Journal of Computer Science

Original Research Paper

Knowledge-based System of Indian Culture using Ontology with Customized Named Entity Recognition

Harsh Raj, Kumar Harsh, Payal Khattri and Md. Tanwir Uddin Haider

Department of Computer Science and Engineering, National Institute of Technology Patna, India

Article history
Received: 16-07-2021
Revised: 16-11-2021
Accepted: 01-12-2021

Corresponding Author:
Harsh Raj
Department of Computer Science and Engineering, National Institute of Technology Patna, India
Email: harshraj.ug18.cs@nitp.ac.in

Abstract: Indian culture is a blend of diversified ethics with different cultures and religions. There arise the sustainable challenges for the people of inter societies and religions to participate in the regional events with lack of cultural information like types of cuisines eaten; rituals followed etc. The existing search engines using a keyword-based approach make it difficult for people to find relevant information among millions of undesirable results. To overcome this problem, we applied the concept of the semantic web that facilitates unstructured data that is represented more precisely by using well-defined vocabularies such as Resource Description Framework (RDF), which is recognized by both people and machines. In this research, we have developed a Knowledge Base System (KBS) using an ontological approach (Ontology Development 101) integrating the concept of Customized Named Entity Recognition (CNER) that supports full-text query searching with structured information. To test the feasibility, several queries from our domain taken from different users were executed successfully. The precision of 0.96 justifies the efficacy of our framework. Finally, the evaluation results were compared with two previous methodologies which show that our methodology outperforms.

Keywords: Natural Language Processing, Ontology, Customized Named Entity Recognition, SPARQL, Knowledge Base

Introduction

India is a diverse country, as evident by its geography, population, culture and weather. Indian cuisine, festivals and traditions differ from region to region. People eat different cuisines, celebrate different festivals and follow different rituals, each with its significance. Travellers need to spend a considerable amount of time on the web studying the location they plan to visit. Before planning vacation, they usually look up details on the place's festivals, staple foods and wedding traditions. There are dozens of search engines available, but they perform keyword-based searches and are isolated from each other. People must inquire separately about food, festivals and rituals. The semantic web solves this issue by translating web data into machine-readable form. This assists the computer in comprehending web data and delivering accurate results to the query. The initial step is to scrape data from websites and build an ontology model. Ontology models link the data, convert it into machine-readable form, validate and infer new knowledge from these data. When a user submits a query to the ontology model, CNER is used to assign tags to the entities in the query that reflect food, festivals, or marriage. Data is extracted from ontology using these tags.

Although Named Entity Recognition (NER) tags are very supportive in web scraping and ontology development throughout several domain problems. But still, the major limitation with these NER tags is that there exists a very limited number of predefined tags mapped with real-world entities. Therefore, it has always been a big challenge for researchers to work on with only a limited number of predefined tags. This key factor enforces the system development process covering a very narrow and specific knowledge base system lacking complex query executions as the system is partially developed and is not user friendly. To avoid these types of inconsistency and system failure, we have introduced the concept of CNER for the ontology development process which incorporates both predefined tags and user-friendly customized tags. These CNER tags provide the flexibility to scrap all the relevant information which are very important for domain-specific ontology development but are not categorized with any of the predefined NER tags. Hereby, CNER tags are user-defined tags that allow the system to categorize with these types of exceptional data and precisely develop remarkable ontology systems.

The goal of our research is to develop a knowledge base system of Indian culture using ontology and CNER approach so that people can access precise data of different cuisines, festivals and marriage rituals on one platform. The ontology of our system has been modelled using the Ontology Development 101 method which consists of five phases, namely specification, integration, conceptualization, implementation and evaluation (Isa *et al.*, 2020). In the specification phase, we have selected Indian culture as our ontology domain, with our focus on food, festivals and marriage rituals in different regions of India. We chose to build an ontology from scratch during the integration phase because there was no acceptable ontology for Indian culture available on the web. The most crucial phase of the paradigm, Conceptualization and Implementation follows. Using the Protege 5.5 tool (Brownlow *et al*., 2015), we created an ontology about Indian culture using the web ontology language (owl). The modelling phase of ontology is followed by customized NER which identifies the key elements in the text thereby sorting the unstructured data and extracting the important information. This is followed by the model which converts natural language query to SPARQL using CNER. For this, CNER tags are mapped to the classes, subclasses and data properties of ontology followed by rule-based matching. The corresponding SPARQL query then generates the desired results for the user query. Lastly, the results phase explains the effectiveness of the model by testing it against the various queries on the Indian culture domain. The detailed implementation is discussed below in the paper.

This study contributed to the realm of Indian culture in many ways. The first is the ontological method and techniques utilised to create and develop the knowledge base of Indian culture. As far as we know, there has never been a study on ontological modelling of Indian culture; consequently, this is the first one. Secondly, we have a live website, where users can query about food, festivals and marriages in India. This platform is user-friendly and provides adequate results within microseconds. Furthermore, this ontological design could be used as a model for establishing and designing knowledge bases in other disciplines.

Related Works

Kumar *et al*. (2021) in their proposed system on Ontology-Based Full-Text Searching using NER, makes use of NER to apply tags to real-world concepts, match them to establish relationships among classes within ontologies and use this relationship between classes and tags to insert and extract data from ontologies. Kohli and Arora (2014) in their research work on Domain-Specific Search using Semantic Web, explain that semantic web gives appropriate results to complex user queries. Their work on the hotel domain describes a method of building ontology and querying it with the help of JENA API. Hasany and Selamat (2011) suggested a framework for hotel-related systems that includes a natural language querying platform for retrieving information from a shared interface for decision-making. Their research shows that semantic search saves time and helps people make better choices with fewer questions than searching on specific websites. Palaniammal *et al*. (2012) proposed a model for semantic search whose results are focused on the user's priorities while searching the tourism domain of interest. The conditional probability for a given input can be estimated using this proposed model and relevant information can be obtained by querying ontology. Collobert *et al*. (2011) proposed a single neural network architecture and learning algorithm that can be used for part-of-speech tagging, chunking, NER and semantic function labelling, among other natural language processing tasks. Their work can be used to create a publicly accessible tagging framework with high performance and low computational requirements. Hao (2011) in their research on the knowledge model for ontology-based knowledge base explain the advantage of using knowledge ontology technology in the field of knowledge engineering. The most effective method to coordinate a knowledge model to the knowledge base by the utilization of ontology reasonably is the key issue that they work out. Brownlow *et al*. (2015) proposed an ontological approach for creating an Andean weaving knowledge base. A significant piece of the research has been the modelling and representation of the knowledge of domain experts and other data about the textile object in the form of an OWL ontology and the development of search facilities for the ontology model. Eftimov *et al*. (2017) presented a novel NER method, called drNER, for knowledge extraction of evidence-based dietary information. DrNER is a rule-based NER that is divided into two stages. The first is concerned with detecting and determining the entities mentioned, while the second is concerned with selecting and extracting the entities. Bontcheva *et al*. (2013) proposed an open-source information extraction pipeline for microblog text. Their model discusses NER and Information Extraction (IE) for microblogs. Shelar *et al*. (2020) explained all the Named Entity Recognition approaches and their comparison for the custom NER Model. The paper explains the different NLP libraries and compares them based on training accuracy, F-score, prediction time, model size and ease of training. Isa *et al.* (2020) proposed an ontological approach for creating a brassware craft knowledge base. They have used ontology development 101 methodologies to model the ontology, then tested it against the 15 competency questions by manually converting the natural language query to SPARQL. All the five phases involved in ontology modelling are explained in their research work.

