SOFT COMPUTING APPROACHES TO DOMAIN SPECIFIC INFORMATION RETRIEVAL IN THE SEMANTIC WEB

Thesis submitted to
University of Calicut in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY
In
COMPUTER SCIENCE

Under the Faculty of Science

By

RESHMA P. K.

Under the Guidance of

Dr. LAJISH V. L.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF CALICUT
KERALA, INDIA -673635

DECEMBER 2020

Dr. LAJISH. V.L
Assistant Professor

UNIVERSITY OF CALICUT
(Department of Computer Science)

Grams : UNICAL
Fax : 0494-400269
☎ : 0494-24017325
9495793094
E-mail : lajish@uoc.ac.in
Calicut University. P.O,
Pin: 673 635
KERALA – INDIA

Date:19-07-2021

CERTIFICATE

This is to certify that the correction/modification suggested by both the adjudicators have made in the Ph.D Thesis entitled "**Soft Computing Approaches to Domain Specific Information Retrieval in the Semantic Web**" submitted by **Ms. Reshma P. K.,** Research Scholar, Department of Computer Science, University of Calicut. Hence the thesis is submitted as such to the University of Calicut with reference to the letter number 75352/RESEARCH-C-ASST-1/2021/Admn Dated 04.07.2021.

Calicut University
19-07-2021

Dr. Lajish V. L.
(Supervising Teacher)

Dr. LAJISH. V.L.
Assistant Professor
Department of Computer Science
University of Calicut, Kerala-673 635

DEPARTMENT OF COMPUTER SCIENCE
Date..............
★ UNIVERSITY OF CALICUT ★

UNIVERSITY OF CALICUT
DEPARTMENT OF COMPUTER SCIENCE

Dr.Lajish V. L.
Assistant Professor

Calicut University (P.O)
Kerala, India – 673635

CERTIFICATE

This is to certify that the thesis entitled "**SOFT COMPUTING APPROACHES TO DOMAIN SPECIFIC INFORMATION RETRIEVAL IN THE SEMANTIC WEB**" is a report of the original work carried out by Ms. Reshma P.K. under my supervision and guidance in the Department of Computer Science, University of Calicut, Kerala and that no part thereof has been presented for the award of any other degree.

Calicut University,
December 2020

Dr. LAJISH. V. L.
(Supervising Guide)
Faculty, Department of
Computer Science &
Director, Calicut University Computer Centre
University of Calicut, Kerala – 673635

DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the supervision of Dr. Lajish V. L., Department of Computer Science, University of Calicut, Kerala and that no part thereof has been presented for the award of any other degree.

Calicut University, **Reshma P. K.**
3rd December 2020

Acknowledgments

This thesis is the outcome of the journey I have traveled for around seven years. This has been kept on track and seen to completion due the support and encouragement of many people. It's a happy thing to thank those unforgettable persons who made this thesis possible.

First of all, I would like to thank my research supervisor Dr. Lajish V L, Faculty, Dept. Computer Science, University of Calicut, for his active, concise and encouraging supervision. He was the strongest pillar for me during the entire period of the research. His positive attitude, friendliness, and confidence in my research inspired me and gave me confidence. His careful reviews contributed immensely to the production of the research papers and this thesis. The understanding, patience, kindness, freedom, and support I could enjoy, made my research tenure a memorable one.

I am deeply indebted to Dr. P.Nagbhushan, Professor & Director, IIT Allahabad and Dr. Rajesh R, Assoc. Professor & Head, Department of Computer Science, Central University of Kerala for giving insight, motivation, and direction to my research studies and helping me in many ways in designing the work pattern.

I remember with gratitude the team members and mentors at IIIT Hyderabad during IASNLP-2017 for the insightful discussions which helped a lot in my work.

I extend my gratitude towards The Management, Dr. Ajitha V, Principal, Dr. Shijo.M. Joseph, Head of the Dept., all my colleagues and students of Mahatma Gandhi College, Iritty for cooperation in several ways during the tenure of this research. I express my heartfelt thanks to Dr. Reji Paikkatt, Head, Department of English, Mahatma Gandhi College, Iritty, for the help in terms of language proficiency in compiling the thesis.

I am grateful to Mrs. Manjula K A, Assistant Professor & Head, Department of Computer Science, University of Calicut for her help and support for the realization of the research. I am also thankful to all the staff members, both teaching and nonteaching and the students at the Department of Computer Science for helping with required facilities whenever needed.

I thank my colleagues Dr. Vivek P, Dr. Benson C C, Dr. Sandesh E Pa, Mr. Baiju, K B, Ms. Habeebath K P, Mr. Anoop K, Ms Manjary P Gangan and the M.Phil Scholars Ms. Jyothi, Ms. Hima, and Ms. Suharshala at Dept. Computer Science, University of Calicut for the extremely enjoyable and productive cooperation while researching and shaping my work.

The support I get from my parents and in-laws cannot be expressed in words as they are always there for everything. I also remember with love and gratitude the help, support, and encouragement I could enjoy from my siblings, family members and friends.

Special thanks to my better half Dr. Aneesh Kumar and our kids Niranjana & Devaj for their unconditional love and cooperation, without which I would not have come this far.

And I bow my head before that superpower for giving me the strength, knowledge, ability, and opportunity to undertake this research study and to preserve and complete it satisfactorily.

