IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Linking Online News Semantically Using NLP and Semantic Web Technologies

Pratulya Bubna[1*], Shivam Sharma[2], Sanjay Kumar Malik[3]

[1*,2,3]USIC&T, GGSIP University, New Delhi, India

**Corresponding Authors: pratulyabubna@outlook.com*

Available online at: www.ijcseonline.org

Accepted: 25/Jul/2018, Published: 31/July/2018

Abstract— With the advent of internet, the world today is very closely connected. One can easily obtain information about the activities taking place in an overseas continent only in a matter of seconds. However, despite the world being so intimately connected, one doesn't find such connections in the news anyone comes across. It is seldom that the news websites provide readers with the news about events that happen in other parts of the world, different from the one the reader is currently in. Thus, there is a need to explore the idea of linking news semantically using different Semantic Web Technologies and concepts like Natural Language Processing (NLP) which may play a significant role in efficient and meaningful information extraction. In this paper, first, various concepts and technologies in regard to the above need have been explored. Second, an architecture is proposed, along with its implementation (in Python),for the media outlets (independents and aggregators) to explore the idea of linking their content semantically using the concepts of Linked Data and NLP. The proposed architecture is intended to be applied at the backend in order to render structured and linked data, with the intention of providing the readers with linked news.

Keywords— Linked Data, Natural Language Processing, Semantic Web Technologies, Ontology, Triplestore, Named Graph, Graph Database, Online News, Structured Data

I. INTRODUCTION

The news we come across often a times is dictated by intention of the provider — how much their coverage is — and somewhat gets limited to how the provider recommends them to us. The recommendation generally is based on methods of collaborative filtering wherein one is recommended next articles based on what other readers chose to read after reading the current article; or at times, new article recommendations from the same author whose current article one is reading are shown. Moreover, almost all the media outlets categorize the news based on a few set of sections such as Technology, Business, and Education etc. But categorizing into one of such sections is often decided manually and is therefore left to imagination of the human(s) who categorizes the article. However, it is never the case that an article belongs to only a particular section. True that it may belong to one major section, but in this paper, the idea presented is to harness the other subtopics/subsections that the article may belong to, and extract the underlying semantics that would help to decide a better scope for the articles. Such mainstream methods, although being used for quite some time now, often tend to limit the scope of news one comes across. It is not unless an event becomes breaking news that one is shown news different from that in their regular feed.

Therefore, the idea of applying linked-ness to news articles is explored in this paper, in which a plug-and-play type architecture harnessing the power of *Natural Language Processing (NLP)* is proposed. Various ideas to create a linked graph of news articles, connected through various contexts extracted using different NLP techniques, are discussed. Moreover, maintaining the essence of structured data, usage of *ontology*, one of the most significant semantic web technology for embedding semantics, is proposed.

Conforming with the idea of linked-ness and the usage of ontology, an implementation of the proposed architecture in Python language is also elaborated. Even though the same architecture can be used with conventional Database Management Systems (DBMS), usage of ontologies and graph databases are advocated to exploit the advantages which these semantic web technologies offer: empowering us with the 'reasoning' and 'inference' relation mappings that would immensely help in exploiting the linked-ness in a smarter way. This aligns with the view of how the online news is connected— semantically. Semantic web technologies allow us to harness the semantics in the connections between articles, contrary to conventional keyword mapping. Additionally, an NLP layer is used to extract meaningful information from the articles that would allow to maintain semantic relations between the articles.

Finally, various state-of-the-art NLP techniques that could be exploited in creating a semantically-connected graph of news articles are discussed.

Thus, not only the idea explores the connectivity among the news articles, but also tries to maintain the semantic relation between them. Implementing the proposed architecture would enable the media outlets (*independent outlets*: those that create/cover news; *aggregators*: that show news offered by such independent outlets) to structure and connect the articles in their backend, which would enable them to provide more inferential, reasonable, meaningful and connected news at the user-end.

The paper is organized as follows: In section II, various concepts and technologies backing the efficacy of the proposed architecture are explored and discussed. In section III, the proposed *Plug-and-Play* architecture is unraveled, elaborated with the utilities of each layer. An implementation of the proposed architecture is presented in section IV, in which certain *pieces* (functionalities) are *"plugged"* into different layers to suit a use-case. Finally, conclusions and future work possible to enhance the utility of the proposed architecture are discussed.

II. CONCEPTS AND TECHNOLOGIES USED

The significant concepts and technologies in this regard are as below:

A. Linked Data

Although the data existing in billions of websites hosted on World Wide Web is mostly overlapping and interrelated, modern data analytics applications can only use a minuscule amount of this data which may be in disorganized form. This unstructured nature of data is partially attributed to myriad of backend technologies used by websites for storing their data. With the advent of Artificial Intelligence, an evident need of organized data has kindled. The term *Linked Data* refers to a set of best practices for publishing and connecting structured data on the Web [1]. Using this Linked Data, a link is established between two entities, which are otherwise connected in the physical world, but are not connected within or across heterogeneous databases of various enterprises. The fundamental implementation of Linked Data requires developers to use a universal identifier to refer to an entity that exists inside their—or inside any other enterprise's—database. For this reason, as Tim-Berners Lee suggests, URIs (Universal Resource Identifiers) must be used to refer to any entity that exists in the database. This methodology creates a virtual link between entities that exist inside databases of enterprises. These virtual links on a global scale create a behemoth data cloud, a cloud which is an essential need of today's data analytics tools.