There has never been a study on the ontological modelling of Indian culture. The diversified nature of Indian culture and the absence of any knowledge base emphasize

DOI: 10.3844/jcssp.2022.172.186

the importance of information retrieval and ontology modelling. Furthermore, several research papers discuss ontology modelling using the Ontology 101 methodology, but none of them discusses the conversion of natural language queries to SPARQL utilizing CNER by applying suitable tags in the query. As a result, this research is critical.

Proposed Framework

The major goal of our research is to create an ontology-based information retrieval model that answers the user's textual questions about Indian festivals, food and weddings, in a personalized manner. For pre-processing, Natural Language Processing (NLP) techniques are used, so that our model can understand the user's queries and then they are also used to convert those questions into their corresponding SPARQL queries, which are used to extract data from our ontology. The first step was collecting the information related to Indian food, festivals and weddings and modelling it into an ontology. Then we process the user query. Our pre-processing module first applies tags to the words of the user's query using stanza's CNER and then maps those tags with the classes and subclasses of our ontology, then a rule-based matching is done to develop a relationship between the classes, subclasses and the data to be extracted. Then this relationship is mapped to the corresponding SPARQL, which is then used to extract relevant data from the ontology. Figure 1 shows our proposed framework to develop the knowledge base of Indian culture. This framework contains three modules which are Information Resources, Ontology Modelling and Query processing, which are elaborated below.

Information Resources

For building any ontology, information collection is a very important task because the results that we will provide to our users should be authentic and precise. Our system makes use of several websites that provide accurate information on Indian festivals, cuisine and weddings. All of the information related to the classes of our ontology is scraped from different websites using the python library selenium and beautiful soup and are filtered using CNER, to obtain only precise results. This information provides the detail about the Indian festivals in different regions, their significance, how they are celebrated, etc. and food eaten in different regions of India, their nutritional values, ingredients, etc. and the customs and rituals performed in different regional weddings of India. The festival details are scraped from festivalsofindia.in, food details are scraped from balancenutrition.in and wedding details are scraped from utsavpedia.com.

Ontology Modelling

The work in this study is focused on the domain of Indian culture, i.e., about Indian festivals, food and weddings. The ontology for the same has been developed using Protege 5.5 (Musen, 2015), a tool for ontology development by´ Stanford University. Several methods of modelling ontology have been proposed in the past like METHONTOLOGY (Fernández-López *et al*., 1997), Ontology Enterprise (Dietz, 2007), Ontology Development 101 (Noy and McGuinness, 2001), the CYC methodology (Lenat and Reed, 2002) and UPON (De Nicola and Michele, 2005). There is no such concept of best method among these since the one chosen for ontology modelling depends on the application, domain and the knowledge engineer's expertise on the domain. We have followed the Ontology development 101 methodologies for developing our ontology because of its flexible, less complex and convenient nature of representing the knowledge of a domain. It has been built from scratch since no ontology concerning our domain was available on the web. We have modelled our ontology based on five phases of Ontology development 101, namely-specification, integration, conceptualization, implementation and evaluation (Isa *et al*., 2020).

Specification

In this phase, we identify the domain, application and scope of the ontology to be developed. Indian culture was selected as our domain and we focused mainly on Indian food, festivals and weddings. This research aims to develop a knowledge base of Indian culture. To limit the scope of the ontology, a set of competency questions were taken from the set of queries provided by the users, which are listed in Table 1. These questions were taken randomly from a survey conducted by us in our department where each user posted their query about Indian culture concerning the domain of food, festivals and weddings. There were approximately 100 queries obtained from the users out of which 30 have been mentioned in the table.

Integration

It's nearly always worthwhile to study what others have done and see if there is any room for improvement and enhance current methodologies for our domain and job. Reusing old ontologies may be needed if our system has to interface with other systems that have already committed to particular ontologies or vocabularies. Many ontologies are currently accessible in electronic form and one may import them into their ontology creation environment. The way ontologies are expressed isn't necessarily relevant because many knowledge-representation systems tradeoffs between existing ontologies that match the domain on which those systems are working. Also, it may happen that a knowledge-representation system cannot operate directly with a particular design and semantics, mapping an ontology from one mannerism to another is usually simple. However, one can also design an ontology from scratch if no such ontology exists concerning the domain or the cost of integrating the existing ontologies into the study is more than that of those built from scratch. In our study of Indian culture, we modelled our ontology from scratch since no suitable ontology concerning our domain was available to integrate.

Conceptualization

In this phase, we define the classes and subclasses for our ontology as well as their structure under a taxonomy, which is a superclass-subclass hierarchy. This hierarchy is defined by the object properties and the class attributes are defined by the data properties. Figure 2 shows the classes and subclasses that were defined for the Indian culture ontology. These classes were defined in a top-down manner, where the broader classes like Food, Festivals, Marriage, etc. were defined at the first level and the specific ones, i.e., north-food, east-food, south-food, west-food, etc. were defined at the next level. As visible in Fig. 2, the superclass Thing is by default provided by the Protege interface. Then food, festivals, weddings and´ the four regions of India, namely – North, South, East and West are partitioned into classes. Further, branching has been done by providing the subclasses such as north food, north festivals and north marriage to the class North India; and similarly, for other regions. The subclass north food is also the subclass of Food, north festivals is a subclass of Festivals, north marriage is a subclass of Marriage and so on.

