Ontology-Driven Data Retrieval: A Protege-Based Approach for Distributed Databases

Kranthi Kumar.R

Department of CSE,JNTUH College of Engineering,Hyderabad,Andhra Pradesh

Email Id: kranthikumar.rudrarapu@gmail.com

B. Padmaja Rani

Dept. of CSE, V.N.R Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Andhra Pradesh

Email Id: padmaja_jntuh@jnuth.ac.in

Abstract: *In today's data-rich environment, the demand for effective database queries has intensified. Researchers are actively seeking methods to enhance information retrieval processes. Our paper explores three key strategies: combining datasets, constructing a database ontology, and implementing semantic information retrieval with the ontology.Firstly, merging datasets allows for a comprehensive pool of information, enabling more thorough analysis and extraction of insights. Secondly, the development of a database ontology provides a structured framework for organizing and categorizing data, facilitating more efficient queries. Finally, leveraging semantic information retrieval based on the ontology enhances the precision and relevance of search results by understanding the contextual meaning of queries and data relationships.Our paper have explained retrieval using D2RMAP approach and SPARQL approach.By integrating these approaches, our research aims to streamline information retrieval processes, making them more effective and user-friendly in navigating the complexities of modern databases. This holistic approach promises to advance the field of database management and contribute to more informed decision-making in various domains.*

Keywords: *Ontology, Distributed Databases, Protege, Information Retrieval, Semantic Information Retrieval*

Introduction

In the vast landscape of today's data-driven world, effective information management stands as the linchpin of success. From the inception of data into databases to its retrieval and utilization, every step hinges on robust information management practices. As data proliferates exponentially across diverse sources and formats, the need for efficient and scalable information management solutions becomes increasingly paramount.At the heart of this information ecosystem lies database management, serving as the foundation upon which organizational data assets are built and maintained. However, the sheer volume and complexity of modern data landscapes pose significant challenges to traditional database management approaches. As databases grow in size and diversity, conventional methods of data retrieval through structured queries may prove inadequate in capturing the full breadth and depth of available information.

Enter ontology-based retrieval—a paradigm shift in information management that leverages semantic structures to unlock the latent potential of distributed databases. Ontologies offer a conceptual framework for representing knowledge within specific domains, encapsulating not only the entities and attributes within the data but also the relationships that define their semantic context. By imbuing databases with semantic richness, ontology-based retrieval transcends the limitations of conventional query-based approaches, enabling more precise and contextually relevant data extraction.

This paper delves into the transformative potential of ontologies in the realm of data retrieval from distributed databases. Rather than relying solely on rigid query structures, ontology-based retrieval harnesses the power of semantic reasoning to infer implicit relationships and contextual nuances within the data. Through the use of ontologies, disparate databases can be seamlessly integrated into a cohesive knowledge graph, facilitating holistic information retrieval across distributed data sources.

The adoption of ontology-based retrieval represents a paradigm shift in information management paradigms, offering several distinct advantages over traditional approaches. Firstly, ontologies provide a common semantic framework that fosters interoperability and integration across heterogeneous data sources. By harmonizing disparate data models and vocabularies, ontologies enable seamless data exchange and interoperability, thereby overcoming the siloed nature of traditional databases.

Secondly, ontology-based retrieval enhances the precision and relevance of data retrieval by leveraging semantic relationships encoded within the ontology. Unlike traditional query-based approaches that rely solely on syntactic matching, ontology-based retrieval incorporates semantic reasoning to infer implicit connections and derive meaningful insights from the data. This semantic enrichment not only enhances the accuracy of search results but also enables more nuanced queries that consider the underlying semantics of the data.

Furthermore, ontology-based retrieval empowers users to express complex information needs in a more intuitive and natural manner. Rather than crafting intricate queries using arcane query languages, users can interact with the data using high-level semantic constructs, such as classes, properties, and relationships defined within the ontology. This abstraction layer shields users from the complexities of underlying data structures, making information retrieval more accessible and user-friendly.