B. Resource Description Framework

RDF(*Resource Description Framework*) forms one of the basic building blocks for forming the web of semantic data. It isn't a data format, but a data model with a choice of syntaxes for storing data files [2].It provides us with a data model that helps us implement concepts of semantic web and linked data. A single unit of data in this data model is described as a triple. Each triple is like a little sentence that states a fact. The three parts of the triple are: the subject (entity), predicate (relationship), and object (entity or value). For example:

Subject	Predicate	Object
http://data.linkedmdb.org/page/film/33	http://dbpedia.org/ontology/writer	http://dbpedia.org/page/Quentin_Tarantino

RDF is used for storing data as well as for publishing ontologies. Linked data works on top of existing World Wide Web; it uses HTTP URIs as a unique identifier of a resource, extracts the resources through HTTP protocol and represents relationships between resources using RDF data model [3].

C. Ontologies

Ontologies are vocabularies that annotate a semantic relation between two entities that exist within or across databases. They provide virtual links of linked data with meaning or relation and help in creating a structured database from which knowledge can be extracted. Several communities have contributed to create widely accepted ontologies that define relationships in certain domains. Some examples include the Friend of a Friend (FOAF) ontology, the Gene Ontology, and the Dublin Core Ontology. These ontologies have defined relationships that most widely exist between entities residing in databases of enterprises. For example, following code depicts a relationship that exists between a publisher/writer of a news article and that news article itself (Figure 1).

```
@prefix : <http://search.com/newsarticle/ontology/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

:publisher
      rdf:type rdf:Property ;
      rdfs:comment "Publisher of the News Article." ;
      rdfs:label "Publisher" ;
      rdfs:range dcterms:Agent ;
      rdfs:domain :NewsArticle ;
      rdfs:subPropertyOf dcterms:publisher, dcterms:rightsHolder .
```

Figure 1. *Predicate (property) 'publisher' defined in RDF format for a News Article (entity).*

Ontologies are made using the RDF framework. Any developer can create his own ontology using the RDF

standards and map his ontology to other entities and ontologies. This in turn makes his ontology—and his data elements—mapped to other vocabularies, and therefore making it part of the linked data existing on semantic web.

D. Natural Language Processing (NLP)

NLP is a component of Artificial Intelligence which enables machines to recognize, understand and manipulate natural language speech or text. It is often used for extraction of entities, relationships between
entities, sentiments, and translation between natural languages. It encompasses a wide variety of information extraction subtasks that are proposed and discussed in later sections.

E. Graph Databases

Graph databases (GDBs) were introduced as a more scalable alternative of traditional relational databases. The fundamentals of GDBs lie upon graph theory and provide a dynamic structure. GDBs are systems which store data entities within nodes of a graph and establish a relationship between those entities through edges connecting those nodes [4]. Complex data, often stuck within the tight constraints of relational databases has found a more favorable storage space in GDBs. At present they have found broad applications in top tech companies: Google uses it for page ranking, Facebook and Twitter use it for storing real world relationships, and Era7 uses it for storing proteins and enzymes [5].