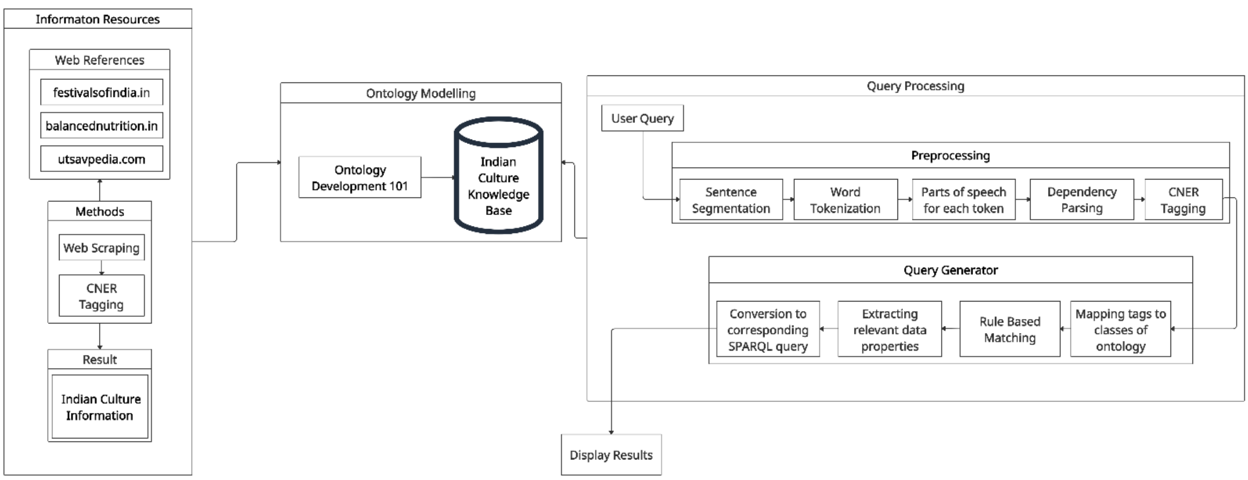

Fig. 1: Proposed framework to develop the knowledge base of Indian culture

Fig. 2: Classes and subclasses of Indian culture ontology

DOI: 10.3844/jcssp.2022.172.186

Table 1: Competency questions on Indian culture

S.NO.	Competency question	S.NO.	Competency question
1	What are the major festivals celebrated in South India? What is their significance?	16	Which Gods are worshipped during Holi, Diwali, Mahashivratri, and Chhath festivals?
2	What types of clothes do people wear in Onam?	17	List the north food options.
3	What food items are eaten in North India during the Holi festival?	18	What is the fat content of Dal Makhni? What is its protein content?
4	How do people celebrate the Chhath festival in West India?	19	How is Holi celebrated in West India? What is its significance?
5	What are the traditional dresses that people wear in East India during the Navratri festival?	20	What do people wear in Hanuman Jayanti? How is it celebrated?
6	How many days do people celebrate Ganesh Chaturthi in West India? What are the celebrations carried on each day?	21	What are dresses worn in Manipuri marriages? What are Manipuri marriage rituals?
7	Which food items are the most famous in East India?	22	During which time of the year Raksha Bandhan is celebrated?
8	What are the major ingredients of Paneer Tikka?	23	What is Chicken Kadai? What ingredients are used in it?
9	Which type of ingredients are mostly used in South India?	24	What food items are eaten in South India during the Pongal festival?
10	What are the nutritional contents of Apple Suji Kheer?	25	What are Bihari marriage rituals?
11	Why do people worship the Sun on Chhath festival?	26	Describe Assamese weddings?
12	Which particular festival is celebrated at this time of the year?	27	What is the food category of Bejar-roti? What is its energy content?
13	What are the marriage rituals followed in Assamese Weddings?	28	How do people celebrate the Campanula boat festival in Kerala?
14	Which type of clothes do people of Tamil Nadu wear during marriages?	29	When is Mahavir Jayanti celebrated? What is its importance?
15	How much time will it take to attend a marriage in West India?	30	What is Baingan ka Bharta? What is its fat content?

Different relationships are defined in the taxonomy, such as subclass-of, superclass-of, disjoint class and instance-of (Noy and McGuinness, 2001). The “is-a” connection denotes a taxonomic subdivision. If each instance of class A is the same as that of B, then A is a subclass of B. For example, the north food class is a subclass of Food because each North-Indian food is an instance of food as well. The superclass-of relationship is the inverse of this subclass-of relationship, i.e., Food is a superclass of north food. A disjoint-class relationship is that which depicts a situation in which two classes cannot have the same instance. This connection improves the system’s ability to produce better ontology. For example, in our case, the Food, Festivals and the Marriage are disjoint classes, as no food item can be an instance of the festival and no festival can be an instance of food. The classes and subclasses are related by the object properties. For example, North India has North Food north food.

West India has West Festivals West Festivals

South India has South Marriage South Marriage and so on

Figure 3 shows this relationship with the ontology visualization plugin of Proteg e called Onto Graf. In the´ said figure, the arrow from North India class to the north food, north festivals and north marriage signify that all of these are related by certain object properties to the class North India.

Then the data properties of the classes are defined such as, ‘Name’, ‘State’, ‘Significance’, ‘Celebration’, ‘Food’, ‘Clothes’ etc. for the Festival class; ‘Name’, ‘Description’, ‘Ingredients’, ‘Calories’, etc. for the Food class and ‘State’, ‘Rituals’, ‘wedding dresses’ for the class Weddings. Then we populate the classes and the subclasses with instances that were imported from the excel data that we had gathered earlier. Again, another Protege plugin named´ Cellie was used to achieve this. A set of transformation rules were written to map the columns of excel data to the data properties of the individual classes. A snapshot of the same is Fig. 4.

Implementation

Our formal semantic model was created using Web Ontology Language (OWL). The OWL is a set of knowledge representation languages for creating ontologies. Formal semantics is a feature of the OWL languages. They’re based on the Resource Description Framework (RDF), an XML standard for objects in the format of subject, predicate and object, developed by the World Wide Web Consortium (W3C) (RDF, 2021). The ontology was created with Protege 5.5, which allows for the´ construction and editing of one or more ontologies in a single workspace with a fully configurable user interface. Interactive navigation of ontology relationships is possible with several visualization tools available in Protege´ Inconsistencies can be tracked down with the use of advanced explanation support. Ontology merger,

DOI: 10.3844/jcssp.2022.172.186

transferring axioms between ontologies, renaming multiple entities and other refactoring procedures are provided.

Our ontology can also be modeled mathematically using First-Order-Logic, which is an extension of the Propositional Logic and used to express complex natural language information concisely. Besides Propositional Logic, it also assumes objects, relations and functions to exist in the world to represent the required knowledge. Here are the set of rules expressed in first-order-logic that concerns our knowledge-base.

Data-Property relations – Such as All Food items has a food name can be expressed as-

∀x Food(x) → has Food Name(x)
Similarly, ∀x Food(x) → has Food Type(x)
∀x Food(x) → has Food Region (x)
∀x Food(x) → has Food Energy (x)
∀x Food(x) → has FoodH Meter (x)
∀x Food(x) → has Food Protein (x)
∀x Food(x) → has Food Carbohydrate(x)
∀x Food(x) → has Food Fat (x)
∀x Food(x) → has Food Ingredients (x)
∀x Festival(x) → has Festival Name (x)
∀x Festival(x) → has Festival State (x)
∀x Festival(x) → has Festival Region (x)
∀x Festival(x) → has Festival Time (x)
∀x Festival(x) → has Festival Description (x)
∀x Festival(x) → has Festival Significance (x)
∀x Festival(x) → has Festival Celebration (x)
∀x Festival(x) → has Festival Food (x)
∀x Festival(x) → has Festival Clothes (x)
∀x Marriage(x) → has Wedding State (x)
∀x Marriage(x) → has Wedding Region (x)
∀x Marriage(x) → has Wedding Origin (x)
∀x Marriage(x) → has Wedding Dresses (x)

Similarly, the Parent-Child relations are shown as – ∀x East Food(x) → Food(x), signifies all items in East Food are members of Food class, or East Food is a child of Food. Similarly, North Food, South Food, West Food are children of Food.