Moreover, ontology-based retrieval facilitates adaptive and personalized information access, allowing users to tailor their queries based on individual preferences and context. By capturing domain knowledge within the ontology, the retrieval system can adapt its behavior dynamically to accommodate changing user needs and preferences. This adaptive approach to information retrieval ensures that users receive relevant and timely information that aligns with their specific requirements.

In summary, ontology-based retrieval represents a paradigm shift in information management practices, offering a holistic and contextually rich approach to data extraction from distributed databases. By leveraging semantic structures encoded within ontologies, this approach transcends the limitations of traditional query-based retrieval methods, enabling more precise, relevant, and adaptive access to distributed data sources. As organizations grapple with the challenges of managing ever-expanding volumes of data, ontology-based retrieval stands as a beacon of innovation, guiding the way towards more efficient, effective, and insightful information management practices..

Ontology

An ontology serves as a structured framework for representing knowledge within a specific domain. It essentially acts as a formalized blueprint, capturing concepts, their attributes, and the relationships between them. This structured representation facilitates communication and understanding, not just among humans but also between machines and computer systems.

In simpler terms, think of an ontology as a structured dictionary for a particular subject area.

It defines the terms used within that subject area and how they relate to each other. For example, in a medical ontology, terms like "disease," "symptom," and "treatment" would be defined along with their relationships (e.g., a disease can have various symptoms, and treatments can be prescribed for specific diseases).

Ontologies are crucial in various fields such as artificial intelligence, where they enable machines to understand and reason about the world. They also play a significant role in information retrieval, helping to categorize and organize data for more efficient search and retrieval processes.

Developing an ontology involves careful consideration of the domain it represents, as well as collaboration among domain experts to ensure its accuracy and completeness. Formal languages like OWL and RDF are often used to create ontologies, providing a standardized way to express complex relationships and constraints.

Overall, ontologies are essential tools for organizing, sharing, and reasoning about knowledge within specific domains, driving advancements in various fields ranging from healthcare and biology to computer science and beyond.

Protege

Protege stands out as a widely embraced, freely accessible, and open-source software solution tailored for the creation, modification, and administration of ontologies. Developed by Stanford University, Protege provides a user-friendly interface and a rich set of features for ontology development and knowledge modeling. Its fundamental purpose revolves around enabling the creation and upkeep of ontologies, which serve as structured representations of knowledge within particular domains.Protege boasts support for a range of ontology languages, such as OWL (Web Ontology Language), RDF (Resource Description Framework), and RDFS (RDF Schema), enhancing its adaptability across diverse application domains.Users can define classes, properties, and relationships between entities, as well as specify constraints and axioms to capture domain knowledge accurately.One of Protege's key functionalities is its support for executing queries and employing reasoning capabilities to infer new knowledge from ontologies. Users can formulate queries using languages like SPARQL and DL Queries, enabling sophisticated information retrieval and reasoning tasks.Overall, Protege plays a vital role in ontology engineering and knowledge representation, empowering researchers and practitioners to create, manage, and utilize ontologies effectively in various domains such as artificial intelligence, bioinformatics, semantic web technologies, and more. Its intuitive interface, support for multiple ontology languages, and extensive features make it a valuable tool for advancing research and innovation in knowledge-based systems.

Ontologies and formal representations of knowledge domains can be created and managed using the free and open-source software program, Protege.

Executing queries or employing reasoning abilities to infer new knowledge are required when retrieving data from an ontology.

Common techniques for retrieving information include:

1. SPARQL
2. DL Queries
3. Inference and Reasoning
4. Visual Investigation
5. Using the Ontology-Java, Python, and RDF libraries programmatically

Information Retrieval from an ontology using protege can be achieved using DL Queries.