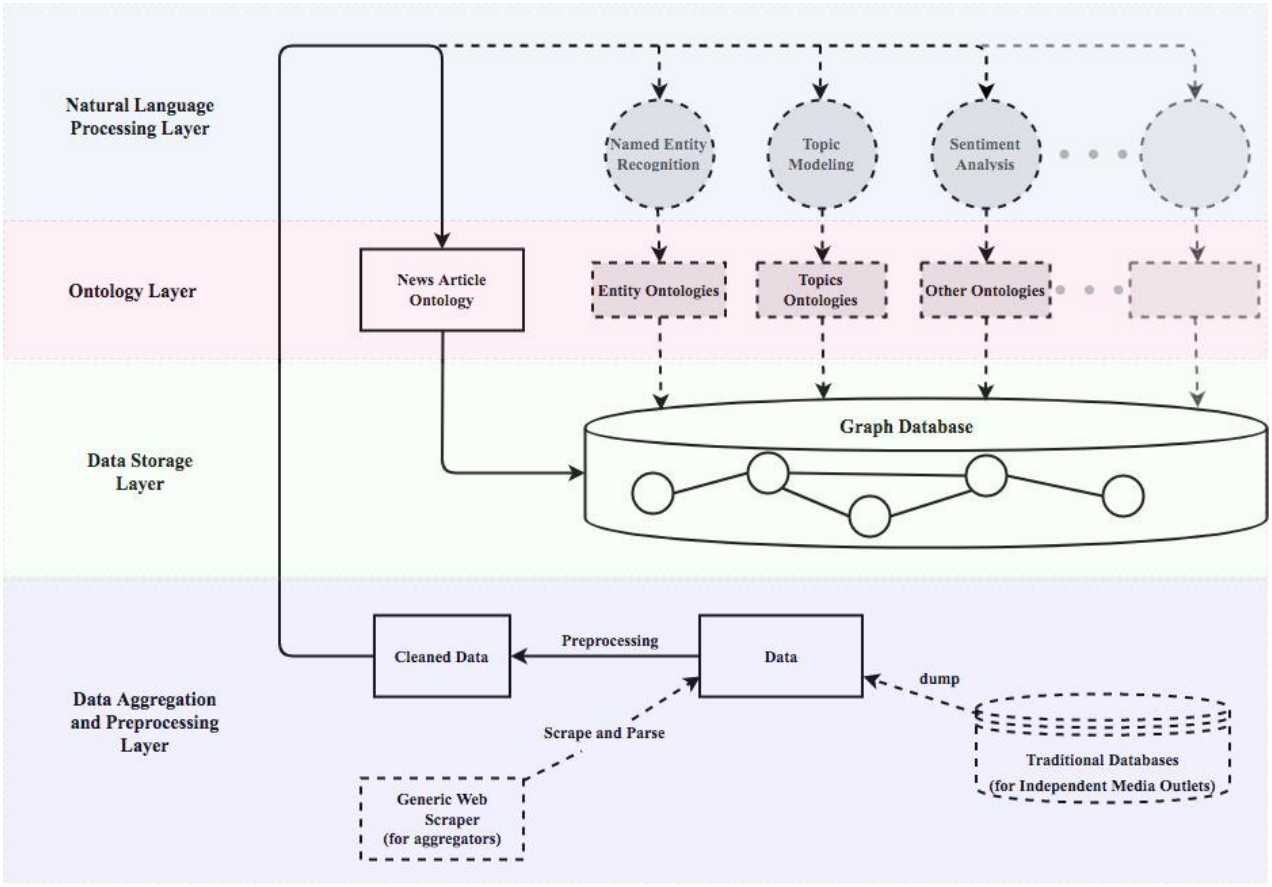

Figure 2.The proposed Plug-and-Play architecture diagram.

III. THE PROPOSED ARCHITECTURE

The architecture proposed is a customizable one, which we like to call the "*plug-and-play*" architecture as shown in Figure 2.The architecture comprises of four logically distinct yet connected layers. Since the architecture is a customizable one, integral pieces of the architecture are drawn with solid lines, whereas replaceable ones are drawn with dotted lines.

The number of "plugs", or pieces, or functionalities can be added, removed or modified depending on the use-case. Various NLP Layer plugs are strictly dependent on some Ontology Layer plugs. Therefore, multiple layers' plugs may be required to be altered to incorporate new structures.

A. Data Aggregation and Preprocessing Layer

This is the first layer of the architecture that corresponds to the data collection and subsequent cleaning. Cleaning of the data is an important step in order to ensure that only warranted data and acceptable data syntax is stored in the database, for example, transforming a local time zone specific timestamp to a Coordinated Universal Time (UTC) timestamp. Such preprocessing allows consistency to be maintained in the data store. The various components in this layer are:

1) Generic Web Scraper

Web Scraping is a method in which loads of data is programmatically extracted from website(s) that contain the desired information. Thereby, the scraper piece is relevant only for the purpose of collecting data from online, that is, for news aggregators. Thus, independent media outlets implementing the architecture won't necessarily require this. The scraper being discussed is generic in nature as it scrapes the attributes of the schema defined by Schema.org, which is a collaborative effort by Google, Yahoo, Microsoft and Yandex to promote the structuring of the data on the Internet based on the vocabularies (schemas) defined by their collective activity. Therefore, all the websites serving news articles and conforming to the structure define by Schema.org can be scraped using a generic scraper, without having to alter the scraper to accommodate HTML changes.

2) Traditional Database

Generally, independent news outlets store their data (articles) in traditional databases. Thus, in order to adhere to the architecture, the data with required attributes needs to be dumped to another storage discussed later

B. Data Storage Layer

In this layer, the storage of the data is handled. To work with the semantics involved in the domain of news and to exploit the inference and reasoning relations efficiently, the usage of GDBs, which use graph structures to store data (tuples), is proposed. Each relation is stored between nodes through edges representing certain properties, conforming to the graph principles. GDBs support semantic querying which essentially is extracting semantic relations between nodes using graph traversal techniques.

To be more specific, usage of Named Graphs is advocated. Named graphs identify each relation stored in GDB by assigning them URIs, providing the quads with web-global scope compared to arbitrary local statement names. Named graphs, a key concept of Semantic Web architecture, can be represented as a quad relation: <subject><predicate><object><graphname>; which is to say that named graphs contain an additional fourth attribute in a triple that holds the name assigned to a triple or a set of triples. To retrieve the data from named graphs, semantic query language like SPARQL can be used, which provides with necessary graph traversal syntax

C. Ontology Layer

This layer pertains to the creation of ontologies depending upon the data that is required to be stored. Ontologies contain schemas according to which the data is to be stored. In the architecture, a few types of ontologies need to be created, which can provide a structure for storing the relationships between entities.