∀x North Food(x) → Food(x), ∀x South Food(x) → Food (x), ∀x West Food (x) → Food(x).
Also, ∀x North Festivals (x) → Festivals (x),
∀x South Festivals (x) → Festivals (x),
∀x East Festivals (x) → Festivals (x),
∀x West Festivals (x) → Festivals (x). and
∀x North Marriage (x) → Marriage (x), ∀x South Marriage (x) → Marriage (x), ∀x West Marriage (x) → Marriage (x), ∀x East Marriage (x) → Marriage (x).

Similarly, there are other parent-child relations such as
has North Festival (North India, North Festivals),
has North Food (North India, North Food),
has North Marriage (North India, North Marriage),

Which signifies North Festivals, North Food and North Marriage are related to North India by the above relations. Similarly, rules are written for South India, West India and East India as:

has South Festival (South India, South Festivals),
has South Food (South India, South Food),
has South Marriage (South India, South Marriage),
has West Festival (West India, West Festivals),
has West Food (West India, West Food),
has West Marriage (West India, West Marriage),
has East Festival (East India, East Festivals),
has East Food (East India, East Food),
has East Marriage (East India, East Marriage).

Now that we have represented our knowledge in first-order-logic. To test our system, we have given queries to our knowledge base to get relevant results. For example, consider a query – "How is Holi celebrated in West India? What is its significance?"

The above query can be written using the first-order-logic as–

{<hasFestivalCelebration(z), hasFestivalSignificance(z)> | ∀x Festival(x) {∃y West Festival(y) ∈ x {∃z ∈ y | hasFestivalName(z)="Holi" }}}

Other examples of query-translation are shown as – "What is the fat content of Dal Makhni? What is its protein content?"
{<hasFoodFat(y), hasFoodProtein(y)> | ∀x Food(x) {∃y ∈x | hasFoodName(y)="Dal Makhni" }}

"Which type of clothes do people of Tamil Nadu wear during marriages? "
{<hasWeddingDresses(y)> | ∀x Marriage(x) {∃y ∈x | hasWeddingState(y)="Tamil Nadu" }}

In the above manner, we can represent each of our query in the first-order-logic and get the required tuples as results.

Evaluation

The last and final phase of ontology modelling is to evaluate the performance of our ontology, compared to other existing methods of information extraction. The evaluation process is discussed in detail in the Results section.

Query Processing

Query processing modules consist of two submodules. The first submodule is named as the pre-processing that uses the steps of NLP, such as sentence segmentation, word tokenization, parts of speech of each token, dependency parsing and customized named entity recognition. Here the input to each method is the output of the previous method. The details are described in section.

DOI: 10.3844/jcssp.2022.172.186

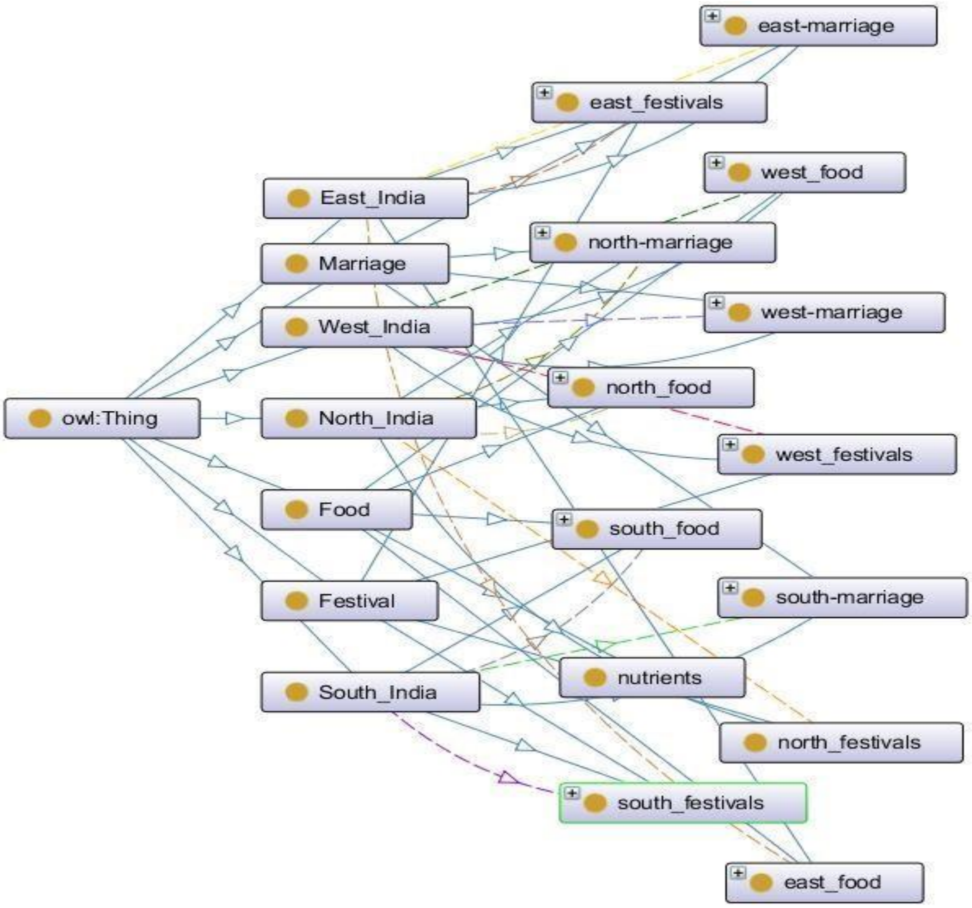

Fig. 3: Snapshot showing relationship between the classes and subclasses (Ontograf view)