Literature Survey

Many ontology development and query languages have been developed over the past few decades, and research on them is still progressing. [1] Selecting the appropriate ontology language for a particular scenario is crucial prior to developing an ontology-based system. Over recent years, numerous ontology languages have emerged, with most being machine interpretable as they are constructed upon the eXtensible Markup Language (XML).[2] Notable examples include the Resource Description Framework (RDF) and RDF Schema, the DARPA Agent Markup Language and the Ontology Inference Layer, as well as the Ontology Web Language (OWL) and its subsequent version, OWL2. Additionally, the DARPA Agent Markup Language coupled with the Ontology Inference Layer (DAML + OIL) stands out among prominent ontology languages. [3]

Semantic search has been present in the realm of information retrieval since at least the early 1980s, if not earlier.[4]

Certain strategies harness and capture vague and imprecise conceptualizations[5], as they rely on statistical methodologies that analyze the co-occurrence of concepts. Other IR approaches use linguistic algorithms [6] that are based on the structures and mechanisms of human language processing. However, these approaches often depend on thesauri and taxonomies, where conceptualizations tend to be shallow and lacking in depth, particularly concerning relations. This aspect is crucial for expressing user requirements and pinpointing solutions.

The traditional keyword-based IR model[7] has been modified to serve as the foundation for the core semantic search model .

Our approach encompasses the four primary processes of information retrieval systems: indexing, querying, searching, and ranking. In contrast to traditional keyword-based IR models, our method represents queries using an ontology-based query language, such as SPARQL[8]. Additionally, the external resources necessary for indexing and query processing include an ontology and its associated knowledge base..

A semantic annotation procedure is analogous to the indexing process[9]. In our ontology-based information retrieval model, the inverted index comprises semantic entities (meanings) that are linked to the documents in which they are found, as opposed to constructing an inverted index where keywords are associated with the articles where they appear.

Recently, area-wise ontology [10] has been used from a semantic perspective for data modeling and information gathering. The main objective of ontology-based information retrieval is to enhance the interaction between user queries and the retrieved data, aiming to provide precise answers that closely align with user preferences and requirements.

In a certain issue domain, an ontology often denotes a widespread, acknowledged, and comprehensive replica.The fundamental advantage of using a domain ontology is that it provides a copy of semantic data [11] along with knowledge of the domain it is associated with. To create lean strategies for discovering data, ontologies are used.

Ontologies[12] play a pivotal role in enhancing data management by fostering a shared understanding of information, articulating domain assumptions, and elevating data quality. When integrated within a relational database management system, an ontology transforms it into an ontology database. These databases can propagate assertions through triggers upon loading or unfold views during query execution, ensuring that user inquiries consider various constraints and specifications.

In domains like biomedicine, ontologies are extensively utilized to annotate data, paving the way for the development of decision-support systems tailored for clinical applications. Each business operates within its own lexicon, with specialized terms and jargon specific to its industry. For instance, a media company might refer to producers, writers, directors, and video masters, all of which constitute its unique language. This specialized language[13] is not only crucial for internal communication within organizations but also for engaging with consumers. It forms the basis upon which programmers and data scientists identify and analyze data. Ontologies serve as a powerful tool for describing and formalizing this language, enabling a structured approach to understanding and managing domain-specific terminology.

At its core, ontology embodies the "study of the names of things," emphasizing the systematic exploration and categorization of concepts[14] and their relationships within a given domain. By providing a standardized framework for representing knowledge, ontologies facilitate effective communication, knowledge sharing, and decision-making processes within organizations and across industries.

In the realm of information systems, the escalating complexity, compounded by exponential data growth and dynamic requirements, presents formidable challenges in conceptual data modeling[15]. To streamline this process and address the pressing need for efficiency, the integration of ontology-based knowledge into conceptual data modeling emerges as a compelling solution. Leveraging ontology not only facilitates the reuse of existing knowledge but also serves as a catalyst for reducing efforts, costs, and time associated with conceptual data modeling endeavors.

Formally establishing the viability of ontology transformation[16] into a conceptual data model is paramount, and this feasibility is substantiated through rigorous graph modeling techniques. Notably, the Cyc ontology stands out for its overarching objective of fostering knowledge utilization across diverse domains, encompassing a vast array of categories. With a knowledge base comprising millions of assertions, the Cyc ontology harnesses the power of its generality and employs efficient inference engines to derive new insights, effectively leveraging trillions of pieces of actionable knowledge.