1) News Article Ontology

An ontology corresponding to news article and its attributes is a required one. This would be used to dump the news articles' data onto the GDB. The attributes and types can be modified as per requirement. All the news articles and their attributes would be stored as URIs in the GDB.

Ontologies may need to be redefined based on the attributes of articles that need to be saved. For instance, if it's required to store the thumbnail URL of an article, then ontology needs to be updated accordingly to store the thumbnail URL attribute. Based on the attributes that are stored, inference and reasoning relations also need to be defined. (Figure 3 & Figure 4)

2) Other Ontologies

Ontologies are also closely connected with the NLP layer as whatever information is decided to be extracted from articles, needs to be stored as well.

Therefore, based on the NLP logic that would be implemented, additional ontologies may need to be defined that would store the extracted features and bind it with the already stored news articles.

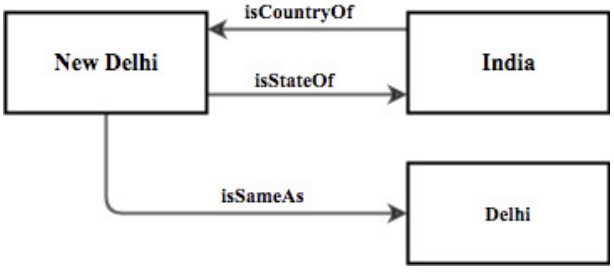

Figure 3. Entities (nodes) connected to each other through relationships (edges) defined by an ontology.

Each of the additional ontologies may include the necessary inference and reasoning relations between the news articles and the features extracted.

D. Natural Language Processing (NLP) Layer

NLP Layer corresponds to the logic that is applied in order to extract necessary features from the article texts that would help link the articles more meaningfully. The plug-and-play architecture provides the flexibility to plug any number of NLP techniques to this layer, thereby suiting the NLP pipeline according to the requirement. A few of the NLP techniques that could be plugged into the pipeline have been discussed below. However, it is to note that the scope of the pipeline isn't limited to only these techniques.

Figure 4. Figure showing how Delhi, an entity, is connected to India, another entity through infrence property.

1) Named Entity Recognition (NER)

NER is identification and subsequent classification of certain chunks of extracted information into labels such as people, organizations, locations, etc. NER models are trained on certain labeled corpuses and thereby enabling it to extract entities from new text [6].