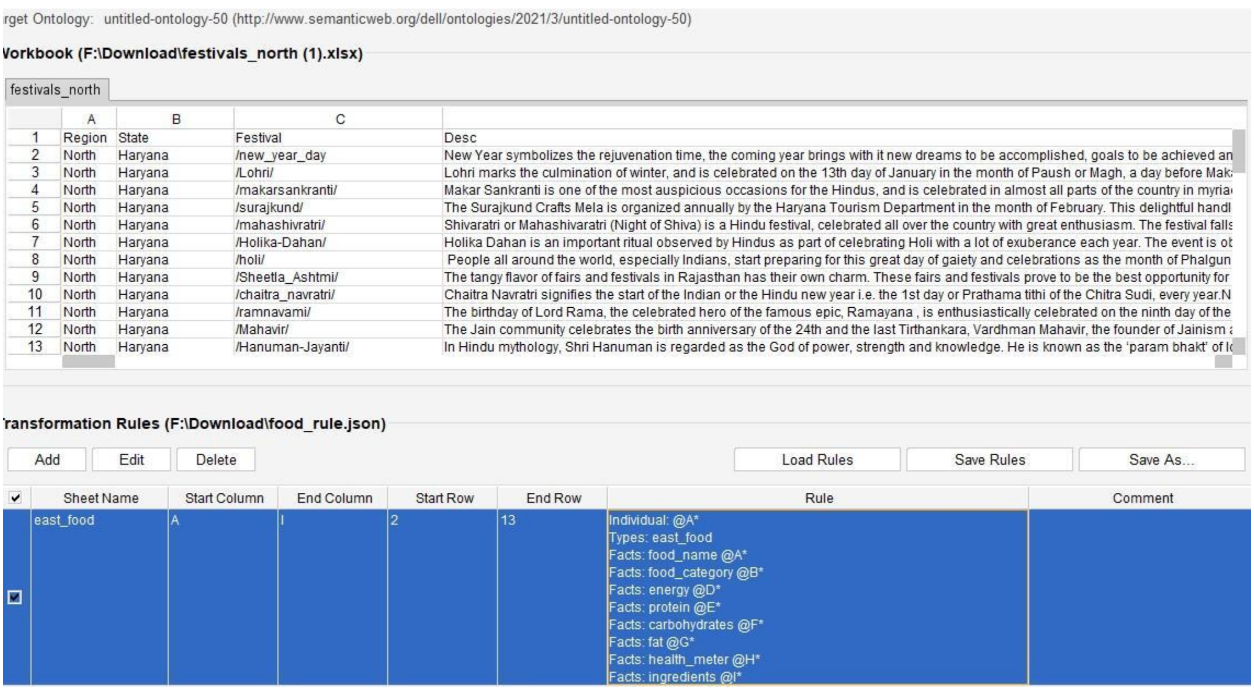

Fig. 4: Snapshot showing importing data from excel in protege to create instances of the class

Whereas the second submodule named Query Generator which consists of processes such as mapping CNER tags to classes of ontology, rule-based matching, extracting relevant data properties and conversion to corresponding SPARQL query. The details are described in section 3.3.2. For queries, the users are provided with a web interface in which they can put queries related to Indian dishes, festivals and weddings and see the results of their query upon clicking search.

Pre-Processing

Textual data does not convey meaningful information to the machine as it cannot understand them. The natural language query obtained from the user has to be first refined to convert it into a corresponding SPARQL query. So, our model pre-processes the textual information using the NLP pipeline. Since the query may contain multiple sentences, they are broken down into individual sentences and then NLP is applied to each of the smaller sentences.

Sentence Segmentation

The user query may contain multiple sentences. To understand the context of each sentence, the text has to be broken down into individual pieces of information, so that the relationship between each such information can be well understood. In our work, the user input query is broken down into its component sentences using stanza's sentence segmenter (Qi *et al*., 2020) which breaks down paragraphs into sentences that are separated by delimiters like full stop and question marks.

For example –

Consider the query – "How is Holi celebrated in West India? What is its significance?" The corresponding segments for this query are –

1. How is Holi celebrated in West India?
2. What is its significance?

Word Tokenization

Tokenization is done to assign parts of speech to each word so that the CNER tagger can easily identify nouns, verbs, adjectives, etc. for mapping it to real-world entities. The sentences are broken down into their component words which are separated by certain delimiters. In English, space is one such delimiter. The tokenization is done by using the Flash Text library of Python (Flash Text–Keyword Processor, 2021) which is inspired by the Aho-Corasick algorithm and builds a trie-dictionary from a sentence and extract the keywords from each sentence. For example -

"How is Holi celebrated in West India?"

The list of extracted keywords is –

["How", "is", "Holi", "celebrated", "in", "West", "India"]

Parts of Speech for Each Token

It is important to know the parts of speech of each word in the phrase to grasp its context. We have used the Stanza library of Python (Qi *et al*., 2020) which assigns each token a POS tag, i.e., noun, verb, adjective, etc. It makes predictions about the tag or label that is most likely to be attached to its context using its training pipeline and statistical models. It uses the bidirectional long-short-term memory to achieve this tagging (Qi *et al*., 2020). For example, considering the above query –

"How is Holi celebrated in West India?" The POS – tagger assigns the following tags to each token –

How|ADV is|AUX Holi|PROPN celebrated|VERB in|ADP
West|PROPN India|PROPN?|PUNCT

Here, we can see that "How" is assigned as an adverb (ADV), whereas "Holi" is assigned as a proper noun (PROPN). Further, to improve the pre-processing, we have used dependency parsing which identifies relationships between two terms of the phrase.

Dependency Parsing

The technique of assessing the grammatical structure of a phrase based on the dependencies between the words in a sentence is known as dependency parsing. Various tags indicate the relationship between two words in a phrase in dependency parsing. The dependency tags are the tags that are used to indicate that something is dependent on something else. For example, in the phrase 'colourful festival' the term 'colourful' modifies the meaning of the term 'festival'. As a result, there is a dependency between the 'festival' and 'colourful', with the festival acting as the head and the 'colourful' acting as the dependent or kid. The ADJV tag, which stands for adjectival modifier, represents this reliance. This dependency parsing improves our pre-processing steps and hence enables better understanding of the query. Again, we have used the Stanza' dependency parser (Qi *et al*., 2020) for the same.

Customized NER

To understand the meaning of the query, one has to extract keywords that represent real-world entities and establish a relationship between them to obtain precise results. NER is one such NLP technique that recognizes entities from the input sentence and categorizes it into predefined tags, such as Person, Location, Organization, etc. It can also be customized to categorize entities based on the system's domain. We have used Stanza's NER-Processor (Qi *et al*., 2020) for customized NER tagging. We have trained the model to recognize food, festivals, region, location, etc. Stanza's NER-Processor can recognize entities like location, person, object, organization and etc. but to recognize Indian food and festivals we had to train it accordingly. We have collected data about Indian festivals, from festivalsofindia.in and other online domains that include 200 Indian festivals.

Similarly, we have collected information of about 400 Indian dishes from balancenutrition.in and other online domains. A snapshot of training data has been shown in the Fig. 5 below. The snapshot shows the entity type against the name of festival or food. There are other entity types also present in the training dataset which are Person, Location, Region and Organization.

For example, consider the query – "When is Holi celebrated in North India?" After tokenizing the query, the tokens are passed through the CNER tagger which displays the following results, as shown in Fig. 6. As we can see in this figure the tag values of the list, 'India' represents a location, 'Holi' a festival and 'north' represents a region. Similarly, different dishes are categorized under the 'FOOD' tag, names under the 'PERSON' tag and so on. These tags form the basis of our natural-language query conversion to SPARQL, which is described in the next section.

Query Generator

Since the machine cannot understand natural language queries, therefore, the query has to be converted to some machine-readable form, so that precise results can be extracted. In this study, SPARQL query is generated from the user query by using techniques like CNER and rule-based matching. Since we have obtained the important entities from the user's query in the form of CNER tags, they must be mapped to the classes, subclasses and relevant data properties of the ontology. It is also important to understand the relationship between the entities so that a correct SPARQL query can be generated which can extract precise results from the ontology. The steps for the same are described below –

Mapping CNER Tags to Classes of Ontology

The CNER tags obtained from the user query are mapped to their corresponding classes, subclasses and data properties. For example, the query "When is Holi celebrated in North India?","Holi" is tagged as a festival that maps to the Festivals class of ontology. Similarly, "north" is recognized as a region that maps to the Region attribute of the Festival class. This Mapping is done by using the difflib library of python.