Reshma. P.K

Dedicated

To my life coaches … My parents

ABSTRACT

This work proposes a framework to facilitate Information Retrieval (IR) based on domain ontology. Usually searching is done with keywords. But if the exact keywords to be used for a search are not known, users may follow links or browse the pages to get the relevant information. To improve this situation, a semantic web is introduced which uses ontologies. Ontology based IR can provide better solutions. Ontology is a vision of shared conceptualization, represented by classes, objects and properties. The retrieval of information can then be conducted by matching the semantics of the user query with that of documents. Incorporating fuzzy provides the easy handling of imprecise and ambiguous terms present in the query. The proposed work uses query and document similarity indexing and ontology based information retrieval on crisp and fuzzy ontology developed for University domain.

The data sets used for this study consist of University and Government orders issued for the sake of students and staff of the University and related institutions and the user queries in natural language. The domain considered for the current study is a State University in Kerala, India and, in particular, the information related to Administration, Academics, Examination, Finance, and Planning & Development. The experimental

arrangement of proposed work is operated on the enquiries on different services offered by the Departments/Centers/ Branches under the University and the general enquiries related to the University administration system. The query dataset generated for this study is named Short Queries for University Services in English (SQUSE) and the corresponding document data set used for this study is named University Document Data Set in English (UDDSE). These are made available publically for research purposes. (https://dcs.uoc.ac.in/#).

In the first phase of this study the query and documents are represented as vector space models and the similarity is measured. Cosine and Jaccard similarity measures are used for the analysis. Fuzzy logic is incorporated to deal with the imprecise and ambiguous terms present in the query. As a result, the best matched documents for a user query are identified in ranked order.

In the second phase translation of a natural language query to corresponding SPARQL query is considered in detail. The ontology in semantic web is represented as Resource Description Framework (RDF) and SPARQL as the language to query RDF. SPARQL stands for the recursive acronym, SPARQL Protocol and RDF Query Language. The natural language queries generated by users for the university related information retrieval are translated to corresponding SPARQL queries and the accuracy is tested.

In the last phase an information retrieval model based on ontology for the University domain is developed. Ontology is developed within the five sub domains of the University such as Academics, Administration, Examination, Finance and Planning & Development. Both crisp and fuzzy ontology based systems are developed and compared as part of this study.

From the experimental results, it is concluded that the developed ontology based semantic information retrieval system can be effectively used for the University domain.

CONTENTS

*I have a dream for the Web (in which computers) become capable of analyzing all the data on the Web— the content, links, and transactions between people and computers. A "**Semantic Web**", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "**intelligent agents**" people have touted for ages will finally materialize.*

- Tim Berners-Lee

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACP	Average Conditional Probability
BM	Best Matching
BoW	Bag of Words
CBR	Case Based Reasoning
D2V	Document to Vector
DAML	DARPA Agent Markup Language
DBMS	Data Base Management System
DBoW	Distributed Bag of Words
DL	Description Logic
DM	Distributed Memory
DTD	Document Type Definition
E-R	Entity Relationship
FBHSM	Fuzzy logic Based Hybrid Similarity Measure
FIRE	Forum for Information Retrieval Evaluation
FJCSM	Fuzzy Jaccard Cosine Similarity Measure
FO	Fuzzy Ontology
GA	Genetic Algorithm
GloVe	Global Vectors for word representation
HTML	Hyper Text Markup Language
IR	Information Retrieval
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing
MRSE	Multi keyword Ranked Search over Encrypted Data
NLP	Natural Language Processing
NLTK	Natural Language Toolkit

NMI	Normalized Mutual Information
OBDA	Ontology Based Data Access
OBO	Open Biological and Biomedical Ontology Foundry
OCML	Operational Conceptual Modelling Language
OIL	Ontology Inference Layer
OKBC	Open Knowledge Base Connectivity
OWL	Web Ontology Language
PAL	Protégé Axiom Language
PDF	Portable Document Format
POS	Part of Speech
QC	Query Classification
RDF	Resource Description Framework
SIRSU	Semantic information Retrieval System for University
SPARQL	SPARQL Protocol And RDF Query Language
Spyder	Scientific PYthon Development EnviRonment
SQUSE	Short Queries for University Services in English
SWETO	Semantic Web Technology Evaluation Ontology
TF-IDF	Term Frequency- Inverse Document Frequency
TREC	Text Retrieval Conference
UDDSE	University Document Data Set in English
UML	Unified Modeling Language
VQS	Virtual Query System
VSM	Vector Space Model
W2V	Word to Vector
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extended Markup Language
XSLT	Extensible Stylesheet Language Transformations

Chapter 1
Introduction

1.1. Background

The escalations in the volume of information and its unstructured nature have made the searching to find relevant information more difficult, complicated and time consuming. Hence strategies and software systems that can access relevant information quickly and efficiently should be designed and perfected as we move on in the days of information explosion, especially in the context of knowledge society. In the previous years, web technologies have become the magic solution for information spreading and service providing across the globe, which is a critical factor in encompassing computers in the common place. Nowadays, the World Wide Web or WWW is a global platform where consumers and providers meet to exchange data, information, opinion, and even money.

One of the most challenging issues in today's world of digital information is to incorporate domain knowledge to retrieve more relevant information from a collection based on a query given by the user. The design of an explicit representation of terms and their relationships defined as ontology can be used to expand the user requests and to retrieve the relevant documents from a corpus. Searching the World Wide Web has become more challenging now due to the rapid growth of information. Every document contains some valuable knowledge about a particular domain within it. The World Wide Web plays a crucial role in information sharing for education, research, business, etc. But a large number of useful information available over the web is unstructured, imprecise, and fragmented formats. The information may include reports, scientific papers, news, emails, etc. Ontology is a set of concepts in a particular domain which shows the associated properties and relations between them. The ontology can be considered as the main source to understand the textual information contained within the documents. Ontology-based information retrieval matches the relevance of a user-generated query against a knowledge-base. Ontologies are often considered a new source of semantics and interoperability in all artificially smart systems. An exponential increase in unstructured data on the web has made the automated acquisition of ontology from unstructured text a most prominent research area. The first applications of web-based ontology are successfully placed in areas like semantic search, information integration, and web community portals.

