to query relationships and properties associated with Wikipedia resources.

A BioSemantic framework [30] speeds up the integration of relational databases. It generates and annotates RDF views that enable the automatic generation of SPARQL queries. However, they are not using natural language queries for SPARQL query generation. The thesis [31], generating SPARQL queries automatically from keywords applied in Linked Data Web, does not explain the extension of using it for image descriptions. AquaLog [32] is a portable question-answering system, which receives queries in natural language and an ontology as the inputs and retrieves answers from the available semantic markup. There are some annotation-based image retrieval systems using ontology, but they do not use SPARQL queries. The feature based reranking algorithm for image similarity prediction using query-context bag-of-object retrieval technique is discussed in [33].

Proposed architectures

Framework for an ontology-based web search engine

The arrangement of this framework consists of an object-attribute-value extraction procedure from a natural English language query and a lightweight ontology-based search engine design [34]. Because most of the information available on the web is in natural language and not machine understandable, there is no way to understand the data and draw out semantic inferences. Ontologies can be used to model the information so that it can be easily interpreted by machines.

Sentence structure A typical clause consists of a subject and a predicate, where the predicate is typically a verb phrase and any objects or other modifiers, as shown in Fig. 1. The parse tree for a sample sentence statement clause is shown in Fig. 2.

Object-attribute-value extraction procedure

When passing the text through the proposed model shown in Fig. 3, 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

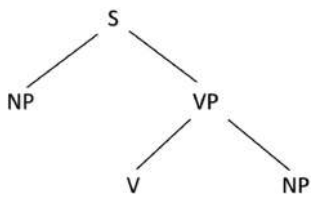

Fig. 1 Parse tree developed from simple clause (S–sentence, NP—noun phrase, VP—verb phrase, V—verb)

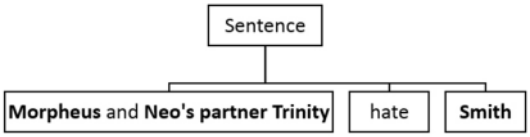

Fig. 2 Breakdown of a clause based on the parse tree [Morpheus—simple subject, Trinity—compound subject, hate—verb, Smith—object (Inferred information: Neo's partner Trinity; Trinity, Morpheus hates Smith)]

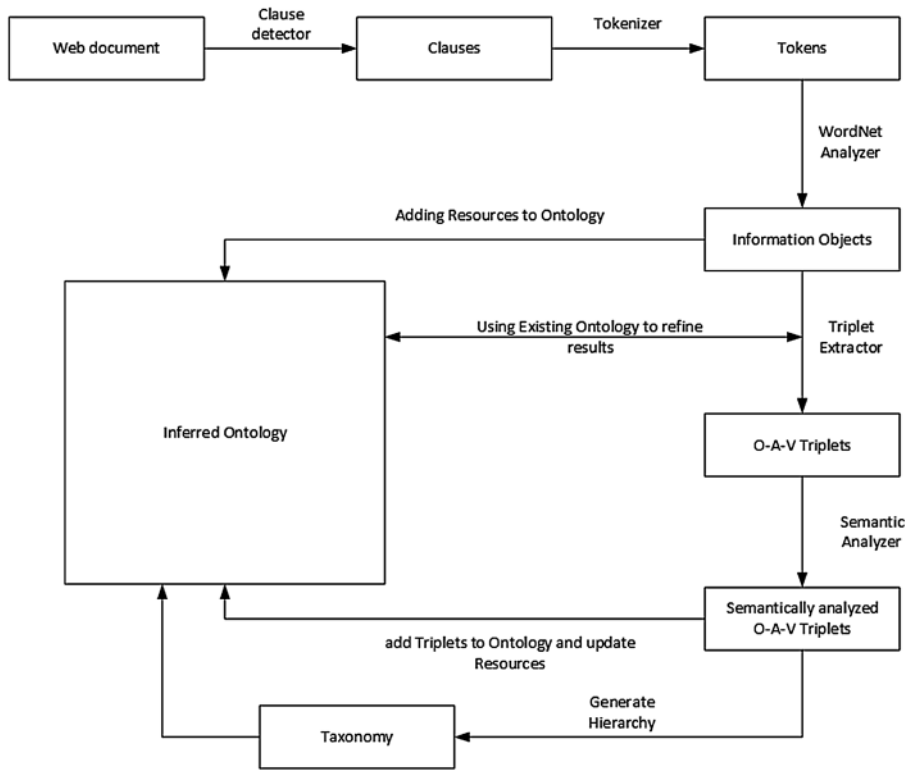

Fig. 3 Proposed architecture for ontology based information extraction

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.

Algorithm design

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 occuring 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:     develope 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

```
while not end of NP do
    if next token ∉ N then
        create the current token as individual in ontology
    else
        create O-A-V triplet between current token and next token with V as a combination of both
        update current token with value of V
        set class of V with the value of class of A
    end if
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. 4.

Algorithm 3 Semantic Analysis of Direct Relations.

```
if O ∈ Ontology and O ∉ class of V then
    V represents a property or a characteristic of O rather than him
else
    set class of O with the value of class of V of A
end if
```

Algorithm 4 Developing a Taxonomy.

```
while no Individual left do
    extract hypernyms for each individuals
    arrange the individuals in order of appearance in their hierarchies
    find common ancestors between the individuals up their hierarchies
    add individuals to a common class having common ancestors
    remove these individuals and add the ancestor as another individual in the given set
end while
```

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. 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 Figs. 5, 6, 7 and 8.

For parsing the sentences taken in Fig. 9 using the proposed algorithm, it generates the semantic networks shown in Fig. 10. The semantic analysis of direct relationships is shown in Fig. 4.

The Web Ontology Language (OWL) representation for the above semantic network is shown in Fig. 11. The entity recognition for unknown entities and known entities during the semantic analysis are shown in Figs. 12 and 13.

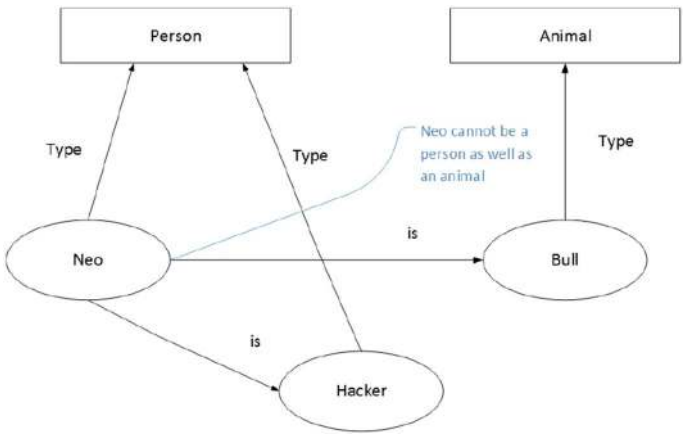

Fig. 4 Semantic analysis of direct relationships

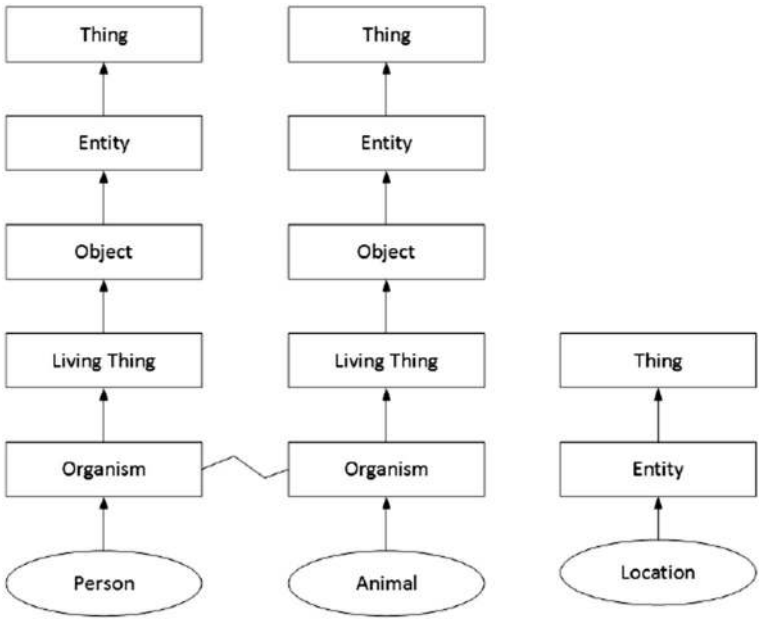

Fig. 5 Initial state while developing the hierarchy

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.

A lightweight 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

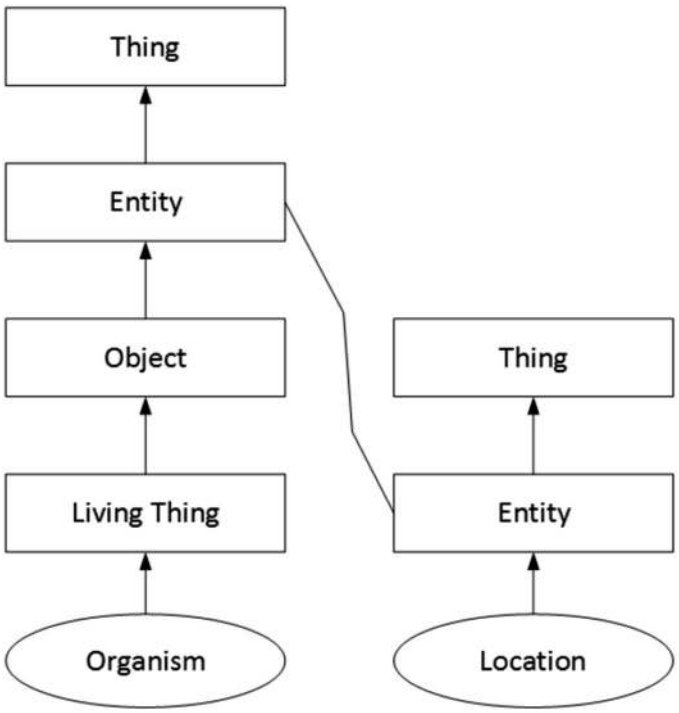

Fig. 6 Intermediate state while developing the hierarchy (Micro-level)

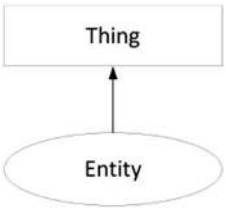

Fig. 7 Intermediate state while developing the hierarchy (Macro-level)

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. 14 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 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. The user should be made aware of the contents of the webpages before

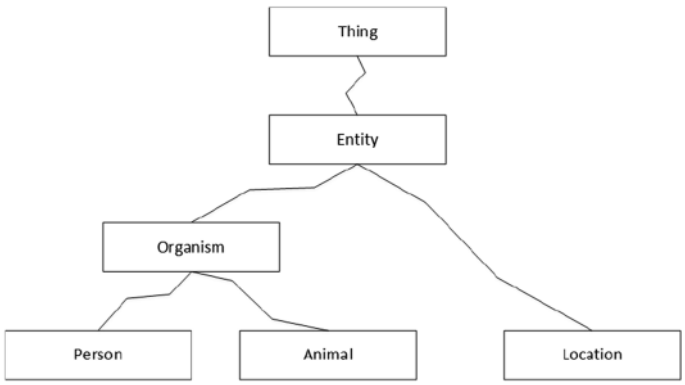

Fig. 8 Developed hierarchy