Rule-Based Matching

Now we have a list of CNER tags mapped to the classes and subclasses of the ontology. To generate a SPARQL query that gives the correct results, we must know which data properties the user has asked for and from which category, i.e., food, festivals, or marriages. For this, we have maintained a hierarchical combination of all possible paths of the classes, subclasses and their data properties in the form of a tree, where the root node is the superclass Thing and its children are - Food, Festivals and Marriage. The next level consists of the data properties of each category and each one of these properties consists of its synonyms as children, since the query may contain synonyms of the property, instead of the exact property itself. We have used the WordNet library of Python to obtain the synonyms (NLTK-Wordnet, 2021). We have also created rules for question words like "When, Where, How, What", etc. which will tell us that the user is asking for "time", "place or location", "methods" and "description", respectively.

Since we have a list of CNER tags, we have to search for the path which will help us to get the exact results. So, we traverse from top to bottom of the tree, matching the CNER tags with the nodes of the tree and we store the ones which match our requirements. We need not traverse the whole tree because we know which category the user is interested in and then continue searching in that subtree. For example, we will consider the query discussed in the previous section, "When is Holi celebrated in north India?". The list of CNER tags of this query contains Holi as Festival, north as region, India as location. Now we start traversing our tree. Since the list of tags contains festivals, we are only interested in the subtree of the node Festival. Now, the Region: North tells us that only North Indian Festivals are to be taken into account. Then we search for the Holi festival and the question word When signifies that the user is asking for the time of celebration of Holi. In this way, we store the particular classes, subclasses and data properties which will help us in generating the SPARQL query.

Extracting Relevant Data Properties

In the previous section, we mentioned that we have added the synonyms of each data property in the searching tree, since the query may not contain the exact property defined in the ontology. For matching the synonyms with their parent word, we have used the Flash Text library of Python (Flash Text–Keyword Processor, 2021), which is used to find or replace keywords from the sentence. For example, the keyword deity is replaced by God and so on and all relevant data properties are stored in a list, which will later be used to generate the SPARQL query.

Conversion to Corresponding SPARQL Query

Now that we have the list of required classes, subclasses and data properties for finding precise results of the user's query, it is easy to generate a SPARQL query from them. We have defined all the required PREFIXES of the query, i.e., the RDF-schema and the namespace of our ontology. Then under the SELECT keyword, we append all the data properties as we obtained in the previous steps, which of course, are the ones the user is interested in and in the WHERE keyword we define the subject, predicate and object which are the category (food, festival, marriage), its corresponding object property and the name of the property, respectively. An example of the query conversion is Fig. 7. The pseudocode for the above steps is mentioned in Algorithm 1.

```
['Vishu is an Indian festival.', {'entities': [[0, 5, 'FESTIVAL']]}],
['Pooram is an Indian festival.', {'entities': [[0, 6, 'FESTIVAL']]}],
['Nehrutrophy boat race is an Indian festival.',
 {'entities': [[0, 21, 'FESTIVAL']]}],
['Champakkulam boat race is an Indian festival.',
 {'entities': [[0, 22, 'FESTIVAL']]}],
['Onam is an Indian festival.', {'entities': [[0, 4, 'FESTIVAL']]}],
['Kojagori-lakshmi-puja is an Indian festival.',
 {'entities': [[0, 21, 'FESTIVAL']]}],
['Cochin-carnival is an Indian festival.',
 {'entities': [[0, 15, 'FESTIVAL']]}],
['Assamese-labra is an Indian food.', {'entities': [[0, 14, 'FOOD']]}],
['Egg-lettuce-and-tomato-sprouts-wrap is an Indian food.',
 {'entities': [[0, 35, 'FOOD']]}],
['Masor-tenga is an Indian food.', {'entities': [[0, 11, 'FOOD']]}],
['Spicy-assamese-fish-curry is an Indian food.',
 {'entities': [[0, 25, 'FOOD']]}],
['Spinach-khar is an Indian food.', {'entities': [[0, 12, 'FOOD']]}],
['Bengali-murghir-jhol is an Indian food.', {'entities': [[0, 20, 'FOOD']]}],
['Bhuna-khichdi is an Indian food.', {'entities': [[0, 13, 'FOOD']]}],
['Cholar-dal-quarantine-sp is an Indian food.',
 {'entities': [[0, 24, 'FOOD']]}],
['Fish-pulao is an Indian food.', {'entities': [[0, 10, 'FOOD']]}],
```

Fig. 5: A snapshot of our training data of CNER

```
[[{'end_char': 38,
   'len': '1',
   'phrase': ['India'],
   'start_char': 33,
   'tag': 'B-LOC'}],
 [{'end_char': 12,
   'len': '1',
   'phrase': ['Holi'],
   'start_char': 8,
   'tag': 'B-FESTIVAL'}],
 [{'end_char': 32,
   'len': '1',
   'phrase': ['north'],
   'start_char': 27,
   'tag': 'B-REGION'}]]
```

Fig. 6: CNER tags applied to the query)

DOI: 10.3844/jcssp.2022.172.186

Fig. 7: SPARQL query for given natural language query

Results

This section depicts the working of our proposed framework by examining it against the complex user queries. It provides a detailed explanation of all the stages involved in the NLP pipeline. Let us first examine the system against the following user query.

Algorithm 1: To covert Natural Language to Machine understandable code which can be further structured to SPARQL

```
function NLTOSPARQL (x);
  Input: Natural Language Query
  Output: Machine Understandable Query
Store all food, festival and marriage attributes ← food
  attribute,festival attribute,marriage attribute. lst ←
  extract relevant keywords from x (using flashtext's
  keyword processor)
if f ind(NamedEntity) ==
   " food"or" f estival"or"marriage" then
  for i ← 0 to
   size(f ind(NamedEntityRecognition) 1 do
      query.append("NER TAG"+ Named Entity
      recognition[i])
      if tags ==
        " food"or" f estival"or"marriage" then
          category ← tags;
          for i ← 1 to size(lst) 1 do
              if category == " f ood"andi! =
                food attribute then
                  lst.remove(food attribute[i])
              else if
               category == "festival"andi! = festival
               attribute then
                 lst.remove(festival attribute[i])
              else if
               category == "marriage"andi! =
               marriage attribute then
                 lst.remove(marriage attribute[i])
          end
```

DOI: 10.3844/jcssp.2022.172.186

```
          else
        end
    else
        category ← find and store close match of all
        keywords among commonlist of food, festival
        and marriage
end
if size(lst) == 0 then
      if ′HOW′ in x or ′Where′ in x then
          lst.append(’celebration’)
      if ′When′ in x then
          lst.append(’time’)
      if ′What′ in x then
          lst.append(’description’)
      else
          lst.append(’name’)
      end
else
end
```

How is Holi celebrated in West India? What is its significance? Figure 8 shows the sequence of steps followed to get the most relevant result of a user query with minimum search time. The first step of the sequence is sentence segmentation where the given user query is segmented into sentences to make the processing easier. This is followed by the word tokenization step which finds all the tokens in the sentence to make tagging easier. Then, the part of speech is recognized in the third step to understanding the context in which the sentence is used. Thereafter, the dependency parsing block is used to identify the relationship among the words in a sentence.