There are different challenges related to traditional Information Retrieval (IR). Some of them are:

- Traditional IR techniques have become insufficient to handle enormous text documents.
- Most of the search results consist of a set of web pages containing both relevant and non-relevant information. Sometimes the high-rank score is assigned to non-relevant pages.
- Finding out the relevant document or information from this collection is a tiresome task for the user.

The semantic web introduced by Tim Berners-Lee in 2001 is an extension of the current web where semantic knowledge is attached to complete information. The semantic web is an extension of the current web in which information is provided in a machine-processable format [1]. If the information is machine-processable, perfect in-depth searching with reduced user time and effort is possible. Hence new techniques for knowledge management and agent-based processing have to be formulated.

In a semantic web, the concept of ontology is used to search by the contextual meaning of input query as an alternative of keyword matching. From the literature, it is clear that a scheme needs to be devised that can provide an easy interface for complex natural language query which can retrieve relevant domain-specific information from the ontology.

Today we are faced with giant, centralized repositories of information. All the things need to be entered into Google or Yelp because those are the most common services used, and usually the only one tool human beings will bother to check. Ontology provides a general outline of a domain for which the semantics of concepts is machine understandable. Ontology works with metadata and assists the search engine to do the functionality of semantic matching along with interchange and incorporation of knowledge.

An ontology study what exists in a domain of interest, and it is a computational artifact that encodes knowledge about this domain in a machine-processable form to make it available for information systems. There are different definitions for ontology according to the various contexts of applications, but in the semantic community, it is defined as an explicit formal specification of a shared conceptualization of a domain of interest [2].

This work introduces a framework to facilitate information retrieval based on domain ontology. The ontology used in the system is a conceptual yet executable model of an application domain. The semantics of the user query with that of documents can be matched to retrieve the correct document. The challenge we face today is to make the computer understand the language or logic just as the way humans understand. This is incorporated by the use of fuzzy logic, which provides the flexibility needed by handling the vagueness or uncertainties to a certain extent. It is made machine-interpretable by the knowledge representation techniques and thus can be used in applications to base decisions on reasoning about domain knowledge.

Semantic search in the Internet and private or semi-private networks can be implemented with the help of ontologies. Semantic search should employ semantic technologies to support human or automated agents in the process of finding information sources from a given search context that satisfies the given information need. In order to answer the information needed precisely and comprehensively, the search system should get an interpretation of the information needed and should employ semantic metadata or background information regarding the domain of the available information.

1.2. Motivation

Usually, the terms used in a document have multiple meanings. Thus, providing ontology helps to formulate more effective retrieval according to the user's requirements. Another essential objective is to retrieve a relevant concept from the corpus. The ontology generates new instances from the text and results in the semantic annotation of the document. The algorithm assumes the semantic similarity between the documents. The main goal of information extraction is to retrieve the relevant information from the document and to create an instance of ontology. Most of the content on the web is not semantically annotated and it exists in the form of raw texts [1] [3]. Hence the present research aims to use a Semantic Web framework based on ontology in order to provide a knowledge-sharing arrangement with greater ease and facility.

For the semantic web, the information is to be presented in a hierarchical order. An ontology represents the concept and relationship among the concepts. The document can be labeled using ontology. The terms in a document can be described as a concept in a specific domain. The various terms and relationships among them can be used to index documents.

Ontology provides a knowledge sharing framework to represent and share domain knowledge. Often ontologies are visualized and thought of as semantic networks that display interrelated conceptual nodes. This semantic network can be used to identify some essential characteristics that are common to most ontologies used in information systems.

In this work, how the effectiveness of the information retrieval can be enhanced by using the ontology-based semantic web and vector-based similarity indexing is studied and the findings are presented. Fuzzy logic is also incorporated to deal with the vagueness in query processing. The classical Cosine and Jaccard coefficients of similarity measures are fuzzified with specific weights and defined the ruleset. Hence the better similarity measures are obtained to match the documents with the given query.

The primary goal of this research work is to design and implement an efficient method that can extract relevant features from a collection of textual documents going from complete plain textual data to semi-structured data. It is essential to understand that classical Information Retrieval (IR) is a well-researched and established discipline. But a semantic-enhanced information retrieval approach can improve the results of traditional IR. The techniques developed in conventional Information Retrieval research are also useful in semantic-enhanced methods.

The key challenge of this work is to move from keyword-based retrieval to concept-based retrieval of information by utilizing Ontologies. Here we need to identify the concepts in information present and queries given by users. Representing this in the information retrieval system helps to process the query in a better way, by using the knowledge about the relations among concepts captured by ontologies.

Based on some of the softness of conventional IR techniques, the motivation towards a soft computing approach for a semantic web-based IR framework is formulated. The goals of the current research presented in this thesis are as follows:

- To develop ontology-based information extraction.
- To develop an interface for semantic search.
- To develop better query – document similarity ranking based on vectorization and word embedding.
- To find the best matching document for a given query using fuzzy logic in similarity measures.
- To study the effect of combining fuzzy representation in the ontology framework.