By integrating ontology-driven approaches [17] into conceptual data modeling, organizations can navigate the complexities of modern information systems with greater agility and precision. This fusion of ontology and conceptual data modeling not only enhances the efficiency of model development but also enriches the resulting models with a deeper semantic understanding, thereby laying a robust foundation for informed decision-making and innovation in the digital era.

Methodology

Metadata Extraction: The methodology aims to extract metadata from databases utilized in generating ontologies. This step facilitates understanding the structure and relationships within the data.

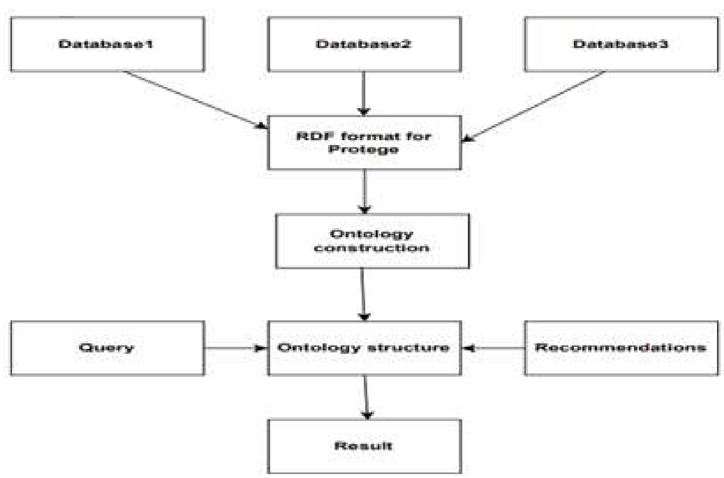

Figure 1: Architectural diagram of information retrieval using ontology

Source: Author

The data from the different databases are collected and made a unified dataset.

This dataset is converted into RDF format.

Ontology is constructed based on the RDF format

For this Ontology an Query is passed requesting the data and result is obtained.

Ontology Application: Utilizing the extracted metadata, the generated ontology is applied to enhance the efficiency of information retrieval. Ontologies aid in comprehending the interconnections between concepts across various databases.

Enhanced Retrieval Efficiency: The system aims to enhance the efficiency of existing information retrieval methods from distributed databases through semantic approaches. By leveraging ontologies, the retrieval process becomes more precise and streamlined, reducing the search space and improving retrieval accuracy.This proposed system offers several advantages:

Improved Accuracy: Ontologies enable a deeper understanding of concept relationships, leading to more accurate information retrieval queries.

Increased Efficiency: By leveraging ontologies, the system can reduce the amount of data that needs to be searched, thus improving query efficiency.

Enhanced Scalability: Utilizing ontologies facilitates the smooth integration of new databases into the system, thereby improving its scalability for future expansion.

Implementation

The different databases can first be integrated using Python's pandas library, with all the missing information filled in with the relevant values. The resultant database is then transformed into RDF format because the protege cannot accept files in other file formats. An ontology is created from this, and information is retrieved via DL Queries."Ontology is not just a buzzword; it's a pivotal tool poised to revolutionize our business landscape, both today and tomorrow. In our increasingly data-driven world, where decision-making processes are becoming automated and data streams from diverse sources, ontology holds immense promise. Its primary allure lies in its ability to address the complex challenges stemming from a lack of semantic understanding within datasets.