```
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. 9 A sample query for analysis

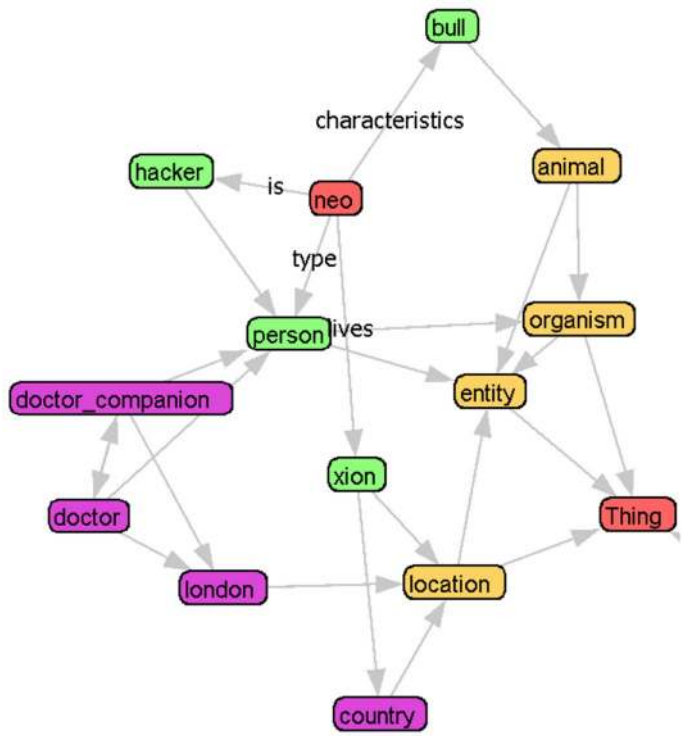

Fig. 10 Semantic network developed using the sample text

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. 15.

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix base-ns: <file:/Ontology.rdf/> .
@prefix property-ns: <file:/Ontology.rdf/property-ns#> .
@prefix class-ns: <file:/Ontology.rdf/class-ns#> .
@prefix resource-ns: <file:/Ontology.rdf/resource-ns#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

resource-ns:neo a              class-ns:person ;
     property-ns:characteristics resource-ns:bull ;
     property-ns:is            resource-ns:hacker ;
     property-ns:lives         resource-ns:xion .

class-ns:organism a    owl:Class ;
     rdfs:subClassOf  class-ns:entity , owl:Thing .

class-ns:animal a      owl:Class ;
     rdfs:subClassOf  class-ns:entity , class-ns:organism .

resource-ns:london  a  class-ns:location .

resource-ns:doctor_companion
     a                 class-ns:person ;
     property-ns:companion-of  resource-ns:doctor ;
     property-ns:live        resource-ns:london .

resource-ns:xion  a    class-ns:location ;
     property-ns:is  resource-ns:country .

class-ns:entity  a     owl:Class ;
     rdfs:subClassOf  owl:Thing .

resource-ns:bull  a  class-ns:animal .

class-ns:person  a     owl:Class ;
     rdfs:subClassOf  class-ns:entity , class-ns:organism .

resource-ns:country  a  class-ns:location .

class-ns:location  a    owl:Class ;
     rdfs:subClassOf  class-ns:entity , owl:Thing .

resource-ns:hacker  a  class-ns:person .

resource-ns:doctor  a       class-ns:person ;
     property-ns:companion  resource-ns:doctor_companion ;
     property-ns:live      resource-ns:london .