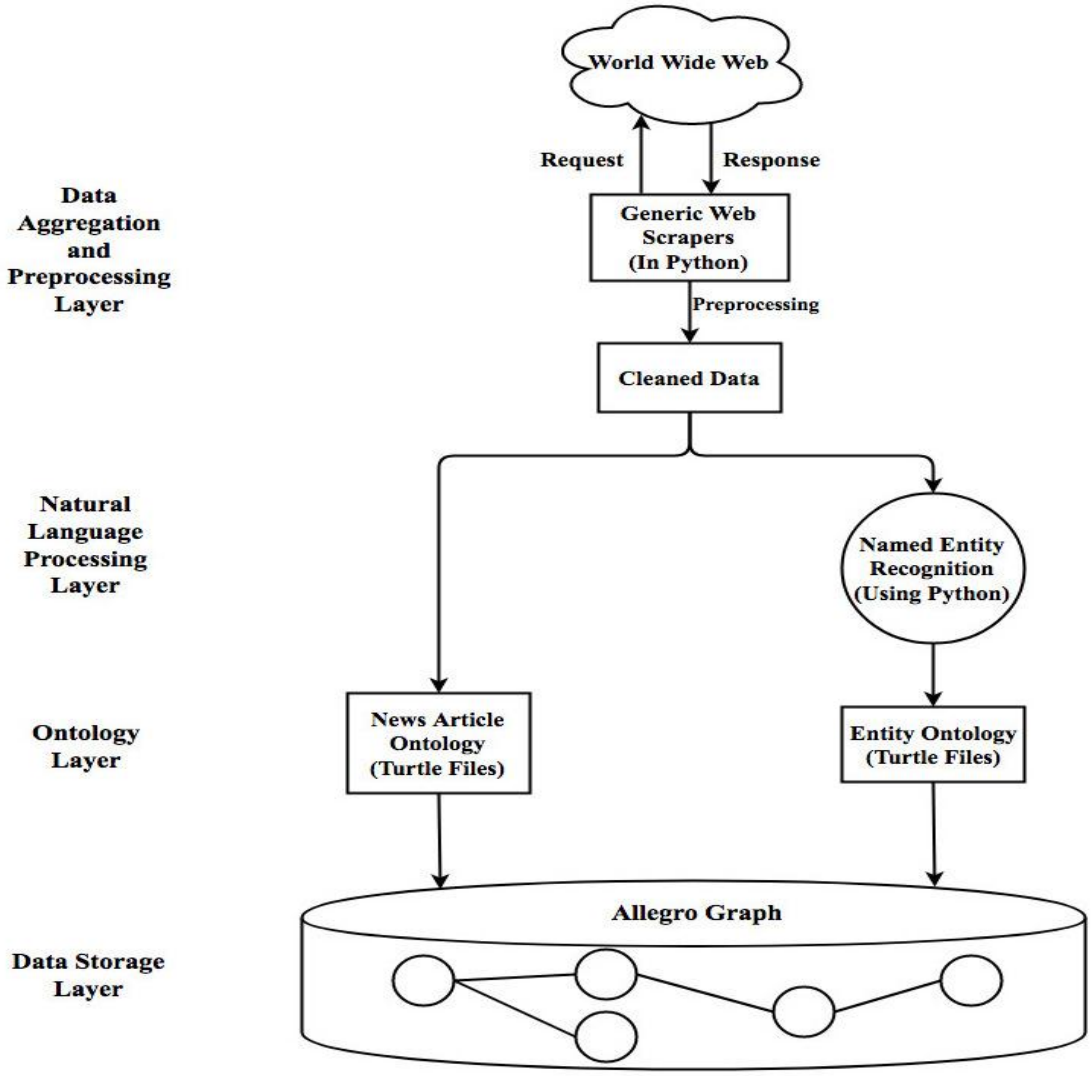

Figure 5. *A customized implementation of the proposed architecture.*

Every news article incorporates at least one of such entities. Connecting news articles through important entities is a very logical procedure as these entities are particular entity, then all the articles containing information about that entity can be linked. Such a process leads to a logical connection of ideas with entities. For instance, reading about an organization, we can easily traverse a graph of related articles in which certain influential people (Person Entities) have commented or discussed about that organization. This is often mentioned many-a-times in different articles. For instance, some predefined entities could be saved. Then whenever the NER piece in NLP layer extracts that leads to creation of structure in our world's news, on which human-like questions can be easily answered; for this case, "What are an X person's views about Y organization"?

Similarly, other reasoning and inferences can be mapped using other entities extracted according to the requirement. To work in tandem with the ontology layer, an ontology could be created containing schemas for different entities that are to be extracted and consequently their relation with news articles could be defined.

2) Topic Modeling

Topic Modeling is a statistical technique and a text-mining tool that discovers certain abstract topics a text is about by mining hidden semantic structures. Thus, topic modeling allows us to extract some important words from a text blob that essentially contain the most information about the text [7]. For instance, topic modeling applied on an article about increasing global warming could extract words like "pollution", "CO2", "global warming", "population", "crisis" etc. (depending on the implementation and the article content).For such a case, the article could be mapped with the major topics extracted, or even with the minor topics (whose probabilities are lesser). Such semantic-relation-mapping would allow semantic structure formation, in which similar topic articles are connected with each other. In such a case, additional weights could be assigned to the links between articles, assigning greater weight to articles that are connect through major topic, and assigning lesser weight to those connected with minor topics; thereby, forming the weak and strong ties between articles.

3) Sentiment Analysis

Sentiment Analysis is an NLP and text-analysis technique that is used to determine the intent, the attitude, the feelings, or the emotions of a document, interaction or an event, for instance, finding whether a particular user review is positive, negative or neutral [8].

Often a time it is important to analyze the sentiment of articles before showing it to the readers: whether the article has a harsh-negative tone, or is an uplifting-positive one. A particular scenario where sentiment analysis piece would offer utility is in the domain of Financial News. For instance, the volatility of the price for particular stocks can be assessed by analyzing the overall sentiment of that stock in the news. Negative emotions are often believed to have a larger and more lasting impact on market sentiment and dynamics when compared to positive or neutral ones [9]. Thus, there could a use-case for media outlets to provide information related to overall sentiments about various stocks. Apart from financial news, sentiment analysis has found wide range of applications in news data, Twitter and Facebook's data, and product reviews [10].

To implement such a use-case sentiment for each article could be stored. Thus, for each topic, for example, there could be links to different articles each describing the type of sentiment. To elaborate, for a topic of "global warming", set of articles that are positively-, negatively-, or neutrally-oriented can be distinctively extracted for that topic. The structuring of the data beneath and storing in a GDB would optimize and ease the retrieval.

E. Advantages of the Proposed Architecture

The Plug-and-Play architecture proposed in the paper has several advantages:

1) Semantic Linking

The intertwining of the NLP and Ontology layers in the architecture allows us to define and improve on the semantic relations in the text. The NLP layers enables us to extract the information, or the semantics from the text, and the ontology layer allows us to link the extracted information in a defined manner; thereby, linking our data semantically as shown in the user interface of a GBD in Figure 11 and Figure 12.

2) Structuring the Data

The usage of ontology to define schemas, allows us to store the data in GDB in a structured manner. Structuring the data enables us to easily retrieve the data while preserving the semantics involved among the data. It is because of structuring that it is possible to apply inferential and reasoning queries.

3) Flexibility

The "plugging" nature of the architecture allows maximum flexibility in using only those pieces that suit to the use-case requirement at hand. Moreover, the architecture minimizes the coupling between layers as it ensures maximum modularity in the logic.

IV. IMPLEMENTATION

For demonstration purposes, the proposed architecture has been implemented in Python language(version: 3.4.3)using various libraries. The architecture is customized to suit a use-case and only the required pieces (functionalities) are "plugged" into different layers (Figure 5). The flow-diagram in figure 5 depicts the flow of data: starting with scraping the websites to storing the data in *AllegroGraph* (GDB) [11] according to the ontologies defined. The implementation involves usage of NER piece in NLP layer, and corresponding entity ontologies (Person, Location etc.) in the Ontology Layer.