Numerous endeavors have been undertaken to harness ontology-based techniques for enhancing the query answering processes within databases and information systems. Advancements in technology now allow ontologies to dynamically rewrite user queries, thereby yielding more meaningful results that align with user intent.The popularity of ontologies stems from their capacity to establish a shared understanding of application domains, elucidating the meanings of terms and their interrelationships within a given context. This shared understanding not only fosters cohesion but also enhances the functionality of web search engines by imbuing web content with semantic significance.Traditional syntactic-based query processing techniques have proven inadequate in navigating the intricacies posed by data heterogeneity in format and structure. Today's databases and information systems demand a deeper level of knowledge about information sources to efficiently retrieve data and meet user expectations.In the quest for efficient query reformulation, establishing mappings between ontologies and underlying databases is paramount. These mappings serve as bridges, linking ontology concepts and relationships with database elements. Consequently, ontologies find widespread utility in information integration systems, where they play a vital role in mitigating data heterogeneity challenges and optimizing query processing across distributed sources."

D2R Mapping Process

The process of mapping a relational database to RDF (Resource Description Framework) using D2R begins with the installation and configuration of the D2R server, along with its requisite dependencies. Key among these dependencies is the Java MySQL connector, which facilitates communication between the D2R server and the MySQL database.

Once the environment is properly set up, the next step involves defining the mapping between the relational database schema and RDF using the D2R mapping tool. This mapping is crucial as it establishes the correspondence between elements of the database schema, such as tables, columns, and rows, and their counterparts in RDF, including entities, properties, and values.

The mapping process entails specifying how each database table, along with its associated columns and rows, is transformed into RDF triples. These triples consist of subject-predicate-object statements that represent the data in a semantic format compatible with RDF. By defining these mappings, the D2R mapping tool creates a bridge between the relational database and the RDF data model, facilitating the conversion of relational data into a graph-based representation.

One of the key advantages of the D2R mapping process is its ability to preserve the semantic relationships present in the original database schema. By accurately mapping database elements to RDF entities, properties, and values, the resulting RDF representation retains the inherent structure and semantics of the relational data. This preservation of semantic relationships enables seamless integration with ontologies, facilitating interoperability and knowledge discovery across heterogeneous data sources.

Furthermore, the RDF representation generated by the D2R mapping process adheres to the principles of Linked Data, making it compatible with the wider ecosystem of semantic web technologies. This compatibility opens up opportunities for data integration, federation, and enrichment, as RDF datasets can be interconnected and queried using standardized protocols and languages such as SPARQL (SPARQL Protocol and RDF Query Language).

In summary, the D2R mapping process provides a robust framework for converting relational databases into RDF, thereby enabling the integration of structured data into the semantic web. By defining mappings between database elements and RDF constructs, this process preserves semantic relationships and facilitates interoperability with ontologies. As organizations increasingly embrace the principles of Linked Data and the semantic web, the D2R mapping process serves as a valuable tool for unlocking the full potential of relational data in a semantic context.

Information Retrieval with SPARQL

Once the database has been mapped to RDF, users can retrieve information using SPARQL queries. SPARQL (SPARQL Protocol and RDF Query Language) emerges as a powerful and versatile query language, expressly designed for interrogating RDF (Resource Description Framework) data. This language facilitates the extraction of relevant information from RDF datasets by enabling users to craft intricate queries that navigate the intricate web of relationships and attributes encoded within the RDF graph.

At its core, SPARQL operates on the principle of pattern matching, where users formulate queries by specifying patterns that the RDF data must match. These patterns can encompass various criteria, including property values, relationships between entities, and hierarchical structures, allowing for nuanced and targeted information retrieval.

The syntax of SPARQL queries is intuitive yet expressive, empowering users to construct queries that precisely reflect their information needs. Queries typically consist of triple patterns, which describe the structure of the data to be retrieved. These triple patterns consist of subject-predicate-object triples, mirroring the structure of RDF statements and allowing users to navigate the graph by traversing edges between nodes.

SPARQL queries can range from simple searches for specific entities or properties to complex inquiries that involve multiple constraints and conditions. Users can employ a variety of operators and functions to refine their queries, such as filtering based on value comparisons, combining multiple conditions with logical operators, and aggregating results to compute statistics or summaries.