class-ns:null  a       owl:Class ;
```

Fig. 11 An OWL representation for the above semantic network

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 Figs. 16, 17 and 18.

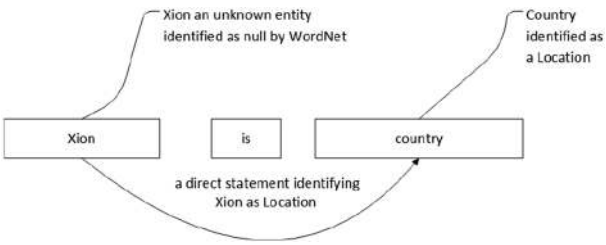

Fig. 12 Named entity recognition for unknown entities

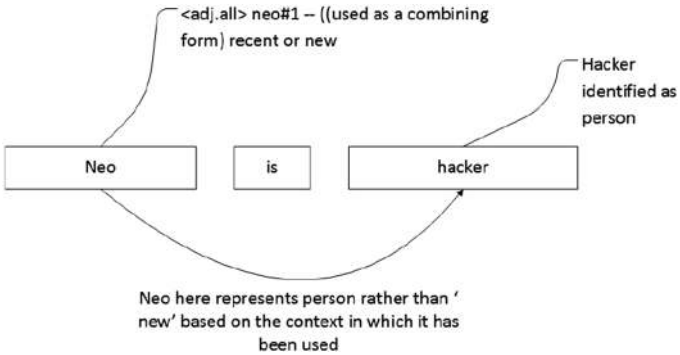

Fig. 13 Named entity recognition for known entities based on context

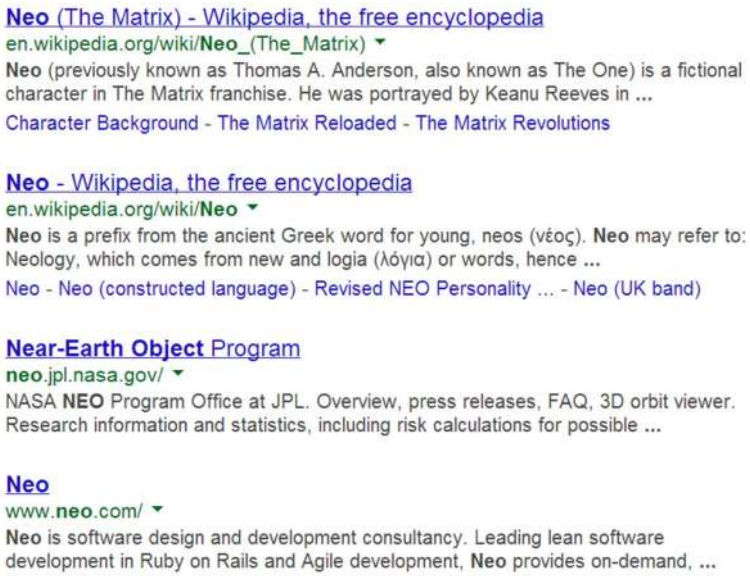

Fig. 14 The results obtained when querying "Neo" on the web using the Google search engine

Proposed framework for ontology-based image retrieval

The arrangement of this framework is shown in Fig. 19 and consists of domain ontology development for the image contents and creation of an RDF for the image descriptions, subject-predicate-object extraction based on [34] and from the natural language queries

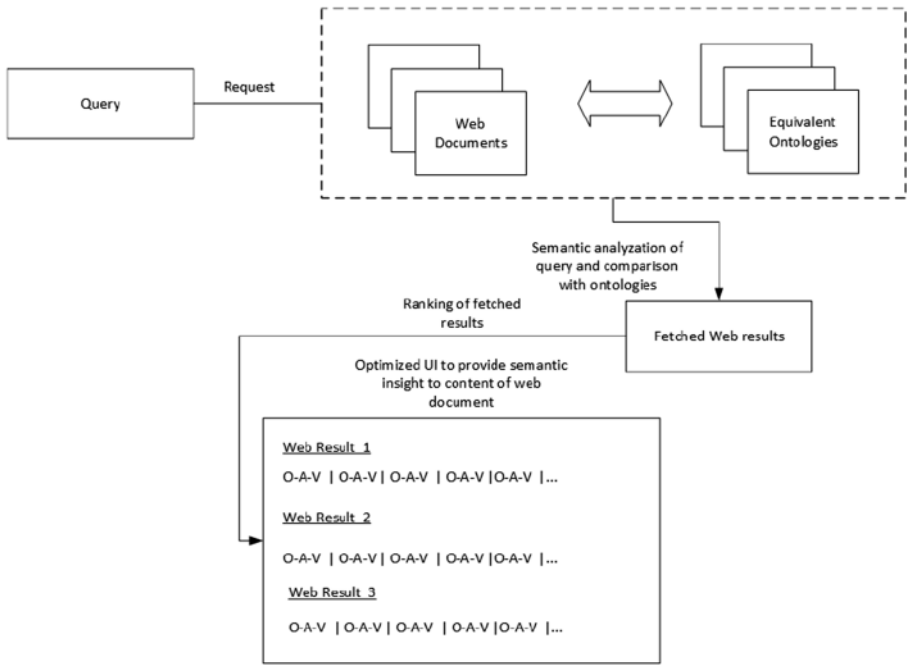

Fig. 15 Enhanced architecture of the ontology-based search engine

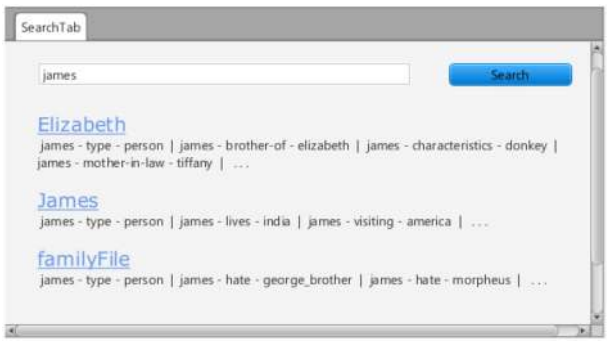

Fig. 16 Search results obtained when querying using semantically extracted information as O-A-V triplets

given by the user and auto generation of SPARQL queries on the ontology to obtain ontology-based image retrieval results.

An ontology refers to a description of a conceptualization. It describes a domain in a formal way. With the help of nearby textual information, the web image retrieval is accomplished. There are text-based image retrieval engines in practice, such as Yahoo, Bing and Google. They use text features, such as file names, as indices for searching for images on the web. At the next level, they search for textual information surrounding the image in the web page. The content-based image retrieval works with low-level image features, such as color, texture and shape.

However, due to the limitations of the current image processing algorithms, there still exists a gap called the "semantic gap", which occurs due to the lack of understanding of image semantics using image processing algorithms when they try to map it with human

Fig. 17 Content inside the selected web content

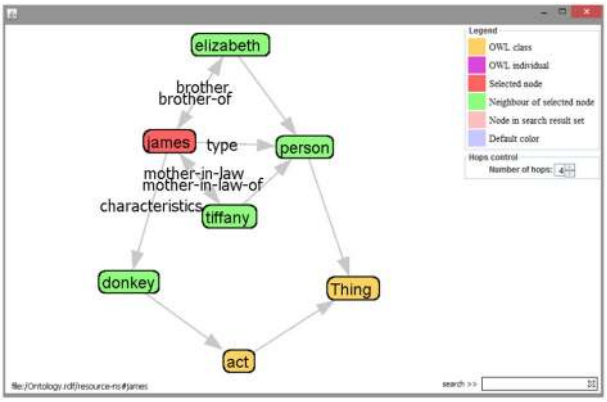

Fig. 18 A graphical display showing the results of the search query for an individual document

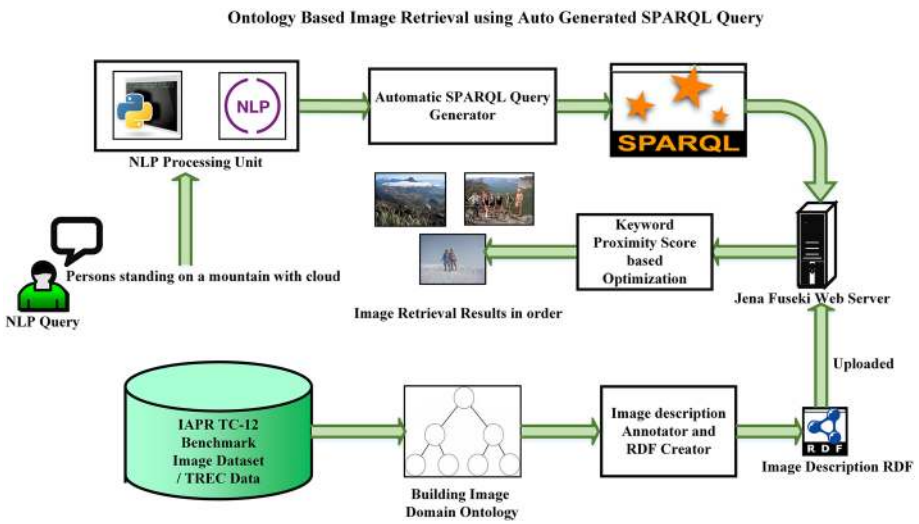

Fig. 19 Ontology-based image retrieval framework

understanding of the images. Image retrieval search engines are still evolving. The low-level descriptors of these engines are far from semantic notions. The other types of systems only rely on annotations. Therefore, there is a need to define an intermediate approach for image analysis by building a domain ontology for image categories. Some systems may define a specific domain with the help of domain experts by identifying vocabularies used to describe objects of interest. For experimental purposes, the image data set from the IAPR TC-12 Benchmark is chosen from ImageCLEF 2006, which contains detailed image descriptions. The image domain ontology is developed as in Fig. 20 for the data set taken with all possible class concepts using Protege [35]. The RDF output is shown in Fig. 21.

Once the ontology is created successfully, it can be stored as an OWL file. The images are annotated with descriptions given along with the data set. RDFs of all individual images are embedded and converted to make a single RDF file which is uploaded to Jena Fuseki Server. Each RDF attribute would be stored as a tuple in the server space and hence the considerable amount of tuples should be generated. These tuples should return values if a proper SPARQL query is fired through Jena engine.

The retrieval of images in this framework has to undergo another crucial process of evaluating the user query, which is given in natural language.

Natural language processing

The user query given in the english language is passed to the NLP processor, which performs operations similar to the O-A-V extraction in the web model. The first step is to perform part-of-speech (POS) tagging. Therefore, the sentence is passed through a POS tagging function within the NLP processing unit. This unit returns a list of tagged words with their parts of speech as tuples. The subject in an english sentence will act as the object in the O-A-V triplet. To identify the subject from the sentence, we need to identify a noun phrase consisting of nouns and adjectives, which define the various properties of the noun. Similarly, the predicate of an english sentence acts as the attribute in the

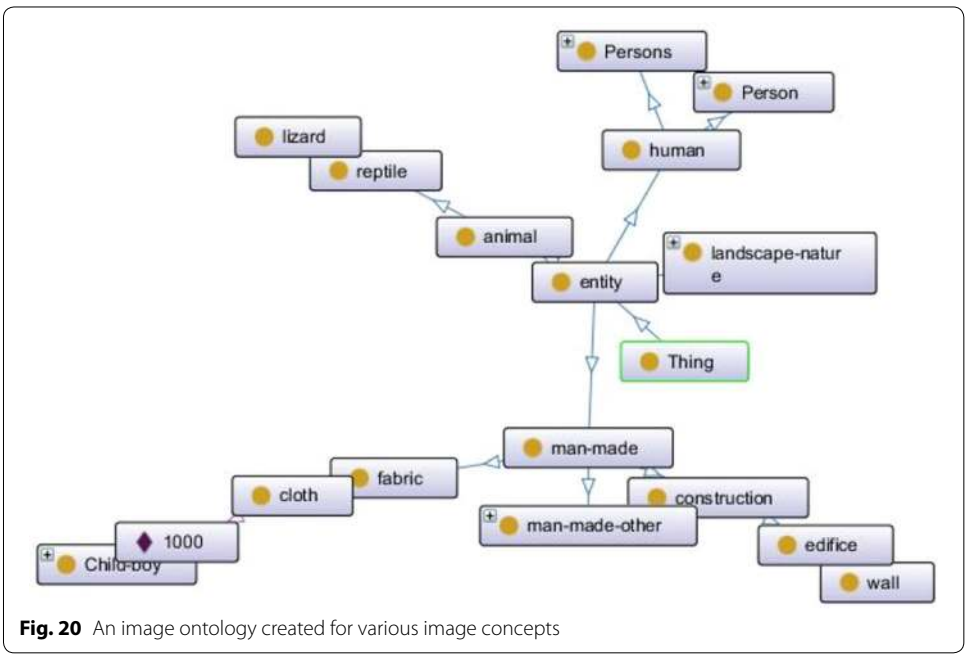

Fig. 20 An image ontology created for various image concepts

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
    xmlns:s0="http://www.w3.org/2000/PhotoRDF/dc-1-0#"
    xmlns:s1="http://sophia.inria.fr/~enerbonn/rdfpiclang#">
    <?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
    xmlns:s0="http://www.w3.org/2000/PhotoRDF/dc-1-0#"
    xmlns:s1="http://sophia.inria.fr/~enerbonn/rdfpiclang#">
  <rdf:Description rdf:about="">
    <s0:subject>Portrait, Group-Portrait</s0:subject>
    <s0:type>image</s0:type>
    <s0:format>image/jpeg</s0:format>
    <s1:xmllang>en</s1:xmllang>
    <s0:date>not available</s0:date>
    <s0:description>five women and two men are standing on a lookout with
    a wooded canyon and steep,grey cliffs in the background</s0:description>
    <s0:creator>Self</s0:creator>
    <s0:title>Photo Snap</s0:title>
  </rdf:Description>
</rdf:RDF>
```

Fig. 21 An image ontology created for various image concepts

O-A-V triplet. To identify the predicate, we need to identify the verb phrase in the sentence. Every grammatically correct English sentence contains a subject and a predicate. For the purpose of this model, we extract only the adjectives, nouns and the verbs from the tagged sentence by eliminating the stop words from the query. Once the desired parts of speech have been extracted, the tagged sentence is parsed, separating the SUBJECTs, PREDICATEs and OBJECTs from the sentence. Regular expressions are used to group all consecutive nouns and adjectives as a noun phrase. The result is stored as a tree object and is then traversed and parsed to separate the subject, predicate and object. The result of this separation is shown in Fig. 22.

These three groups (subjects, predicates, objects) are then used to search for the appropriate images in the database. A number of operations and transformations are applied to the natural language query for extracting keywords. Part of speech tagging is performed, followed by splitting the query into sentences and further into word tokens. Noun, adjective and verb tokens are lemmatized and stemmed to their appropriate roots.