Finally, custom NER tagging is applied. It extracts the most significant words from the user query. This reduces the total search time as compared to a normal web search where all the keywords present in the user query are searched in articles present on the web. CNER tagging selects the subset of words from the user query and stores them separately thereby reducing the search time. These CNER tags are used to get the desired result of the query by top-down parsing of the ontology graph. These tags are mapped to classes, subclasses and data properties of the ontology, by matching them against the pre-defined tags in the ontology model. We have created an API in a flask that does this conversion from natural language query to SPARQL query and the generated query is sent to Apache Jena Fuseki server integrated with our web app http://indianculture.herokuapp.com/ that gives the desired results, as shown in Fig. 9.

A comparison chart always helps in finding the similarities and the differences between the different methodologies for knowledge base creation. It also depicts the advantages and limitations of one framework over others. Table 2 shows our comparison chart with some of the papers discussed in our related works.

The below table shows the efficiency of our framework against the previously implemented methodologies. As compared to the work done by Isa *et al*. (2020), information collection based on interviews limits the scope of their search space and also affects the precision of their results, whereas our system scrapes data from online sources and filters it using CNER, which gives an added advantage of precise results and makes our system automated. Unlike their system where natural language is converted to SPARQL manually, we have developed a system that automatically generates SPARQL for corresponding Natural Language Query. The major advantage of our work is that we have an evolving system where additional data can be easily added without writing explicit rules for them, unlike the methodology proposed by Kohli and Arora (2014), where a new rule has to be generated for each input set. Also, if one needs to add additional subdomains to our knowledge bases, like dance and art culture, it can be easily integrated since all the classes and subclasses related to it are automatically inserted in our ontology, because our model gathers data and populate our ontology using CNER and then information can be retrieved easily. The only rule in our framework is to identify the subdomain of the query asked and rest all the processes are automated. The precision of 0.96 justifies the efficacy of our framework. This precision was obtained by running 50 queries from our domain, out of which 48 gave correct results and 2 of them gave irrelevant results, which depicts why semantic search is better than keyword-based searching.

Table 2: Comparative results with existing work

Authors	Approac	Conversion of natural language to machine understandable query	Evaluation precision	Recall
Isa *et al*. (2020)	Ontology Development 101	Manually (with the help of experts JENA)	Ontology Quality Evaluation (4.03/5)	---
Shruti Kohli and Sonam Arora (2014) [5]	Google AJAX API, JENA API to retrieve results	Rule-Based Conversion from User input to SPARQL. New Rule for	Precision = 0.93 (44 out of 47 relevant results)	Near to 1 (ontology having records)
Our proposed work	Ontology Development 101, CNER based query conversion, JENA API	CNER-based Conversion from Natural Language queries directly into SPARQL. No explicit rule is required.	Precision = 0.96 (48 out of 50 relevant results)	Near to 1 (ontology having approximately 1000 records)