At first glimpse, ontologies seem to be very analogous to other kinds of conceptual models used, especially in Computer Science. The typical constructs, notions such as interrelations, instantiations, can be found in Unified Modeling Language (UML) class diagrams or Entity-Relationship (E-R) diagrams for the specification of information systems or database schema respectively. But in terms of purpose and usage, ontology is different from other conceptual models. Conceptual models are used to construct a technical system, while the ontologies are descriptively capturing the knowledge observed to characterize a domain.

1.3. Outline of the Thesis

Chapter 2 is intended to provide necessary background knowledge for the following chapters. This chapter reviews the Ontology-based Information Retrieval in the Semantic Web. The review on Vector Space Model for Information Retrieval is presented to recognize the scope of merging the vector space model with ontology for better retrieval of information. To deal with vagueness contained within the query and retrieved documents, soft computing approaches are needed to be incorporated. By studying the various strategies, it is observed that Fuzzy logic is best paired with Ontology and document representation to improve the results of information retrieval. So a detailed review of Soft computing Approaches to IR in semantic web with fuzzy logic is also carried out here.

Chapter 3 deals with the preparation of the dataset used for the current study. The data set consists of both University and Government orders issued for the sake of students and staff of the University and related institutions. The user queries are framed in natural language. The domain considered for the conduct of the present study is a state University system and, in particular, the information related to various sections of the University, including Administration, Academics, Examination, Finance, and Planning & Development. The experimental setup for the conduct of the proposed work is operated based on the enquiries pertaining to different departments under the

University and the general enquiries related to the University system. The dataset includes a Document set and a corresponding Query set. Various text preprocessing methods applied to the document and query samples within the dataset are also described in this chapter.

Chapter 4 discusses the implementation of TF-IDF and Word embedding for document and query similarity indexing. A study on Document and Query Similarity is conducted with TF-IDF, Word2Vec, and Doc2Vec methods, and the results are analyzed. A survey on GloVe is also performed, which reveals the means to improve the results. After discussing the features of each measure, a hybridization of similarity measures with fuzzy logic is done to enhance the performance of similarity measures, so that the best match can be found out.

Chapter 5 proposes a Semantic Web model that can be effectively used in the university domain for Information Retrieval. To facilitate the functionality of semantic matching on the search engine, ontology works on metadata in conjunction with the exchange and integration of knowledge. Development of Ontology for University Domain consists of the process of conceptualization, grouping, and interlinking of the descriptors, and organizing the Keywords/Descriptors file. Protégé is used as a tool for ontology development. The proposed system architecture is also presented with detailed functional modules, as shown in figure 1.1.