Auto generation of the SPARQL query

Normally in a SPARQL query, the FILTER operator is used to screen the desired output when querying from the database. For example, if a user enters a query and n keywords

```
enter a query: tourists walking on a sandy beach
subject(s):
('tourists', 'NNS')
predicates(s)
('walking', 'VBG')
object(s)
('sandy', 'JJ')
('beach', 'NN')
```

Fig. 22 O-A-V (subject-predicate-object) extraction

have been picked, then the best possible retrieval results will be the images whose descriptions contain all n query words. However, there may arise situations where not all keywords are present in the description. Therefore, this query will give no result. However, there may exist subsets of the n keywords, which are present in descriptions of the images. It is always safe to assume that an image with a description containing more query keywords is most likely to give a better retrieval result. Still, it is very difficult to determine which keywords to eliminate when trying the next query. Therefore, to tackle this problem, all combinations of the n keywords are queried. For n keywords, $2^n - 1$ subsets can be formed.

The combinations are considered for querying in decreasing order of the number of elements (keywords) in the set. The UNION operator is used to ensure that the results of all queries are considered. The DISTINCT operator eliminates duplicate results. The retrieved images will be in decreasing order of likeliness, similar to any search engine's page ranking results with the top queries having higher chances of being the desired results. First, a function builds the phrase dictionary containing the Subject-Predicate-Objects. The function then generates queries for all of the words present in the dictionary shown in Algorithm 5.

Algorithm 5 SPARQL query auto generation algorithm.

```
1: while not end of NLP sentence do
2:     tokenize the sentence and perform POS tagging
3:     analyze the tagged sentence to extract O-A-V triplet
4:     if next token ∈ S or P or O then
5:         generate all possible combinations of the keywords
6:     else
7:         ignore them
8:     end if
9: end while
10: while not end of SPARQL query generation do
11:     use DISTINCT operator to omit redundant results
12:     while not end of possible combinations of subsets of keywords do
13:         construct SPARQL sub-queries by taking subset combinations in decreasing order of number
    of elements
14:         search for appropriate images using the FILTER keyword;
15:         account for all sub-query results using the UNION operator
16:     end while
17: end while
```

An effective search query is one where the maximum number of keywords match the description of multiple images. The higher this intersection of keywords to description, the higher the chance of that particular image being the most appropriate image. It is logical to search for all keywords in the same description as the first query. The descriptions of images are searched using the FILTER operator. Filtering a description with all keywords of the search query is most likely to produce the best results. However, it is possible that the query keywords are not a complete subset of the description of the image. This query will give no result, even though some keywords match. Then, the next step would be to remove certain keywords and re-query the database, which is where the problem arises. It is impossible to identify keywords whose elimination will produce results. Therefore, the program creates all possible combinations of the keywords present in the phrases dictionary. The result is stored in a list that contains all possible combination shown in Fig. 23 of the phrase words for the query shown in Fig. 22.

If 'n' keywords have been selected in the phrase dictionary, then a total of $2^n - 1$ combinations are stored in the list AC (all combinations) as tuples, where every tuple

```
(('walking', 'VBG'), ('sandy', 'JJ'), ('beach', 'NN'), ('tourists', 'NNS'))
(('sandy', 'JJ'), ('beach', 'NN'), ('tourists', 'NNS'))
(('walking', 'VBG'), ('beach', 'NN'), ('tourists', 'NNS'))
(('walking', 'VBG'), ('sandy', 'JJ'), ('tourists', 'NNS'))
(('walking', 'VBG'), ('sandy', 'JJ'), ('beach', 'NN'))
(('beach', 'NN'), ('tourists', 'NNS'))
(('sandy', 'JJ'), ('tourists', 'NNS'))
(('sandy', 'JJ'), ('beach', 'NN'))
(('walking', 'VBG'), ('tourists', 'NNS'))
(('walking', 'VBG'), ('beach', 'NN'))
(('walking', 'VBG'), ('sandy', 'JJ'))
(('tourists', 'NNS'),)
(('beach', 'NN'),)
(('sandy', 'JJ'),)
(('walking', 'VBG'),)
```

Fig. 23 Unique combinations of the O-A-V triplet

represents one of the $2^n - 1$ subsets, consisting of the word and the POS as a tuple.The combination() function returns a list of all combinations of subsets in increasing order of number of keywords. Every subset contains tuples of words, where every tuple contains the keyword and the part of speech. For more effective results, this list is reversed before generating the query. This step ensures that the program will recursively consider all combinations of keywords in decreasing order of number of keywords while generating the query. Once the list of all combinations is generated and reversed, the elements of the list are considered one by one to generate the query. An element of the list is one subset out of the $2^n - 1$ subsets.

This subset represents a single SPARQL sub-query. All words inside the subset are the FILTER operator variable, which are to be searched for in the description of the images. The complete query is generated as follows: The query is initialized as just the prefix values in the beginning of the program. Every time the program runs, it generates a query string containing the prefix statements as the 'SELECT DISTINCT * WHERE' statement. Because every element of the list is a sub-query, the function adds 'select * where ?identifier s0:description ?value.' to the existing query. Every element in the list is a subset containing different combinations of the keywords. Once it is an element of the list, it considers every tuple in the subset to filter the description. The 'FILTER (REGEX(STR(?value),"' is then added to the query, which is followed by the keyword present in the tuple.

Before the keyword can be used to filter the description, it must be lemmatized. This process of lemmatization will help to consider all words, including different conjugations, infinitives, plurals, etc. Every filter expression is closed with '", "i"))'. Before moving on to the next subset, every sub query ends with '}}UNION'. If a subset of AC contains m keywords, then m filter options are added to the query. The UNION operator is concatenated before moving on to the next tuple in the list. This procedure generates $2^n - 1$ sub-queries joined by $2^n - 1$ UNION operators, for (n) keywords.

However, for $2^n - 1$ sub-queries, only $2^n - 2$ UNION operators are required. Therefore, before returning the query, the function removes the last 'UNION' and adds a ??. The UNION operator ensures that all subsets are being considered while querying the database. Before the programs exits, the SUBJECT, PREDICATE and OBJECT phrases are displayed followed by the resultant query as shown in Fig. 22. The auto generated SPARQL query is shown in Fig. 24.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/TR/1999/PR-rdf-schema-19990303#>
PREFIX s0: <http://www.w3.org/2000/PhotoRDF/dc-1-0#>
PREFIX s1: <http://dummyURL.com/PhotoOntilogy#>
SELECT DISTINCT * WHERE{
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "sandy", "i"))
        FILTER (REGEX(STR(?value), "beach", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "sandy", "i"))
        FILTER (REGEX(STR(?value), "beach", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "beach", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "sandy", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
     {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "sandy", "i"))
        FILTER (REGEX(STR(?value), "beach", "i"))
        }}  UNION
     {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "beach", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
     {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "sandy", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
     {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "sandy", "i"))
        FILTER (REGEX(STR(?value), "beach", "i"))
        }}  UNION
     {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
     {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "beach", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        FILTER (REGEX(STR(?value), "sandy", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "tourist", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "beach", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "sandy", "i"))
        }}  UNION
    {select * where { ?identifier s0:description ?value.
        FILTER (REGEX(STR(?value), "walking", "i"))
        }}
}
```

Fig. 24 The auto generated SPARQL query using genOAVquery() program

Ontology-based image retrieval using auto-generated SPARQL query

The above auto-generated query in Fig. 24 is fed to the Jena-FUSEKI Server. The results retrieved by the server are shown in Fig. 25. As explained previously, the results at the top are more likely to be more relevant than the results at the bottom. However, these results are not optimized.

Keyword proximity score based optimization

While generating the SPARQL query, all possible combinations of the O-A-V key words are included. The reasons for considering all possible combinations has been explained in the previous section. The search results from the Jena Fuseki-Server are optimized in a two step process. First, the results are in decreasing order of number of matching keywords. A description which has more common keywords with the query is more likely to be a return of a better picture. However, since many of the descriptions in the dataset are elaborate, it is possible that keywords are spread out over the description. Consider these query keywords and two of its results as an example:

query_KeyWords = ['man','looking', 'mountain']
⟨ *upload_base/1111.jpg* ⟩ ⟨ *A man standing on the roof and looking at the mountain* ⟩
⟨ *upload_base/2222.jpg* ⟩ ⟨ *A man is looking at his children playing near the lake across the mountain* ⟩

Both image descriptions contain all three keywords of the query. However, in 2222.jpg, the context of the query is lost since there is considerable distance between the words. But considering 1111.jpg where the the description is more meaningful and the keywords are closer. Given a set of sentences containing equal number of keywords, the word distance or keyword proximity can further optimize the search results. A higher keyword proximity score for a image suggests that its description contains phrases which are similar to the user's query.

The keyword proximity score is calculated by taking the absolute differences of the position of consecutively appearing keywords in the description of an image and then normalizing these total distances for a range 0–1 where zero represents low-proximity