A. Data Collection and Cleaning

To collect news data from online, a generic web scraper was created to scrape the news websites that adhere to the NewsArticle schema defined by Schema.org. The scraper parsed the required attributes of the news articles. The generic nature of the scraper lies in the fact that with the same scraping and parsing logic, the scraper could scrape data from multiple media outlets that had varying HTML structures. The sole reason for the robustness of the scraper in successfully scraping the media outlets' websites is their adherence to the Schema.org. (FIG. 6) The parsed data was also cleaned to remove any ambiguities and any unwarranted data.

B. Using AllegroGraph as GDB

AllegroGraph, a closed source triplestore designed to store RDF Triples, was used as the GDB in the implementation. AllegroGraph has a client interface for Python, through which the interactions with the GDB were made (Figure 7). As a named graph, AllegroGraph provides functionality to name the triples stored, for the purpose of which, universal identifiers (URIs) were used.

The data could also be retrieved through its interface in Python using SPARQL semantic query language. Moreover, AllegroGraph had built-in reasoners and inference-drawing functionalities that could automatically create reverse relations according to defined ontologies.

```
def _parse_itemscope(scope_tag):
    has_itemscope = Parser._tag_has_attr('itemscope')
    has_itemprop = Parser._tag_has_attr('itemprop')
    itemscopes = scope_tag.find_all(has_itemscope)
    properties = {}

    if any(itemscopes):
        for each in itemscopes:
            result = Parser._parse_itemscope(each)
            if result[0]:
                properties[result[0]] = result[1]
    properties['type'] = urlparse(scope_tag['itemtype'])

    for prop_tag in scope_tag.find_all(has_itemprop):
        if not prop_tag or not prop_tag.attrs:
            continue

        itemprop = prop_tag['itemprop']
        if prop_tag.name in ['a', 'link']:
            properties[itemprop] = prop_tag['href']
        elif prop_tag.name == 'meta':
            properties[itemprop] = prop_tag['content']
        elif itemprop == 'articleBody':
            properties[itemprop] = prop_tag.text.strip()
        if not has_itemscope(prop_tag):
            prop_tag.decompose()

    return (scope_tag.attrs.get('itemprop'), properties)
```

Figure 6. *Code snippet for parsing logic of generic scraper implemented in Python.*

C. Writing Ontologies

First and foremost, NewsArticle ontology was defined that would contain all the necessary attributes and the corresponding relations about news articles (Figure 8). This ontology was used by GDB to load schema and allow the storage of news articles in a defined manner.

```
def establish_connection(self):
    try:
        server = AllegroGraphServer(host = '192.168.1.27',
                                    port = 10035,
                                    usr = self.username,
                                    password = self.password)
        catalog = server.openCatalog(None)
    except RequestError:
        print("Please run the script again.")
        exit()
    repo = None
    try:
        repo = catalog.getRepository(self.repo , Repository.OPEN
    except ServerException:
        print("No repository found. Creating repository...")
        repo_name = input("Enter the name of Repository.")
        if repo_name == 'y':
            repo_name = 'DailyNewsEngine'
        self.repo = repo_name
        catalog.createRepository(repo_name)
        repo = catalog.getRepository(self.repo , Repository.OPEN
    if repo != None:
        repo = repo.initialize()
        connection = repo.getConnection()
    self.connection = connection
    return {    'server' : server,
                'catalog' : catalog,
                'repository' : repo,
                'connection' : connection  }
```

Figure 7. A python script connecting to AllegroGraph programmatically using the python client of AllegroGraph. reverse relation between two entities. Here, 'belongsToCountry' acts as a reverse predicate to 'hasState'.

For the NER plug that was used in the NLP layer, separate ontologies (Figure 9) for required entities (Person, Organization etc.) were defined. Segregating the ontology logic in separate ontology files ensured achieving modularity in the implementation. All the ontologies were loaded into the AllegroGraph server before adding the data into the GDB. Any change in ontologies required restarting the AllegroGraph server.


```
@prefix : <http://search.com/newsarticle/ontology/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dctype: <http://purl.org/dc/dcmitype/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

 :NewsArticle
  rdf:type rdfs:Class ;
  rdfs:subClassOf dctype:Text .

 :headline
  rdf:type rdf:Property ;
  rdfs:comment "Headline of the News Article." ;
  rdfs:label "Headline" ;
  rdfs:range rdfs:Literal ;
  rdfs:domain :NewsArticle ;
  rdfs:subPropertyOf dc:title .

 :description
  rdf:type rdf:Property ;
  rdfs:comment "Description of the News Article." ;
  rdfs:range rdfs:Literal ;
  rdfs:domain :NewsArticle ;
  rdfs:subPropertyOf dc:description .
```

Figure 8. Two predicates (among many others): 'headline' and 'description', defined for a News Article.

```
@prefix : <http://search.com/location/ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dctype: <http://purl.org/dc/dcmitype/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

:Country
 rdf:type rdfs:Class ;
 rdfs:subClassOf dcterms:Location .

:State
 rdf:type rdfs:Class ;
 rdfs:subClassOf dcterms:Location .

:City
 rdf:type rdfs:Class ;
 rdfs:subClassOf dcterms:Location .

:hasState
 rdf:type owl:ObjectProperty ;
 rdfs:comment "Gives the state/province of a country." ;
 rdfs:domain  :Country ;
 rdfs:range   :State .

:belongsToCountry
 rdfs:domain :State ;
 rdf:type owl:ObjectProperty ;
 owl:inverseOf :hasState .
```

Figure 9. OWL property 'inverseOf' is used to create an inferred reverse relationship.