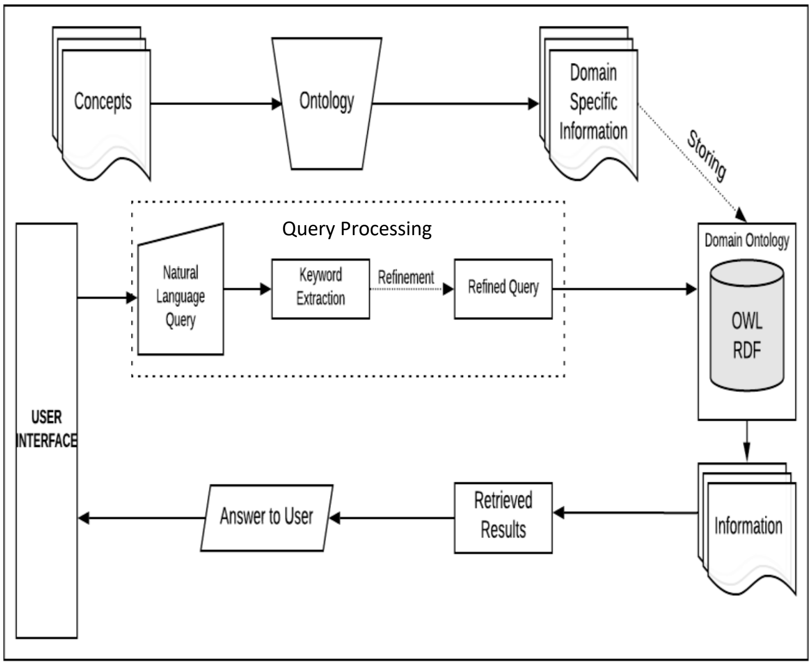


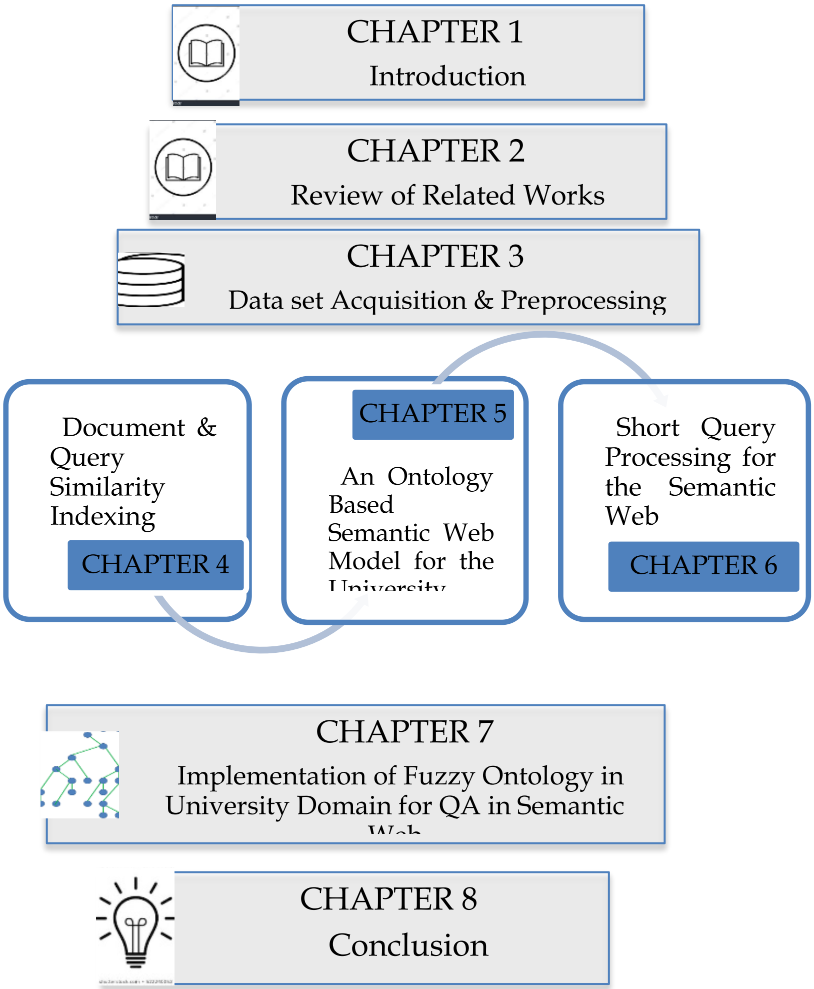
Figure 1.1 Architecture of the Proposed Information Retrieval System for the University domain

This work aims to process the query, which is in natural language. Chapter 6 deals with the implementation of a short query processing system that can be used in the Semantic Web. For this, preprocessing is carried out as the first phase, and then the queries are classified into five different predefined categories identified earlier. To deal with the synonyms of words and phrases, WordNet is used. The natural language queries are then converted into SPARQL, which is specific to the ontology. The converted queries are fired to the document ontology, and the results are tabulated and analyzed in detail.

This work aims to combine the methods of ontology and information retrieval for the semantic web by trying to explore the relevant information for the user needs. The challenges of implementing domain knowledge into the information retrieval system consist of designing a retrieval model, building an ontology, and identifying the relevant documents by the ontology. The domain considered for the proposed study is University and in particular the information related to Academics, Administration, Examination, Finance and Planning & Development. The data set considered for this study consists of 2800 documents, which are University orders and 7500 queries. The document ontology is created with the collected documents, and the converted natural query is fired for the information. The results are analyzed and compared.

An attempt to generate a fuzzy ontology for the domain of the collected document is described in Chapter 7. A framework for the Fuzzy OWL for the University domain is proposed for the constructed Ontology. The concepts of the documents are represented as classes and their associations in the corresponding ontology. Fuzzy logic is integrated into the ontology to deal with vague or incomplete information. Typically, fuzzy ontology is constructed using a predetermined concept hierarchy with the fuzzy OWL plugin used with the Protégé tool.

Finally, Chapter 8 concludes the thesis with an analysis of the works carried out as part of the study and suggests a few areas for future research. The complete outline of the chapter organization and research work reported in this thesis is shown in Figure 1.2.



CHAPTER 1
Introduction
CHAPTER 2
Review of Related Works
CHAPTER 3
Data set Acquisition & Preprocessing
Document & Query Similarity Indexing
CHAPTER 4
CHAPTER 5
An Ontology Based Semantic Web Model for the University
Short Query Processing for the Semantic Web
CHAPTER 6
CHAPTER 7
Implementation of Fuzzy Ontology in University Domain for QA in Semantic Web
CHAPTER 8
Conclusion

Figure 1.2: Block diagram of the chapter organization

Chapter 2
Review of Related Works

2.1. Introduction

Textual information retrieval is an empirically defined problem centered on expression or keywords extracted from documents and working as the foundation of both document and query representations. But, the scalability of keywords may vary. Some phrases will be more general. Hence some may refer to general topics while some others have more specific descriptors. Sometimes there is a possibility to have fewer keywords describing the broad themes of a document to be present in the documents. However, they are suitable for classifying the document and mentioning its content. Recently several Information Retrieval (IR) approaches have been developed based on keywords to describe the topical content and structure of documents [4].

An IR model specifies the details of the document representation, the query representation, and the retrieval functionality [4]. The fundamental IR models can be classified into Boolean, vector, probabilistic, and inference network model [5] [4].

This study is made for concept-based information retrieval, which is intended for retrieving relevant documents based on their meaning rather than their keywords. The critical concept of conceptual IR is that the purpose of the text be determined by the conceptual relationships to objects in the world rather than to linguistic relations found in the text [6].

There are IR models that are based on concepts referred to as the semantic web, which is considered to be the next hike of the Internet. The Semantic Web alters the document-oriented web to a data-oriented web enriched with semantics embedded as metadata. This change in perspective towards the web results in several benefits for the vast amount of data-intensive industries that are bound to the web and its related applications.

The concept of Semantic Web has become a vital instrument in varied fields since its introduction in 2001 in the seminal paper by Berners-Lee *et al.*, [7]. It allows us to design links within and across the data as it functions at the data level and hence is extra granular than the level linking in the traditional web. The Semantic Web has its role at the data level with a well-specified meaning, which makes this methodology extensively applicable throughout diverse domains with seamless integration with a variety of technologies.

According to the W3C, "The Semantic Web offers a common framework that permits data to be shared and reused through application, enterprise, and community boundaries" [8] [7]. The Semantic Web is, therefore, regarded as an integrator across different content, information applications, and systems [9].