Moreover, SPARQL supports the concept of federated queries, allowing users to query multiple RDF datasets distributed across different endpoints. This capability enables seamless integration of data from diverse sources, enhancing the comprehensiveness and utility of information retrieval tasks.

Executing SPARQL queries involves submitting them to a SPARQL endpoint, which processes the queries against the RDF data and returns the results in a structured format, typically as a table or graph. The query engine optimizes query execution to ensure efficient performance, leveraging indexing and caching techniques to minimize latency and resource utilization.

One of the key advantages of SPARQL is its flexibility and scalability, making it suitable for a wide range of applications and use cases. Whether retrieving structured data for analytical purposes, performing semantic searches to discover related concepts, or integrating heterogeneous data sources for knowledge discovery, SPARQL offers a robust and adaptable solution.

In summary, SPARQL stands as a cornerstone of RDF-based information retrieval, providing users with a powerful means to navigate and extract knowledge from RDF datasets. Its expressive syntax, support for complex queries, and ability to integrate distributed data sources make it an invaluable tool for exploring and harnessing the wealth of information encoded in RDF. As the volume and complexity of RDF data continue to grow, SPARQL remains essential for unlocking insights and driving innovation in the realm of semantic data management and analysis.

Results and Discussion

D2RMap Results:

Given a relational database containing tables for "Employees" and "Departments":

D2RMap will map these tables to corresponding classes in the ontology.

Each employee's attributes (e.g., name, department ID) will be mapped to properties of the "Employee" class.

Relationships between employees and departments (e.g., employee belongs to a department) will be mapped to object properties.

The resulting RDF triples will represent individual employees and their relationships with departments.

Information Retrieval using SPARQL:

Suppose you want to retrieve information about employees who belong to the "Marketing" department:

You would write a SPARQL query to select individuals (employees) who have a relationship with the "Marketing" department.

The query might look like this:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX ex: <http://example.org/ontology#>

SELECT ?employee

WHERE {

?employee rdf:type ex:Employee .

?employee ex:belongsTo ex: Marketing Department .

}

Running this query would return a list of employees who belong to the Marketing department.

Integration with Protege

You can execute SPARQL queries directly within Protege's interface.

After loading the RDF data generated by D2RMap into Protege:

You can use Protege's SPARQL query plugin to write and execute queries.

Results will be displayed within Protege, allowing you to explore and analyze the retrieved information alongside your ontology.

Example Result:

Suppose the query returns the following results:

```
+-----------------+
|    employee     |
+-----------------+
| ex:Ashwin       |
| ex:Bobbin       |
+-----------------+
```

This indicates that both Ashwin and Bobbin are employees belonging to the Marketing department, as per the RDF data.

These results demonstrate how D2RMap can map relational database data to RDF triples, and how SPARQL queries can be used to retrieve specific information from this RDF data within Protege.

Table1: Comparison of different approaches

S. No	Method Adapted	Performance	Conclusion
1	Traditional method	Takes more time	Not used frequently
2	SQL Query	Average	Used in simple databases
3	SPARQL Method	Better	Preferred more especially when we have large databases

Source: Author

Conclusion

In conclusion, ontology-based retrieval using D2R mapping and SPARQL presents a robust framework for integrating relational databases with ontologies. This methodology offers a streamlined approach to accessing structured data, enabling organizations to leverage the full potential of their information repositories. By converting database schemas into RDF format and executing SPARQL queries, users can retrieve precise and relevant information tailored to their needs. The user-friendly interfaces further enhance accessibility, allowing for intuitive interaction with the retrieved data. As data volumes continue to grow, the ability to harness the semantic relationships encoded in ontologies becomes increasingly valuable. Ontology-based retrieval serves as a cornerstone for data-driven decision-making, facilitating deeper insights and informed actions. By bridging the gap between relational databases and ontologies, this approach empowers organizations to navigate complex datasets with agility and precision, ultimately driving innovation and competitive advantage.