D. Extracting Information using NLP

To extract meaningful information from the scraped data about news articles and for data cleaning and preprocessing, NLP was used. Using the NLTK (Natural Language Toolkit) library in Python (as shown in figure 10), the data was cleaned using various functionalities provided like removal of stop words and unnecessary punctuation marks.

Plugging the NER piece in the NLP layer and making use of NLTK's NER functionality, labeled entities could be extracted for text. These labeled entities were further segregated and mapped onto the respective ontologies.

E. Implementation Flow

To exemplify the implementation(restricted to the Indian news), first, using the generic web scraper, data was scraped from multiple media outlets that conformed to the NewsArticle schema defined by Schema.org. Then, after loading the NewsArticle ontology to the AllegroGraph GDB, the scraped news articles' data was added to it using its Python interface.

Working with the NER plug, Python's NLTK library was used to extract the labeled entities from the articles. Having the necessary entity ontologies loaded onto to the GDB server, the articles and the entities were linked by forming necessary triples. For instance, if "NarendraModi" was found to be labeled as Person Entity, then a triple <NarendraModi><isMentionedIn><Article X>was created according to the definition in the Person ontology.

```
def named_entity_recognition(text, lower=True):
    tokenizer = PunktSentenceTokenizer()
    tokenized = tokenizer.tokenize(text)
    '''
    per: Person
    org: Organization
    loc: Location
    '''
    per, loc, org = [set() for _ in range(3)]

    for sentence in tokenized:
        words = word_tokenize(sentence)
        tagged = pos_tag(words)
        ner = ne_chunk(tagged)
        entities = traverse_ner(ner)
        for entity in entities:
            ne = entity[0]
            ent = entity[1].strip().lower()
            if ne == 'PERSON':
                per.add(ent)
            elif ne == 'ORGANIZATION':
                org.add(ent)
            else: # GPE, LOCATION
                loc.add(ent)

    return (per, loc, org)
```

Figure 10. Code snippet of NER function that extracts the entities (Person, Location, Organization) from a news text.

Statements with
headline
» as the subject.

Predicate	Object	Graph
rdf:type	rdf:Property	newsarticleontology
rdfs:comment	"Headline of the News Article."	newsarticleontology
rdfs:label	"Headline"	newsarticleontology
rdfs:range	rdfs:Literal	newsarticleontology
rdfs:domain	NewsArticle	newsarticleontology
rdfs:subPropertyOf	dc:title	newsarticleontology

Add statement...

Statements with
headline
» as the predicate.

Subject	Object	Graph
fc091c99-00e0-4008-b7d0-9015985eca0b	"Samson's 128 frustrates Sri Lanka, tour match ends in draw"	articledata
b7a3538e-5892-4bba-8813-62d3bf5a9f5f	"One lakh Indians book ticket for Mars"	articledata
8dc41042-39c6-40b2-b0ca-30084c5119a5	"Get a glimpse of Jupiter's stunning southern hemisphere"	articledata
84b4fef1-8def-4592-9219-3c7dc6e00d52	"Pic: Salman Khan's alleged girlfriend Iulia Vantur spotted on a night out in the city"	articledata
187dba4a-fdb1-4b8f-b01b-91eda2739aa7	"Now you can eat your lipstick, literally!"	articledata
1478ebf5-ec4b-4af0-bdff-909793a3a459	"Tata Sons to invest over Rs 36,000 crore in Tata Tele"	articledata

Figure 11.“Statements with headline as the predicate” depicts NewsArticles as the subject having "headline" as one of their predicates. The news articles are stored as Universally unique identifier (UUID) strings to represent each article uniquely in the GDB.[11]

Predicate	Object	Graph
mentionedOrganizations	"Cognizant Technology Solutions"	articledata
dcterms:issued	"2017-11-11T04:00:00+06:30"	(null)
category	f542f0c3-f83b-4cb8-a97d-d79acf5c2bd7	articledata
dcterms:date	"2017-11-11T04:00:00+06:30"	(null)
url	"https://timesofindia.indiatimes.com/business/india-business/cts-offers-cash-in-lieu-of-options-for-senior-mgmt/articleshow/61599938.cms"	articledata
dc:date	"2017-11-11T04:00:00+06:30"	(null)
headline	"Cognizant Technology Solutions offers cash in lieu of options for senior management"	articledata
dc:description	"Earlier this year, Cognizant Technology Solutions (CTS) announced its plan to return return $3.4 billion to its shareholders over the next two years through share repurchase and dividends."	(null)
description	"Earlier this year, Cognizant Technology Solutions (CTS) announced its plan to return return $3.4 billion to its shareholders over the next two years through share repurchase and dividends."	articledata
dc:title	"Cognizant Technology Solutions offers cash in lieu of options for senior management"	(null)
publishedOn	"2017-11-11T04:00:00+06:30"	articledata
dc:source	"https://timesofindia.indiatimes.com/business/india-business/cts-offers-cash-in-lieu-of-options-for-senior-mgmt/articleshow/61599938.cms"	(null)
secondaryLocations	dcterms:india	articledata
dcterms:type	f542f0c3-f83b-4cb8-a97d-d79acf5c2bd7	(null)
primaryLocations	dcterms:chennai	articledata
rdf:type	NewsArticle	(null)