The Semantic Web originated to address several drawbacks of the traditional document-oriented web. Primarily, the documents which contain unstructured text are indexed and retrieved using text matching algorithms that do not take the semantics of the document into account [10]. Irrelevant documents are also included in the search result with this approach. Secondly, the websites produce content based on structured backend data; thus, the underlying data remains hidden. Hence it is not available for consumption directly by machines [11]. Further, the most crucial factor is that the current web does not incorporate semantics as metadata.

The use of ontologies is at the heart of all Semantic Web applications. A universally agreed definition of ontology is: 'ontology is an explicit and formal specification of a conceptualization of a domain of interest' [12].

Ontology is a systematic description of knowledge within a domain as a collection of concepts and the relationships between them. To enable such a description, a formal specification of the components, like individuals which are instances or objects, classes, attributes, and relations as well as restrictions, rules, and axioms, is required [13]. Thus ontologies can add new knowledge about the domain. The domain knowledge consists of the concept and properties of a domain which users generally searched for and therefore specified in the ontology.

The rest of this chapter is organized as follows. The next section 2.2 presents a summary of the research works reported in various methods of information retrieval, especially in the ontology. A review and comparison of ontology development tools for semantic information retrieval is presented in sections 2.3. Section 2.4 gives a review on text preprocessing techniques. Section 2.5 describes different strategies adopted in vector-based similarity indexing, and 2.6 shows a discussion on short query processing. Tools and techniques used for developing and working with fuzzy ontology is described in section 2.7. Finally Section 2.8 draws the concluding remarks.

2.2. Review on Semantic Information Retrieval

Theoretically, any information can be the object of information retrieval. Different types of information can be retrieved by the same information retrieval system also. But in most of the information retrieval systems, the contents handled are text documents, and the retrieval system can be considered as text retrieval or document retrieval.

The perfect benchmark for text retrieval was developed with the name Text Retrieval Conference (TREC) [14]. Still, it continues as one of the best benchmarking systems for text and document retrieval applications. The TREC gives better results, but annotation is tedious.

Semantic web refers to a web with a meaning. It describes things in a way that computers can understand. The next generation intelligent web called the semantic web offers users the ability to work on shared meaningful knowledge representations on the web. Semantic Web creates an application which will make web content useful to computers, that intends to support machine-processing capabilities that will automate web applications and services. Thus semantic web can be understood by computers, doing the searching, aggregating, and combining the results without human interference. Software programs called agents will perform various tasks by communicating with other agents and seeking information from web resources [15]. The Semantic Web is a vision of having data on the web defined and linked in a way that machines can use it not just for display purposes, but for automation, integration, and reuse of data across various applications. It is a concept of how computers, people, and the web can work together more effectively than is possible now [16]. The semantic web architecture relative to the current web can be shown as in Figure 2.1. In this architecture, the existing web uses the UNICODE characters for data representation and cryptographic techniques for data security. Here the semantic part is enabled by a stack of developing languages such as SPARQL for querying, OWL for ontology representation in the Resource Description Framework.

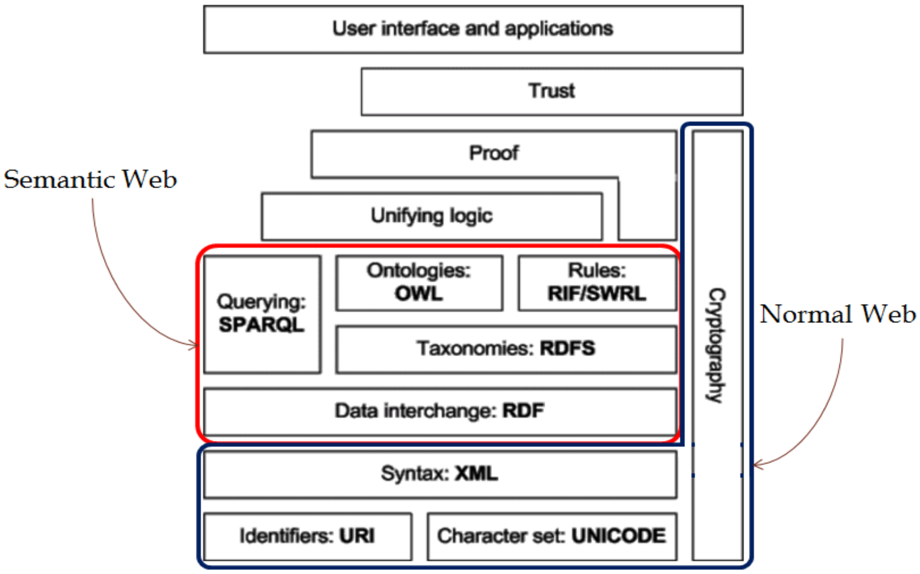


Figure 2.1: Semantic web architecture

(Source: http://semanticsage.blogspot.in)

Solyu *et al.*, presented a new multi-paradigm and ontology-based Visual Query System, named Optique VQS, for the end-users who are having no technical knowledge and skills [17]. It was built on a powerful Ontology-Based Data Access (OBDA) framework and had a flexible and extensible architecture that allowed linking and organizing different representation and communication paradigms. The results of the usability experiment suggested that Optique VQS provided a decent level of expressivity and high usability.

Muller *et al.*, proposed that either the generic concepts can be mapped between two ontologies, or the generic concepts be entirely discarded [18]. The mapping of the generic concepts links shared the concepts without unnecessary noise. At the same time, the removal of the generic concepts resulted in an information loss related to all the variables and the associated values to terms from the interoperable ontologies listed at the Open Biological and Biomedical Ontology Foundry (OBO Foundry). OBO foundry topologies provided the benefit of extensive coverage, but it could also be selectively imported to create application ontology such as the EuPath [19] ontology. When the existing terms were not available for mapping, the new ones were created for introduction into the source ontology or just placed in the application ontology.