```
-----------------------------------
| <upload-base/6633.jpg> | "tourists are walking on a sandy beach, with a brown cliff on the left;"
| <upload-base/6408.jpg> | "tourists are sitting on a sandy beach and having a picnic; a clean surf in the backgournd;"
| <upload-base/6821.jpg> | "a group of tourists is sitting at a sandy beach with the breaking waves of the sea in the background;"
| <upload-base/6967.jpg> | "two red and one white jeep with blue luggage bags are parked in the middle of a sandy desert; tourists are standing
                            in between and around the cars;"
| <upload-base/6497.jpg> | "a steep, sandy coast with some rocks in the foreground, and a set of breaking waves on a sandy beach in the
                            background;"
| <upload-base/6341.jpg> | "top view of a little bay with a sandy beach and black rocks with trees in the background;"
| <upload-base/6923.jpg> | "a group of people is sitting on a sandy beach and is having a picnic; a mountain and the sea in the background;"
| <upload-base/6991.jpg> | "a man is standing at the sandy beach and is watching the breaking waves of the sea;"
| <upload-base/6996.jpg> | "people at a sandy beach at the sea with a broken wave and a rocky coast in the background;"
| <upload-base/6465.jpg> | "tourists are walking on a path in the mountains; a wooded slope and rocky mountains in the background;"
| <upload-base/6931.jpg> | "four women are walking on a paved road with a sandy desert landscape in the background;"
| <upload-base/6925.jpg> | "tourists are standing in front of a brown wall, with the steep rocky coast and the sea in the background;"
| <upload-base/6500.jpg> | "Tourists are watching a condor that is flying over a deep canyon; there is some snow on the mountain on the side;"
| <upload-base/6858.jpg> | "five tourists with backpacks are standing in a courtyard of a yellow building;"
| <upload-base/6935.jpg> | "tourists are sitting in a pool with a grey and brown wall behind them;"
| <upload-base/6652.jpg> | "a woman is offering souvenirs to tourists; floor and houses in the background are made of reed;"
| <upload-base/6807.jpg> | "tourists are standing on a concrete square with yellow buildings in the background;"
| <upload-base/6848.jpg> | "a tourist group is squatting at a lookout with a city in the background;"
```

Fig. 25 Retrieved image results

and one represents high-proximity. For descriptions with only one matching keyword, the score is kept as minimum = 0.001.

Since the retrieval results are first sorted by number of occurrences of keywords, a description with a higher number of keyword matches suggests more similarity to the query. Hence while sorting with the keyword proximity score, image descriptions with equal number of keywords are sorted together and then displayed in descending order of their keyword proximity scores, while maintaining the previous structure of decreasing number of keyword occurrences. The keyword proximity score of a description with n keyword occurrences, cannot be compared to another description with m keyword occurrences where $n \neq m$. The keyword proximity score is comparative only amongst image descriptions with equal number of keyword occurrences. After optimization, the retrieval results in Fig. 25 are re-ranked and shown in Fig. 26 as follows. Every ranked result contains four items separated by a "|". The items in order are the image location, the image description, the keyword proximity score for the image, the number of query keywords present in the image description.

Experimental methodology and metrics

For testing and validating the effectiveness of our techniques, we built two retrieval systems (with and without optimization), each for text retrieval and image retrieval. The IAPR TC-12 benchmark dataset was used for testing the image retrieval. This benchmark collection contains 20,000 still natural images. Each image is associated with a text caption. The English language caption was taken for feature extraction for natural language processing. Similarly, the web document retrieval is experiment is done using TREC web test collections.

A user will query the retrieval systems, with and without the proposed algorithms. The two metrics chosen for evaluation are time and click-count. The click count is the number of clicks a user makes before the user arrives at the desired result (webpage or image). The time is the duration of one query session, till the user arrives at the desired result. The proposed system aims to define an optimized system as a search system which uses its feature extraction and ranking techniques. For a non-optimized system,