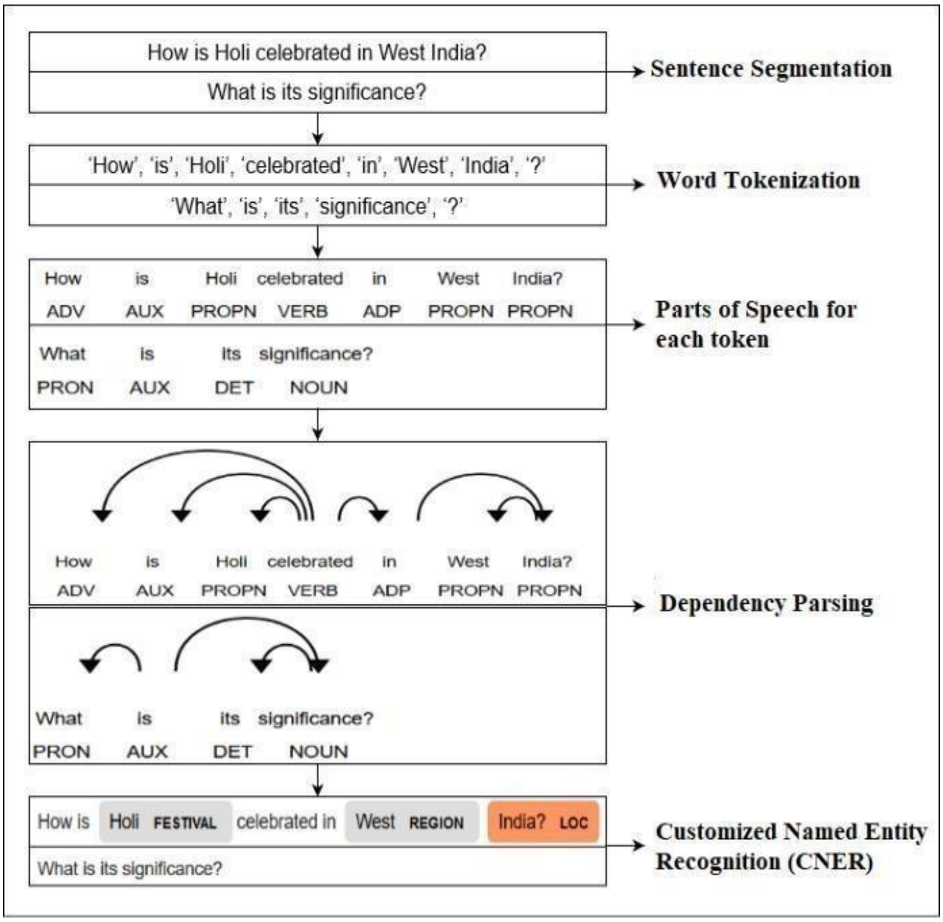

Fig. 8: Processing steps of a user query in NLP Pipeline

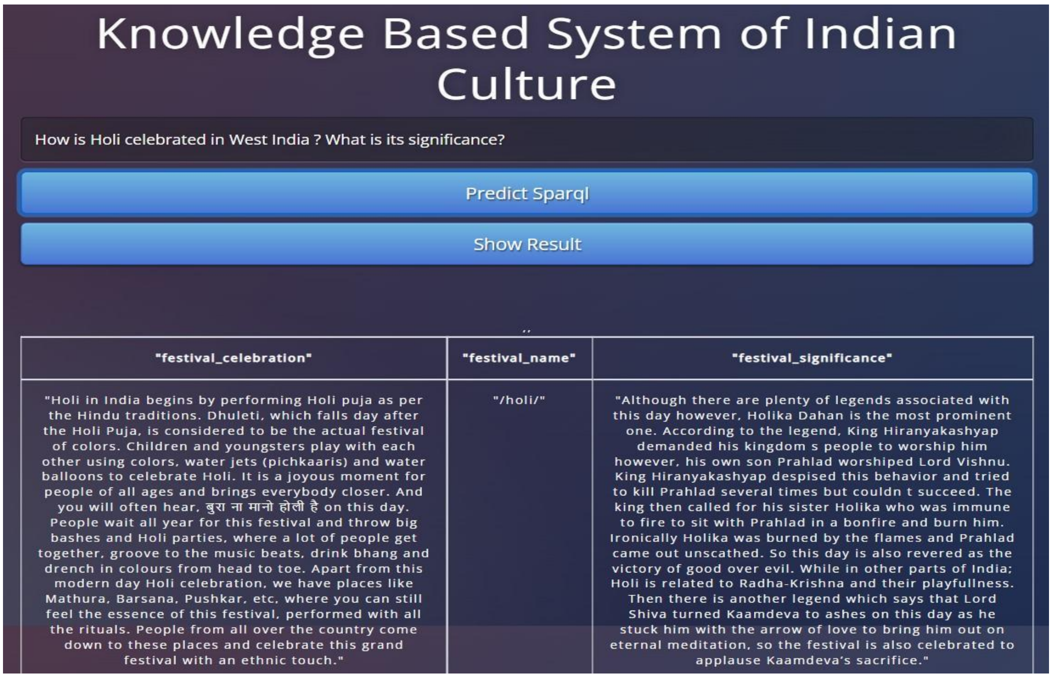

Fig. 9: User Interface with query and results) Table 2. Comparative results with existing work

Conclusion and Future Works

In this article, we have presented the knowledge base of Indian Culture developed using an ontological approach integrating the concepts of Customized Named Entity Recognition (CNER). This CNER effectively scrapes relevant information from multiple websites about Indian food, festivals and marriages in different regions. The key advantage of our scraping system is to inter-link multiple websites' relevant information and further derive new information. Further, CNER is also used to tag the stored information and the user's query to establish a relationship that helps in automatically generating the corresponding SPARQL for the user's natural language query, which is consequently used to extract relevant data from the ontology.

We have discussed the Ontology Development 101 methodology for data collection, design and implementation of the knowledge base and evaluation of the ontology results. The efficacy of our suggested framework is demonstrated in the results section. It can process complex queries consisting of large paragraphs with the overall precision of 0.96. (48 correct results out of 50 test queries) and a recall of near 1 (for our ontology consisting of 1000 records).

However, our model is limited to the discrete queries on food, festivals, or marriage. It cannot find accurate results for the queries which are a union of food, festival and marriage. The future work includes expanding the proposed framework to provide results of all types of queries and improving the accuracy of the CNER tags applied on the user query. The inclusion of images and videos to the query results will also prove to be very informative and our framework will be expanded accordingly. Another feature of suggesting correct queries to users can be added if they misspell them while typing. All of these can be added in future works.

Author's Contributions

Harsh Raj: Worked on query generation submodule.

Kumar Harsh: Worked on query preprocessing module.

Payal Khattri: Worked on Information Gathering module

Md. Tanwir Uddin Haider: Worked on Ontological Modelling module.

References

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013, September). Twitie: An open-source information extraction pipeline for microblog text. In Proceedings of the international conference recent advances in natural language processing RANLP 2013 (pp. 83-90). https://aclanthology.org/R13-1011.pdf

Brownlow, R., Capuzzi, S., Helmer, S., Martins, L., Normann, I., & Poulovassilis, A. (2015). An ontological approach to creating an andean weaving knowledge base. Journal on Computing and Cultural Heritage (JOCCH), 8(2), 1-31. doi.org/10.1145/2700427

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE), 2493-2537. https://www.jmlr.org/papers/volume12/collobert11a/collobert11a

De Nicola, A. M., & Michele, N. R.. (2005). "A Proposal for a Unified Process for Ontology Building: UPON". 655-664. doi.org/10.1007/11546924 64.

Dietz, J. L. (2007). Enterprise ontology-understanding the essence of organizational operation. In Enterprise Information Systems VII (pp. 19-30). Springer, Dordrecht. doi.org/10.1007/978-1-4020-5347-4 3

Eftimov, T., Korousic, S. B., Korosec, P. (2017). A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. PLoS ONE 12(6): e0179488. doi.org/10.1371/journal.pone.0179488

Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. https://oa.upm.es/5484/

Hao, C. (2011). "Research on Knowledge Model for Ontology- Based Knowledge Base". 201 International Conference on Buisness Computing and Global Informatization. doi.org/10.1109/bcgin.2011.105

Hasany, N., & Selamat, M. H. (2011, August). Answering user queries from hotel ontology for decision making. In 2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing (pp. 123-127). IEEE. https://ieeexplore.ieee.org/abstract/document/6047854

Isa, W. M. W., Zin, N. A. M., Rosdi, F., Sarim, H. M., Wook, T. S. M. T., Husin, S., ... & Lawi, S. K. (2020). An Ontological Approach for Creating a Brassware Craft Knowledge Base. IEEE Access, 8, 163434-163446. doi.org/10.1109/ACCESS.2020.3022795.

Kohli, S., & Arora, S. (2014). Domain specific search engine based on semantic web. In Proceedings of the Third International Conference on Soft Computing for Problem Solving (pp. 217-224). Springer, New Delhi. doi.org/10.1007/978 -81-322-1768-8 20

Kumar, K., Haider, M., Uddin, T., & Ahsan, S. S. (2021). Ontology-Based Full-Text Searching Using Named Entity Recognition. In Advances in Communication and Computational Technology (pp. 211-222). Springer, Singapore. doi.org/10.1007/978-981-15-5341-7

DOI: 10.3844/jcssp.2022.172.186

Lenat, D., & Reed, S. (2002). Mapping ontologies into Cyc. In AAAI 2002 Conference Workshop on Ontologies for the Semantic Web, Edmonton, Canada. https://www.aaai.org/Papers/Workshops/2002/WS-02-11/WS02-11-010.pdf

Musen, M. A. (2015). The protégé project: A look back and a look forward. AI matters, 1(4), 4-12. doi.org/10.1145/2757001.2757003

NLTK-Wordnet) (Accessed 20th June 2021) (https://www.nltk.org/ modules/nltk/corpus/reader/wordnet.html

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. https://corais.org/sites/default/files/ontology_development_101_aguide_to_creating_your_first_ontology.pdf

Palaniammal, K., Devi, M. I., & Vijayalakshmi, S. (2012, April). An Unfangled approach to semantic search for e-tourism domain. In 2012 International Conference on Recent Trends in Information Technology (pp. 130-135). IEEE. https://ieeexplore.ieee.org/abstract/document/6206758

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082. https://arxiv.org/abs/2003.07082

RDF. (2021). – Semantic Web Standards). 15 https://www.w3.org/RDF/

Shelar, H., Kaur, G., Heda, N., & Agrawal, P. (2020). Named entity recognition approaches and their comparison for custom ner model. Science & Technology Libraries, 39(3), 324-337. doi.org/10.1080/0194262X.2020.1759479