Bansal *et al.*, designed a novel approach of ontology based information retrieval system for classified ads [20]. The ads database was taken for the house data from the Hindi newspaper. Various features were extracted using the ontology-based rules which have not been dealt with in the past. The results obtained were found to be quite advantageous and proved the effectiveness of the proposed algorithm.

Bourgonje *et al.*, addressed the issue of combining the Natural Language Processing, Information Retrieval, and Machine Translation procedures into a system that enabled knowledge workers to explore a collection of documents intuitively and efficiently [21]. The goal is to support knowledge workers in their daily work, i.e., to automate or to semi-automate routine processes that the human experts are generally required to do intellectually and without tool support. The study focused on combining individual components and linking the methods, rather than improving the production of the individual state of the art procedures. The information enclosed in multiple documents were combined and presented in a form that allowed the knowledge workers to view the documents. A locational reference deriving model with its associated prototype preprocessing layer is presented which has the potential to promote critical spatial thinking by expanding the data source options. The integrity of the model was restricted by the credibility of the data retrieval sources, and limited to handling the vector data.

Vigneshwari *et al.,* utilized WordNet [23] and Semantic Web Technology Evaluation Ontology (SWETO) [24] for the generation of data sets [22]. They performed the cross ontology with hashing, without hashing and also with both semantic annotation and hashing. The experiment was conducted by utilizing 50 queries, and the hybrid approach involved both semantic annotations and with hashing produced better performance while compared to other approaches.

Cao *et al.*, defined and solved the problem of multi-keyword ranked search over the encrypted cloud data and established a variety of privacy requirements [25]. Among the multi-keyword semantics, the efficient similarity measure of the "coordinate matching" was chosen. As many matches as possible, to capture the similarity between the search query and data documents, and further use "inner product similarity" to formalize such a principle for similarity measurement quantitatively. To meet the challenge of supporting multi-keyword semantic without privacy cracks, the basic idea of Multi Keyword Ranked Search over Encrypted Data (MRSE) using the secure inner product computation was proposed. Two improved MRSE schemes were given to achieve various stringent privacy requirements.

Rodriguez *et al.*, proposed a semantic platform for cloud service annotation and retrieval from their descriptions [26]. The system automatically annotated different cloud services from their natural language description, which was available in several document formats such as XML, HTML, or PDF. The research method was additionally implemented by considering a multi ontology environment to be able to cope up with several domains.

Jain *et al.*, suggested a novel approach for extracting information from the web [27]. A massive number of documents were presented on the web, retrieving the relevant information from them was a tiresome task. It generated the concept of information retrieval and the semantic web. To extract information from web documents, they used semantic mark-up documents.

Chauhan *et al.*, instigated a domain-specific semantic information retrieval system by using suitable tools and the proposed technique of ontology based automatic query expansion [28]. It utilized the query concept as well as the synonyms of the concepts to perform query expansion. The new terms were added only for consistency of similarity within a threshold. Only the most important document acquired the top rank [29].

Singh *et al.*, emphasized the concept of the semantic web and several approaches used for retrieving information from the web [30]. Information Retrieval over the collection of these documents offered new challenges and opportunities. In this study, a framework for integrating the search that supported the Inference engine is presented by the authors.

The summary of the new ontology based information retrieval systems reported in the literature is given in Table 2.1.

Table 2.1: Summary of the recent ontology based information retrieval methods

Name of Authors	Topics / Concept/ Domain	Dataset used	Pros	Cons	Performance Accuracy (Precision)	Year
Yang Yang, Linjun yang, Gangshan Wu, Shipeng Li [31]	Query Context Bag of-object Retrieval	Object vocabulary containing Query relative objects from frequent patches	Image relevance prediction Model using query context	Did not attempt to reduce Semantic gap	88.4	2014
Luke S, Spector L, Rager D [32]	Google and Knowledge Graph	Helena Cook's Web page	Probabilistic modeling framework	linking and representing facts Feel like a compulsion to go to the links suggested		2013
Drymonas, E., Zervanou, K. and Petrakis,	OntoGain	Medical	Unsupervised ontology acquisition from the unstructured	Term ambiguity is not addressed	89.7%	

E.G [33].		Computer Science	text which relies on multi-word term extraction		86.67%	2010
Researches From Free University Of Berlin and Leipzig University [34]	DBpedia	9.5 billion RDF triples extracted from Wikipedia (English)	Extract information from Wikipedia and organizes	Can’t trust All info in Wikipedia		2009
Hepp M [35]	GoodRelations	Products and services offered on the web	Consumers and enterprises can search for suitable suppliers using products and services ontologies			2008
Vanessa Lopez Victoria Uren [36]	AquaLog	Department web site in the Knowledgebase	Question-answering system with no	Can’t capture the whole semantics	69.11	2007

2.3. Review on Ontology development Tools

Ontology development demands the use of various software tools [37]. A range of open-source and commercial tools are available which assist in the development of various ontologies called Ontology Editors. These tools can be applied to several stages of the ontology life cycle including the creation, implementation, and maintenance of ontologies. Ontologies are independent of the applications that use them and lead to more accessible software and knowledge maintenance and contribute to the semantic interoperability between applications. Today a variety of developing environments exist for building ontologies like Protégé 3.4, IsaViz, Apollo, and SWOOP [38] [39] [40] [41]

To construct ontology, one should have an ontology specification language. Among several ontology languages, the Web Ontology Language (OWL) is extensively accepted as a standard for indicating and sharing knowledge in the Semantic Web context [42].