```
----------------------------------
| <upload-base/6633.jpg> | tourists are walking on a sandy beach, with a brown cliff on the left;" | 0.111 | 4
| <upload-base/6408.jpg> | "tourists are sitting on a sandy beach and having a picnic; a clean surf in the backgournd;" | 0.143 | 3
| <upload-base/6821.jpg> | "a group of tourists is sitting at a sandy beach with the breaking waves of the sea in the background;" | 0.143 | 3
| <upload-base/6923.jpg> | "a group of people is sitting on a sandy beach and is having a picnic; a mountain and the sea in the background;" |
0.500 | 2
| <upload-base/6996.jpg> | "people at a sandy beach at the sea with a broken wave and a rocky coast in the background;" | 0.500 | 2
| <upload-abse/6991.jpg> | "a man is standing at the sandy beach and is watching the braking waves of the sea" | 0.500 | 2
| <upload-base/6341.jpg> | "top view of a little bay with a sandy beach and black rocks with trees in the background;" | 0.500 | 2
| <upload-base/6465.jpg> | "tourists are walking on a path in the mountains; a wooded slope and rocky mountains in the background;" | 0.333 | 2
| <upload-base/6967.jpg> | "two red and one white jeep with blue luggage bags are parked in the middle of a sandy desert; tourists are standing
in between and around the cars;" | 0.333 | 2
| <upload-base/6931.jpg> | "four women are walking on a paved road with a sandy desert landscape in the background;" | 0.125 | 2
| <upload-base/6497.jpg> | "a steep, sandy coast with some rocks in the foreground, and a set of breaking waves on a sandy beach in the
background;" | 0.056 | 2
| <upload-base/6549.jpg> | "tourists are sitting on the roof of a train;" | 0.001 | 1
| <upload-base/6697.jpg> | "two tourists on a concrete square with a fountain, many people and a grey building with many arches in the
background;" | 0.056 | 2
| <upload-base/6688.jpg> | "a man is standing on a black rock with a sandy desert, a green valley and the sea in the background;" | 0.001 | 1
| <upload-base/6233.jpg> | "tourists wearing traditional red chagras are sitting on a slope of a hill and are drinking tea;" | 0.001 | 1
| <upload-base/6552.jpg> | "tourists wearing red life jackets are sitting on black tubes on a river with a wooded riverbank;" | 0.001 | 1
| <upload-base/6725.jpg> | "a woman with a yellow rain coat over her head is sitting on a grey wall; more tourists, ruins on terraces and dense
fog in the background;" | 0.001 | 1
| <upload-base/6532.jpg> | "tourists are relaxing in the middle of a white salt desert; a brown hill in the background on the left;" | 0.001 | 1
| <upload-base/6630.jpg> | "Tourists in a building with a red floor and arched windows behind bars" | 0.001 | 1
| <upload-base/6908.jpg> | "a group photo of tourists in front of a grey train" | 0.001 | 1
| <upload-base/6468.jpg> | "three tourists are sitting at a lookout, with a mountain landscape with ruins and steep, rocky mountains in the
background;" | 0.001 | 1
| <upload-base/6712.jpg> | "a cop is standing next to a grey car with luggage on the roof in the middle of a flat sandy desert" | 0.001 | 1
| <upload-base/6404.jpg> | "tourists are posing on and in front of a white bus on a grey gravel road in the middle of a flat, dry landscape with
snow covered mountains in the background" | 0.001 | 1
| <upload-base/6593.jpg> | "a large group shot of tourists and locals on the shore of a lake;" | 0.001 | 1
```

Fig. 26 Optimized image retrieval results

the user will take a longer time to get the desired search result. It will take more clicks to get to the desired result. Whereas, in an optimized system where the results are ranked and organized, it will be faster on both metrics (time and click-count).

For every query, these two metrics are tracked and recorded. The test subjects were faculty members and professors of the institute. For document retrieval, 100 test subjects tested the system with 10 queries per subject. For image retrieval, 57 test subjects tested the system with 10 queries each. The results of all tests were compiled in Tables 1 and 2. Queries were separated into simple and complex on the basis of mean click-count and mean time.

Also, these tables show the improvements in user mean click-time and mean click-count. The average time for the image retrieval system improved by 32.01 s and for the document retrieval system improved by 59.53 s with our techniques. The average click-count with optimization was 1.46 clicks less than the non-optimized system for image retrieval. For document retrieval, the click-count improved by 2.71 clicks.

For simple queries, the image retrieval results improved by 31.37 s on average, using the optimized system and document retrieval system improved by 57.72 s. The click-count for simple image retrieval queries improved by 1.44 clicks and by 2.63 clicks for document retrieval.

For complex queries, the image retrieval system improved by 30.63 s using our proposed techniques and the click-count improved by 1.48 clicks. The average time for complex document retrieval queries improved by 60.72 s and click-count improved by 2.76 clicks.

For document retrieval tested with simple queries vs click-count with and without optimization is shown in Fig. 27 and graph of complex queries vs click-count with and without optimization is shown in Fig. 28. For document retrieval tested with simple queries vs time with and without optimization is shown in Fig. 29 and graph of complex queries vs time with and without optimization is shown in Fig. 30.

Table 1 Mean time

	Image retrieval		Document retrieval	
	With optimization	Without optimization	With optimization	Without optimization
Query mean (s)	50.14	82.15	58.15	117.68
Simple queries mean (s)	48.08	79.45	56.3	114.02
Complex queries mean (s)	51.52	82.15	59.38	120.10

Table 2 Mean click-count

	Image retrieval		Document retrieval	
	With optimization	Without optimization	With optimization	Without optimization
Query Mean (clicks)	2.37	3.83	2.73	5.44
Simple queries mean (clicks)	2.27	3.71	2.65	5.28
Complex queries mean (clicks)	2.43	3.91	2.79	5.5

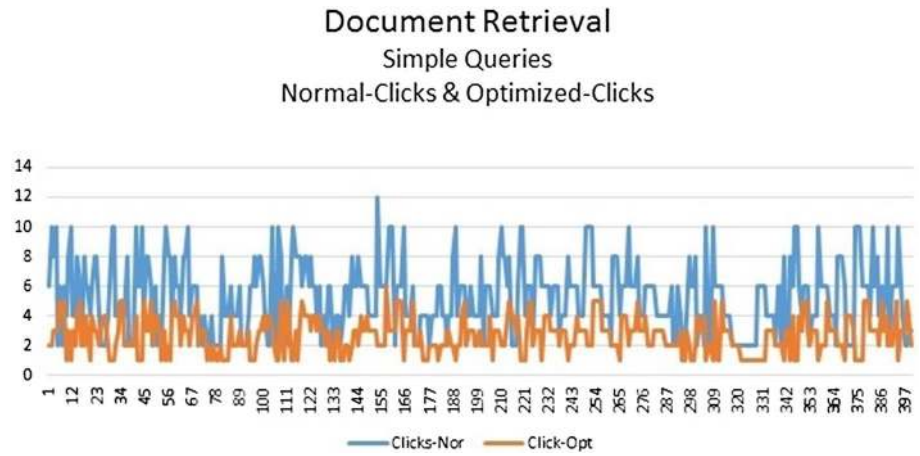

Fig. 27 Normal and optimized document retrieval system tested with simple queries vs clicks

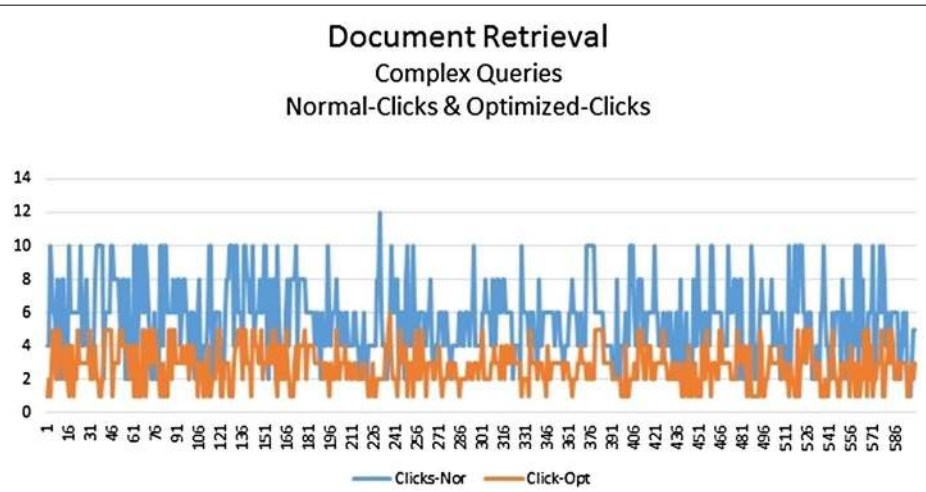

Fig. 28 Normal and optimized document retrieval system tested with complex queries vs clicks

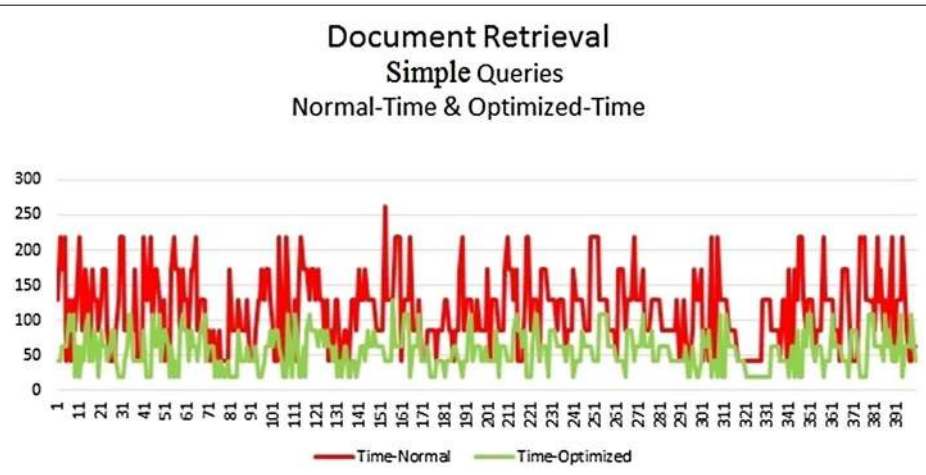

Fig. 29 Normal and optimized document retrieval system tested with simple queries vs time

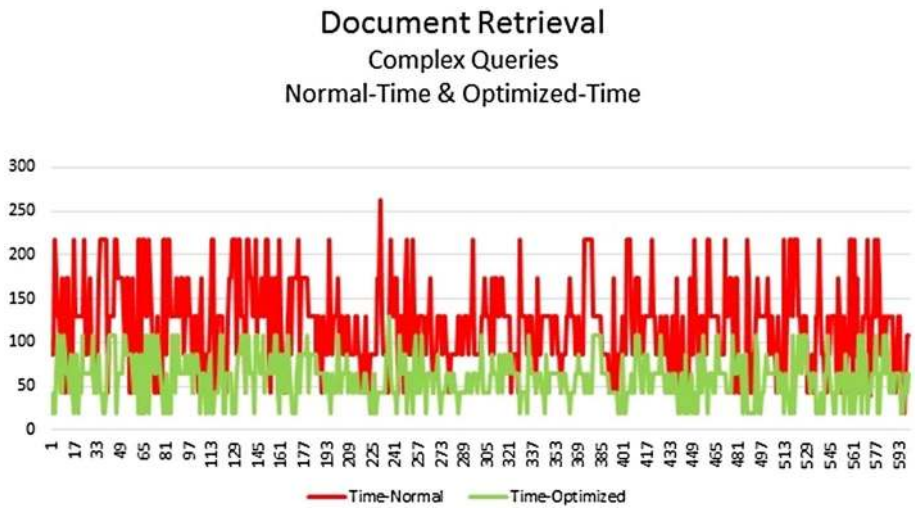

Fig. 30 Normal and optimized document retrieval system tested with complex queries vs time

For image retrieval tested with simple queries Vs click-count with and without optimization is shown in Fig. 31 and graph of complex queries Vs click-count with and without optimization is shown in Fig. 32. For image retrieval tested with simple queries Vs time with and without optimization is shown in Fig. 33 and graph of complex queries Vs time with and without optimization is shown in Fig. 34.

System evaluation and comparison

The evaluation of proposed system against standard information retrieval systems is as follows. Since, this is a new approach of combining ontology based information retrieval and NLP based information retrieval, standard benchmarks for evaluating such a combined technique were missing. Hence, we have compared our system against relevant evaluation benchmark consisting: the TREC WT10G Doc collection [36], queries selected from TREC9 and TREC2001 competitions with their respective judgements

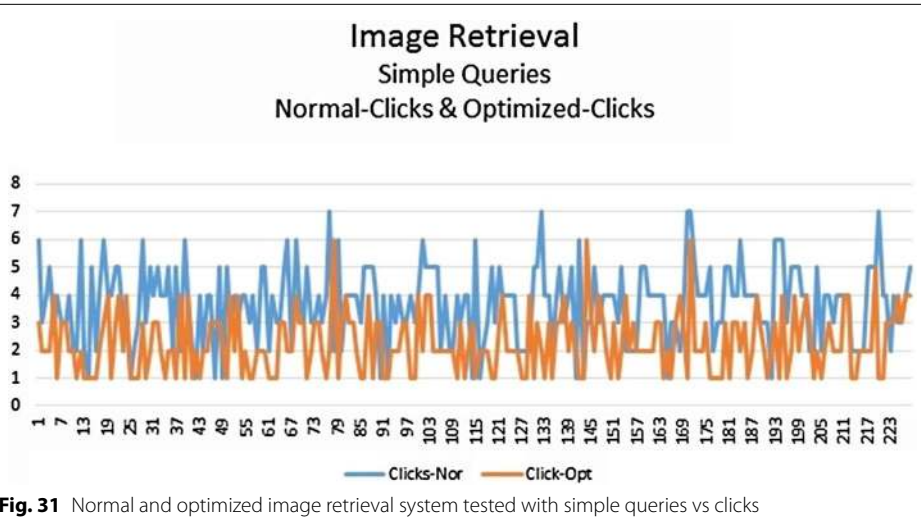

Fig. 31 Normal and optimized image retrieval system tested with simple queries vs clicks

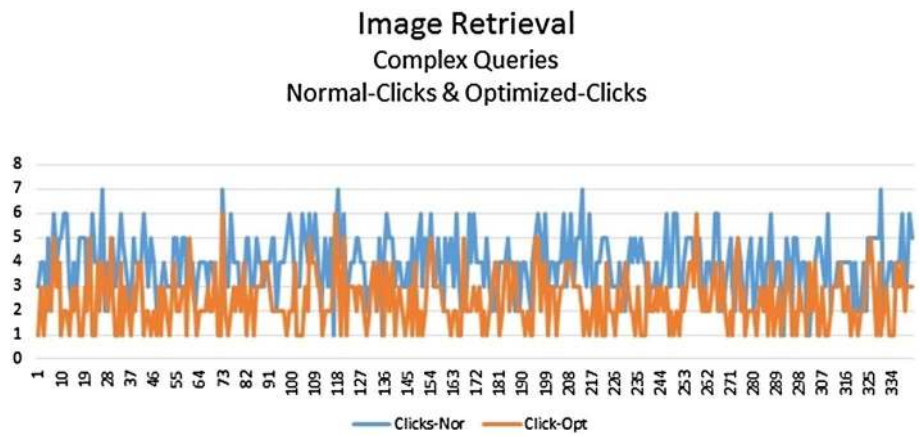

Fig. 32 Normal and optimized image retrieval system tested with complex queries vs clicks