Future trends

Indeed, while ontology tools have revolutionized knowledge representation and management, they also encounter several challenges that hinder their efficacy and widespread adoption. One primary challenge lies in enhancing the expressive power and scalability of these tools to handle large knowledge bases and complex reasoning tasks effectively. As datasets grow in size and complexity, ontology tools must evolve to support efficient querying, matching, and inference processes.Furthermore, there is a demand for ontology tools to support high-level languages, modularity, and visualization capabilities. High-level languages enable users to express intricate domain concepts and relationships more intuitively, while modularity facilitates the development and maintenance of large ontologies by breaking them into smaller, manageable components. Visualization tools aid in comprehending the structure and content of ontologies, making them more accessible to users.Despite the proliferation of ontology tools, interoperability and database storage remain significant challenges. Many tools operate within isolated environments, limiting their ability to exchange ontological data seamlessly with other systems. Additionally, the integration of ontologies with databases is often cumbersome, leading to inefficiencies in data storage and retrieval.

Reference

1. Lu, L., & Li, M. (2023). Development of a virtual interactive system for Dahua Lou loom using knowledge ontology-driven technology. Heritage Science, 11(178).
2. Aminu, E. F., Oyefolahan, I. O., Abdullahi, M. B., & Salaudeen, M. T. (2022). MaCOnto: An ontology for maize crops based on soils, fertilizers, and irrigation knowledge. Intelligent Systems with Applications.
3. Jain, S., Seeja, K. R., & Jindal, R. (2021). A fuzzy ontology framework for information retrieval with semantic query expansion. International Journal of Information Management Data Insights, April.
4. Deepak, G., Gulzar, Z., & Leema, A. A. (2021). An intelligent system for modeling and evaluating domain ontologies in Crystallography. Computers and Electrical Engineering.
5. Mendonçaa, M., Perozoa, N., & Aguilarb, J. (2020). Ontological emergence scheme in self-organized and emerging systems. Advanced Engineering Informatics, 44(101045).
6. Munir, K., & Anjum, M. S. (2017). Utilization of ontologies for effective knowledge modeling and information retrieval.
7. Abbes, H., & Gargouri, F. (2016). Big Data Integration: A Mono DB Database and Modular Ontologies based Approach. Proceedings of the 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems.
8. Gorskis, H., Aleksejeva, L., & Polaka, I. (2016). Database analysis for ontology learning. Proceedings of the 12th International Conference on Application of Fuzzy Systems and Soft Computing.
9. Santipantakis, G., Kotis, K., & Vouros, G. A. (2017). Ontology-Based Distributed framework for Accessing Integration and Reasoning with data in disparate sources.
10. Annane, A., Bellahsene, Z., Azouaou, F., & Jonquet, C. (2018). Constructing an effective background knowledge resource to enhance ontology matching.
11. Mandankar, M., Chandak, M. B., & Chavhan, N. (2015). Review of Information Retrieval Systems and Machine Translation. Proceedings of the International Conference on Information Security and Privacy.
12. Gupta, R., Halevy, A., Wang, X., Whang, S. E., & Wu, F. (2014). Biperpedia: An ontology for search applications.
13. Hassen, F., & Amel, G. T. (2017). A synchronous indexing technique for full text retrieval in distributed databases. Proceedings of the International Conference on Knowledge Based and Intelligent Information and Engineering Systems.
14. Konys, A. (2018). Systematizing knowledge for ontology learning methods. Proceedings of the 22nd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems.
15. Ibanescu, L., Buche, P., Dervaux, S., Touhami, R., & Dibie-Barthelemy, J. (2016). Evolution of ontologies for experimental data integration systems. International Journal of Metadata, Semantics and Ontologies, 11(4), 231-242.
16. Kranthi Kumar.R,Dr. B.Padmaja ani(2022). Ontologies' importance in efficient knowledge modelling for information retrieval: A Survey. The International journal of analytical and experimental modal analysis
17. Youn, S., & McLeod, D. (2006). Development Tools for Ontology-Based Knowledge Management. Retrieved from http://research.create.usc.edu/nonpublished_reports/100.