Figure 12. A graphical view of AllegroGraph [11]."Statements with headline as the subject" depicts the relations in which Headline acts as the subject, and has its predicates correspond to some object. For example, “label” predicate of Headline has value “headline”.

V. CONCLUSIONS AND FUTURE WORK

- A customizable architecture or the *Plug-and-Play* architecture has been proposed which incorporates four layers, each catering to different logic.
- The proposed architecture is aimed at linking the news that media outlets (independents and aggregators) provide, in order to establish semantic connections among news articles.
- The semantic relations obtained using the proposed architecture would help to present the news in a more semantically intact manner.
- Any domain application that can incorporate the discussed concepts can make use of the proposed architecture.

Despite the advantages of the proposed architecture as discussed in the paper, there are a few short-comings to the current architecture and the implementation. However, significant work could be done in future that would augment the utility of the architecture. First, the concept of *Ontology Learning* – the automatic or semi-automatic creation of ontologies for a text– could be harnessed in order to automate the manual work currently involved at the ontology layer. Second, the NER model discussed in the paper (NLTK's built-in) could be improved. That is, using an NER model trained on news-domain corpora would render more accurate entity labels as compared to the NLTK's built-in, trained on general corpus. Third, the domain discussed in this paper is restricted to that of online news. However, the concepts of Linked Data along with Natural Language Processing can be applied to any domain where the scope lies.

REFERENCES

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far", International Journal on Semantic Web and Information Systems, Vol. 5, Issue. 3, pp. 1–22, 2009.

[2] B. DuCharme, "Learning SPARQL: Querying and Updating with SPARQL 1.1", O'Reilly Media, USA, pp. 19-45, 2013.

[3] I. Jacobs, N. Walsh, "Architecture of the World Wide Web", W3C Recommendation, Vol. 1, 2004.

[4] J. Pokorný, "Graph Databases: Their Power and Limitations" In Proceedings of 14th International Conference on Computer Information Systems and Industrial Management Applications, Poland, pp. 58-69, 2015.

[5] S. Patil, G. Vaswani, A. Bhatia, "Graph Databases: An Overview", International Journal of Computer Science and Information Technologies, Vol. 5, Issue. 1, pp. 657-660, 2014.

[6] D. Balasuriya, N. Ringland, J. Nothman, T. Murphy, and J. R.Curran. "Named entity recognition in wikipedia", In Proceedings of the Workshop on The People's Web Meets NLP, Singapore, pp. 10–18, 2009.

[7] R. Alghamdi, and K. Alfalqi. "A Survey of Topic Modeling in Text Mining" International Journal of Advanced Computer Science and Applications Vol. 6, pp. 147-153, 2015.

[8] I. Mohan, K. Janani, M. Karthiga, "A Survey on Sentiment Analysis on Social Network Data", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 2, Issue. 2, pp. 1-7, 2017.

[9] R.Schumaker, Y.Zhang, C. Huang, and H. Chen, "Sentiment analysis of financial news articles", Decision Support Systems, Vol.53, pp. 458-464, 2012.

[10] C. Nanda, M. Dua, "A Survey on Sentiment Analysis", International Journal of Scientific Research in Computer Science and Engineering, Vol. 5, Issue. 2, pp. 67-70, 2017.

[11] AllegroGraph 6.3.0 [2017-10-24], computer program, Franz Inc., CA 94612, USA.

Authors' Profile

Pratulya Bubna ispursuing his B.Tech. in the field of Information Technology from USIC&T, GGSIP University, Delhi.His research interests are: Deep Learning, Computer Vision, NLP and Semantic Web Technologies.

Shivam Sharma ispursuing his B.Tech. in the field of Information Technology from USIC&T, GGSIP University, Delhi. His research interests are: Computational Biology, Deep Learning and Semantic Web Technologies.

Sanjay Kumar Malik completed his Ph.D. in the area of "Semantic Web" from USIC&T, GGSIP University, Delhi. He is currently working as Associate Professor in University School of Information, Communication and Technology, GGSIP University. He has more than 18 years of industry and academic experience in India and abroad (Dubai and USA). His areas of research interest are Semantic Web and Web Technologies. He has several research papers in reputed international conferences and Journals. He has been session chair in several international IEEE/Springer conferences and honoured with third best researcher award in 2011 by GGSIP University for his contributions in research.