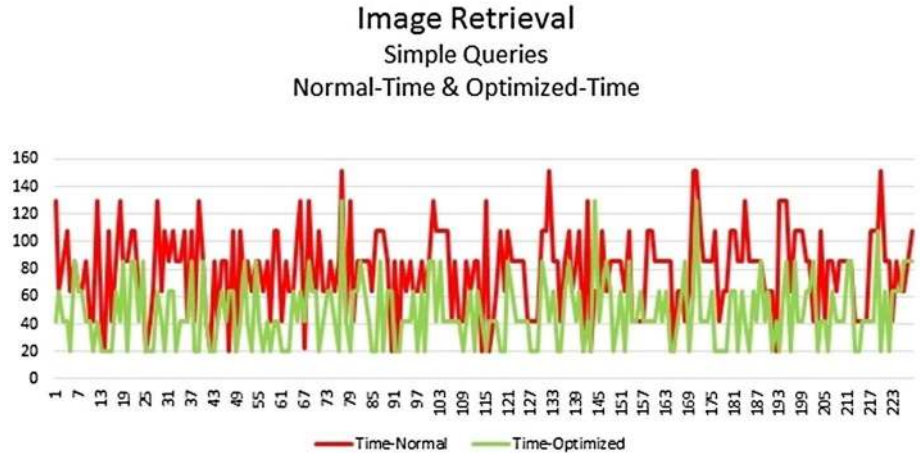

Fig. 33 Normal and optimized image retrieval system tested with simple queries vs time

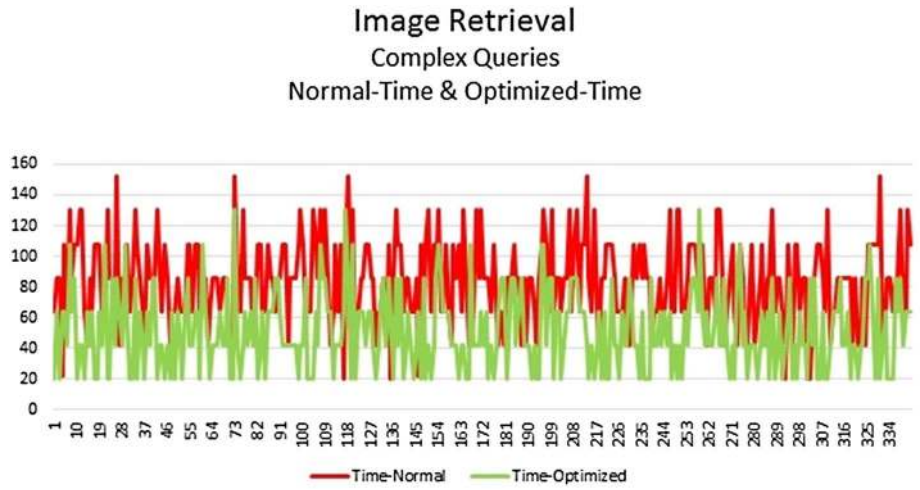

Fig. 34 Normal and optimized image retrieval system tested with complex queries vs time

[37], the semantically enhanced-ontology based information retrieval system [38] which has 40 public ontologies that covers TREC domain subsets.

The proposed generic framework compares the five different systems: three classical key-word based retrieval systems (TREC manual, TREC automatic, Lucence [39]), a semantic based retrieval system and the proposed generic framework. Tables 3 and 4 shows the results of TREC evaluation with topics and two information retrieval metrics

Table 3 Quality of results by MAP

Query Topic	Proposed generic framework	Semantic	Lucence	TREC automatic	TREC manual
451	0.41	0.42	0.29	*0.58*	0.54
452	*0.23*	0.04	0.03	0.2	0.33
454	*0.57*	0.26	0.26	0.56	0.48
457	0.04	*0.05*	0	0.12	0.22
465	*0.19*	0.13	0	0	0.61
467	0.11	0.1	*0.12*	0.09	0.21
476	*0.39*	0.13	0.28	0.41	0.52
484	0.16	*0.19*	0.12	0.05	0.36
489	*0.14*	0.09	0.11	0.06	0.41
491	*0.09*	0.08	0.08	0	0.7
494	0.49	0.41	0.22	*0.57*	0.57
504	0.19	0.13	0.08	*0.38*	0.64
508	*0.15*	*0.15*	0.03	0.06	0.1
511	0.12	0.07	0.15	*0.23*	0.15
512	*0.32*	0.25	0.12	0.3	0.28
Mean	*0.24*	0.17	0.13	*0.24*	0.41

The table values in italic indicates that the respective algorithm performed well comparing others excluding TREC manual

Table 4 Quality of results by P@10

Query Topic	Proposed generic framework	Semantic	Lucence	TREC automatic	TREC manual
451	0.8	0.7	0.5	*0.9*	0.8
452	*0.3*	0.2	0.2	*0.3*	0.9
454	*0.9*	0.8	0.8	*0.9*	0.8
457	*0.2*	0.1	0	0.1	0.8
465	*0.3*	*0.3*	0	0	0.9
467	*0.4*	*0.4*	*0.4*	0.3	0.8
476	0.4	*0.5*	0.3	0.1	1
484	*0.3*	0.2	*0.3*	0	0.3
489	*0.3*	0.2	0	0.1	0.4
491	0.2	0.2	*0.3*	0	0.9
494	*0.9*	*0.9*	0.8	1	1
504	0.1	*0.2*	*0.2*	0.5	1
508	0.4	*0.5*	0.1	0.3	0.3
511	0.6	0.4	0.5	*0.7*	0.2
512	0.4	*0.5*	0.2	0.3	0.3
Mean	*0.43*	0.41	0.31	0.37	0.69

The table values in italic indicates that the respective algorithm performed well comparing others excluding TREC manual

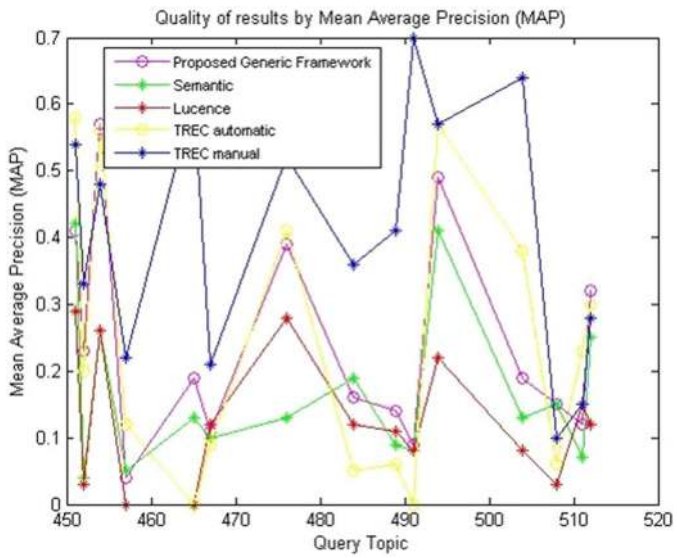

Fig. 35 Quality of results by mean average precision (MAP) graph

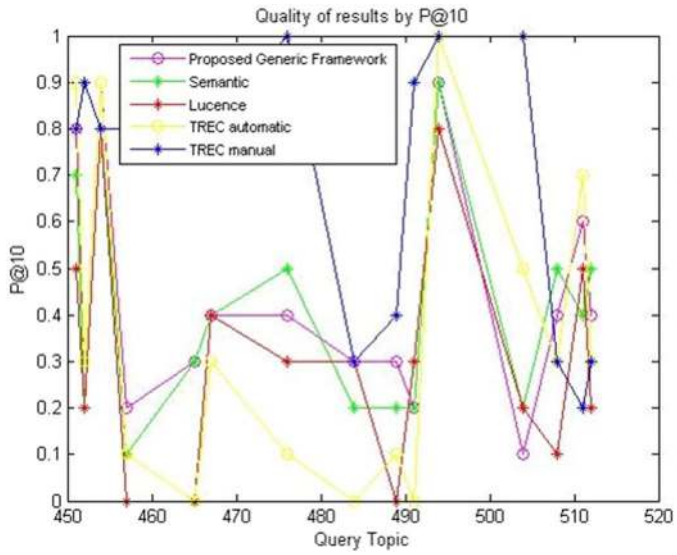

Fig. 36 Quality of results by P@10 graph

called MAP (Mean average precision) and P@10 (Precision at 10). The same graph is shown in Figs. 35 and 36. The MAP metric generates the overall performance in precision, recall and ranking. The P@10 metric relates to the accuracy of the top-10 results which are mostly discovered by the users. The TREC manual method will not be affected by these metrics because of manual adjustments to the query. The values that are bold represent the best scores for the respective topic and metric. From Table 4, the P@10, the generic framework proves 10 % better results than semantic approach, and outperforms the other three methods by providing highest quality for 65 % of the queries. It also obtained the highest mean value for this metric.

There are some limitations studied in semantic based approach [38], as it lacks in relevance judgement in TREC collection and its restrictive annotation process. But the proposed approach minimizes this disadvantage by combining ontology and NLP processing which is shown in the results. In TREC collection, only three possibilities should exist as the document may be judged as relevant, irrelevant or it could not be judged. As per the semantic retrieval approach, only 44 % of the results returned by it was previously evaluated in TREC collection and 66 % of the total set are non-judged, but they may be relevant. Using this it showed its improved performance. On the other hand, the

Table 5 Recall-precision averages for generic framework model

Recall level precision averages	
Recall	**Precision**
iprec_at_recall_0.00	0.9746
iprec_at_recall_0.10	0.8616
iprec_at_recall_0.20	0.7989
iprec_at_recall_0.30	0.6958
iprec_at_recall_0.40	0.5331
iprec_at_recall_0.50	0.3417
iprec_at_recall_0.60	0.2807
iprec_at_recall_0.70	0.2114
iprec_at_recall_0.80	0.1928
iprec_at_recall_0.90	0.1401
iprec_at_recall_0.10	0.0492
Average precision overall relevant docs	
Non-interpolated	0.4618

Table 6 Documents retrieved by generic framework vs semantic approach that are evaluated

Topic	**Generic framework evaluated (%)**	**Semantic approach evaluated (%)**
451	51.60	44.6
452	33.30	31.3
454	51.40	49.4
457	58.60	54.6
465	39.50	38.5
467	37.00	38.0
476	49.60	50.6
484	15.40	13.4
489	53.60	51.6
491	54.20	47.2
494	58.30	57.3
504	38.80	32.8
508	61.80	62.8
511	62.30	61.3
512	45.80	39.8
Mean	*45.54*	44.4

The table values in italic indicates that the respective algorithm performed well comparing others excluding TREC manual

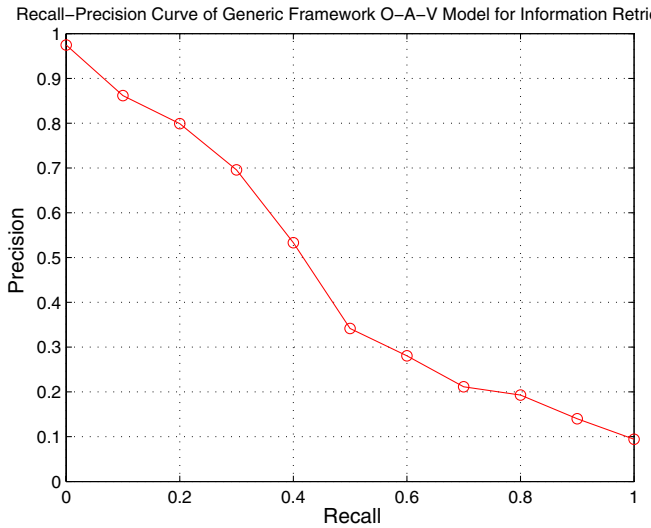

Fig. 37 Recall-precision curve of generic framework O-A-V model

generic framework outperforms than semantic and other approaches by its improved result of 45.54 % as shown in Table 6. The recall level precision average is shown in Table 5 The recall-precision graph is shown in Fig. 37. The summary statistics details for the proposed methodology is given below:

Summary statistics
Test title: Generic framework_O-A-V
Number of topics: 50
Total number of documents in overall topics
Retrieved: 50,000
Relevant: 4821
Rel_ret: 3215

Hence, the proposed model used the TREC_EVAL program for evaluating its retrieval performance, since it is the standard evaluation system for information retrieval and search engines. The results obtained for query topics were shown in Tables 3, 4, 5 and 6. It implies that the proposed framework gives an average improvement in precision-recall and that holds well when compared to other related works.

Conclusions

The amount of information on the web has increased exponentially in recent years. Going through every web result is time consuming for an impatient user who wants to obtain the best results with minimum work. Providing the top web results does not complete the task if the user still has to browse through them; providing semantically extracted O-A-V triplets with each web link will provide the user with valuable insight and save time. The scope of this ontology-driven information extraction is not limited to providing insight into the content of web pages or documents; it can also be used for the integration and sharing of information among various web resources. This information is machine interpretable and can be used by web agents to perform complex

operations and provide users with better search results. By using the proposed image ontology model, the system extracts the O-A-V triplets from the user's query and then uses it to match the appropriate image descriptions of the images stored in an ontology for improved image retrieval. These results are then ranked in a two-step process, first by decreasing order of number of keyword occurrences and further by using the keyword proximity score, proposed in this paper. The effectiveness of our proposed unified framework was tested by applying it to document retrieval and image retrieval. The controlled experiment demonstrates that retrieval is better and faster when our techniques have been implemented. Also, comparisons with related works and the system evaluation using TREC_EVAL suggests improvements over standard techniques.

Limitations and future work

Though the results show an improvements over existing information retrieval techniques, an independent standard benchmark is required for evaluating semantic search systems. The lack of such exclusive and specific benchmarks made it difficult for us to evaluate our system. Currently, the web based document retrieval system is in its most primitive state. Later, we will try to add semantic links within web pages to other web resources along with the integration of information using the ontologies of the target web resources. We developed an algorithm that determines the most apt triplets that should be displayed with each web link and a service that determines the mind-set and searching patterns of users by developing ontologies that enhance his search experience. The use of ontologies to connect web resources can also be used to validate the classification groups that an entity belongs to using web resources. Also, the keyword proximity score ranking technique is just one of the multiple techniques that can be employed while ranking images. In future, we will try to correlate O-A-V triplets extracted from text with features extracted from image processing for further improving image retrieval.

Authors' contributions

VV and MD constructed the problem statement, performed the literature review, formulated the core theory of the retrieval algorithms (document retrieval and image retrieval), evaluated the proposed framework and analysed the experiment results. PT collected and analysed the data, designed the experiment and implemented the algorithms. ML built the search engine used in experiment and the application for testing the proposed framework. VV, MD wrote the manuscript. PT contributed to edits in the manuscript. All authors read and approved the final manuscript.

Author details

[1] School of Computing Science and Engineering, VIT University, Vellore 632014, Tamilnadu, India. [2] School of Information Technology and Engineering, VIT University, Vellore 632014, Tamilnadu, India.

Competing interests

The authors declare that they have no competing interests.

Received: 16 June 2015 Accepted: 13 July 2016
Published online: 05 November 2016

References

1. Berners-Lee T, Hendler J, Lassila O et al (2001) The Semantic Web. Sci Am 284(5):28–37
2. Bizer C, Heath T, Berners-Lee T (2009) Linked data-the story so far. Semantic services, interoperability and web applications: emerging concepts, p 205–227
3. Meehan A, Brennan R, O'Sullivan D (2015) Sparql based mapping management. In: IEEE International Conference on Semantic Computing (ICSC), 2015. IEEE, New York, p 456–459
4. Heath T, Bizer C (2011) Linked data: Evolving the web into a global data space. Synth Lect Semant Web Theory Technol 1(1):1–136
5. Kompridis N (2000) So we need something else for reason to mean. Int J Philos Stud 8(3):271–295

6. Dhingra V, Bhatia KK (2015) Semcrawl: framework for crawling ontology annotated web documents for intelligent information retrieval. In: Intelligent distributed computing. Springer, Berlin, p 213–223
7. Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220
8. Sowa JF (1999) Knowledge representation: logical, philosophical, and computational foundations
9. Fellbaum C (1998) WordNet. Wiley Online Library, Hoboken
10. Miller GA, Beckwith R, Fellbaum C, Gross D, Miller KJ (1990) Introduction to wordnet: An on-line lexical database. Int J Lexicogr 3(4):235–244
11. McBride B (2004) The resource description framework (rdf) and its vocabulary description language rdfs. In: Handbook on ontologies. Springer, Berlin, p 51–65
12. Mony M, Rao JM, Potey MM (2014) Semantic search based on ontology alignment for information retrieval. Int J Comput Appl 107(10)
13. Enser PGB, Sandom CJ, Lewis PH (2005) Automatic annotation of images from the practitioner perspective. In: Image and video retrieval. Springer, Berlin, p 497–506
14. Hanbury A (2008) A survey of methods for image annotation. J Vis Lang Comput 19(5):617–627
15. Liu Y, Zhang D, Lu G, Ma WY (2007) A survey of content-based image retrieval with high-level semantics. Pattern Recognit 40(1):262–282
16. Vijayarajan V, Khalid M, Mouli PC (2012) A review: from keyword based image retrieval to ontology based image retrieval. Int J Rev Comput 12:1
17. Vijayarajan V, Dinakaran M (2013) Feature based image retrieval using fused sift and surf features. Int Rev Comput Softw 8(10):2500–2506
18. Shi R, Feng H, Chua TS, Lee CH (2004) An adaptive image content representation and segmentation approach to automatic image annotation. In: Image and video retrieval. Springer, Berlin, p 545–554
19. Wang M, Zhou X, Chua TS (2008) Automatic image annotation via local multi-label classification. In: Proceedings of the 2008 international conference on content-based image and video retrieval. ACM, New York, p 17–26
20. Escalante HJ, Hernández CA, Gonzalez JA, López-López A, Montes M, Morales EF, Sucar LE, Villaseñor L, Grubinger M (2010) The segmented and annotated iapr tc-12 benchmark. Comput Vis Image Underst 114(4):419–428
21. Grubinger M, Clough P, Müller H, Deselaers T (2006) The iapr tc-12 benchmark: a new evaluation resource for visual information systems. In: International workshop ontoImage, p 13–23
22. Prud E, Seaborne A, et al (2006) Sparql query language for rdf
23. Quilitz B, Leser U (2008) Querying distributed RDF data sources with SPARQL. Springer, Berlin
24. Jena A (2014) Fuseki: serving rdf data over http, 2014. http://jena.apache.org/documentation/serving_data/. Accessed 6 Jan 2015
25. Pujara J, Miao H, Getoor L, Cohen W (2013) Knowledge graph identification. In: The Semantic Web–ISWC 2013. Springer, Berlin, p 542–557
26. Singhal A (2012) Introducing the knowledge graph: things, not strings. Official Google Blog
27. Wang C, Gao M, He X, Zhang R (2015) Challenges in chinese knowledge graph construction. In: 31st IEEE International conference on data engineering workshops (ICDEW), 2015. IEEE, New York, p 59–61
28. Dzbor M, Domingue J, Motta E (2003) Magpie—towards a Semantic Web browser. In: The Semantic Web-ISWC 2003. Springer, Berlin, p 690–705
29. Bizer C, Lehmann J, Kobilarov G, Auer S, Becker Christian, Cyganiak Richard, Hellmann Sebastian (2009) Dbpedia—a crystallization point for the web of data. Web Semant Sci Serv Agents World Wide Web 7(3):154–165
30. Wollbrett J, Larmande P, De Lamotte F, Ruiz M (2013) Clever generation of rich sparql queries from annotated relational schema: application to Semantic Web service creation for biological databases. BMC Bioinform 14(1):126
31. Shekarpour S (2011) Dc proposal: automatically transforming keyword queries to sparql on large-scale knowledge bases. In: The Semantic Web–ISWC 2011. Springer, Berlin, p 357–364
32. Lopez V, Pasin M, Motta E (2005) Aqualog: an ontology-portable question answering system for the Semantic Web. In: The Semantic Web: research and applications. Springer, Berlin, p 546–562
33. Yang Y, Yang L, Wu G, Li S (2014) Image relevance prediction using query-context bag-of-object retrieval model. IEEE Trans Multimed 16(6):1700–1712
34. Vijayarajan V, Dinakaran M, Lohani M (2014) Ontology based object-attribute-value information extraction from web pages in search engine result retrieval. In: Advanced computing, networking and informatics, vol 1. Springer, Berlin, p 611–620
35. Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Tu SW (2003) The evolution of protégé: an environment for knowledge-based systems development. Int J Human Comput Stud 58(1):89–123
36. Bailey P, Craswell N, Hawking D (2003) Engineering a multi-purpose test collection for web retrieval experiments. Inform Process Manag 39(6):853–871
37. Knowledge Media Insititute (2009) Power aqua. http://technologies.kmi.open.ac.uk/poweraqua/trec-evaluation.html. Accessed 6 Jan 2015
38. Fernández M, Cantador I, López V, Vallet D, Castells Pablo, Motta Enrico (2011) Semantically enhanced information retrieval: an ontology-based approach. Web Semant Sci Serv Agents World Wide Web 9(4):434–452
39. Lucence J (2005) Jakarta lucene text search engine in java. http://jakarta.apache.org/lucene/docs/index.html