Four commonly used ontology authoring tools such as Apollo, Protégé 3.4, IsaViz, and SWOOP are considered here for the comparative study, taking into account the advantages of these tools. Tools that provide support for the different phases of the ontology engineering process are referred to as ontology building tools. These tools are used for building a new ontology either from scratch or by reusing existing ontologies, which usually supports editing, browsing, documentation, export, and import from different formats, views, libraries and they may have attached inference engines, etc. [43]

Protégé is an ontology and knowledge base editor created by Stanford University. Protégé is a tool that assists the construction of domain ontologies, customized data entry forms to enter data. Protégé allows the definition of classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and the properties of these relationships. Protégé is free and can be downloaded from http://protégé.stanford.edu [38].

IsaViz is a visual environment for browsing and authoring RDF models as graphs. W3C Consortium offers this tool. Emmanuel Pietriga developed IsaViz [39].. The first version was developed in collaboration with Xerox Research Centre Europe, which also contributed with XVTM, the ancestor of ZVTM (Zoomable Visual Transformation Machine) upon which IsaViz is built.

Apollo is a user-friendly knowledge modeling application [40]. The modeling is based around the fundamental primitives, such as classes, instances, functions, relations, etc. The internal model is built as a frame system according to the internal model of the OKBC protocol. Apollo's class system is modeled according to the OKBC. The knowledge base consists of ontology that is organized hierarchically. Ontology can inherit other ontologies and then use classes of inherited ontology like its own. Every ontology inherits at least one ontology − a default ontology, which contains all primitive classes: Boolean, integer, float, string, list, etc. Class contains slots of two types: non-template and template slots.

SWOOP is a Web-based OWL ontology editor and browser [41]. Validation and presentation syntax is contained in which provides multiple ontology environments with excellent reasoning support. Ontologies can be compared, edited, and merged in this editor. Different ontologies can be compared against their Description Logic-based definitions, associated properties, and instances. SWOOP's interface has hyperlinked capabilities so that navigation can be simple and easy. It does not follow a methodology for ontology construction. Users can reuse external ontological data [37].

Bansal.R., *et al.*, compared various tools for ontology development based on specific features such as modeling features/limitations, base language, web support and use, import/export format, graph view, consistency checks, multi-user support, etc. [44].

The result for comparison of tools are shown in Table 2.2 to Table 2.5 which are categorized based on

1. Tool architecture – to examine the extensibility and ontology storage shown in Table 2.2.

2. Tool's interoperability – to check the Import Format, Export Format, and Merging features shown in Table 2.3.

3. Tool's inference services – to examine the Inference Engine, Exception Handling, and Consistency Checking shown in Table 2.4.

4. Tools' usability – to discuss the Collaboration with other tools, Ontology Library, and Visualization shown in Table 2.5.

Table 2.2: Architecture of tools

Feature	Apollo	IsaViz	Protégé	SWOOP
Extensibility	No	No	Via plug-ins	No
Ontology Storage	Files	Files	Files & DBMS	Files

Table 2.3: Interoperability of tools

Feature	Apollo	IsaViz	Protégé	SWOOP
Import Format	OCML	XSLT,RDF (S), OIL,DAML+ OIL , OWL	XML, RDF (S), XML Schema and OWL	RDF (S), OIL, DAML
Export Format	OCML	XSLT,RDF(S) , OIL,DAML+ OIL , OWL	XML, RDF (S), XML Schema, Java, html	RDF (S), OIL, DAML
Merging	No	No	with ANCHORPROMPT plug-in	No

Table 2.4: Inference services of tools

Feature	Apollo	IsaViz	Protégé	SWOOP
Inference Engine	No	Yes	With PAL	No
Exception Handling	No	No	No	Yes
Consistency Checking	Yes	with type inheritance and detection of cycles in hierarchies	with plugins like FACT and PAL	Only checks writing mistakes

Table 2.5: Usability of tools

Feature	Apollo	IsaViz	Protégé	SWOOP
Collaboration with other tools	No	No	No	No
Ontology Library	Yes	No	Yes	No
Visualization	No	with plug-ins like GraphViz	No	No

By considering the above listed features, Protégé seems to be a much better tool for developing ontology among the various tools available.

2.3.1. Review on Ontology Learning Techniques

The representation of ontology is commonly defined as an explicit specification of a conceptualization of the real world domain such as education, agriculture, health services, e-government etc. These are explicitly represented with existing objects, entities, and associations between them. The ontology learning techniques are broadly classified into three, namely, linguistic, statistical, and logical approaches.

As ontology is defined as an explicit conceptualization of terms and their relationship to a domain [45], ontologies can be created by extracting relevant instances of information from text using a process called ontology population. However, manually designing such large ontologies is a difficult task, and it is impossible to build ontologies for all available domains [46]. Therefore, studies are also active in the area of automatic ontology learning.

Textual information retrieval is based on keywords or expressions extracted from documents and employed as the building blocks of both document and query representations. However, keywords may have different levels of granularity. and the web was disconnected, inconsistent, and dumb. Extracting useful information from such a type of web was an erroneous process. To tackle this problem, the concept of the semantic web was introduced by Maedche and Staab in 2001 [47].

For comparing different identifiers to identify the same concept, for instance, surname and last name, an application needs to have a method of discovering such common meanings for whatever databases it queries. This method of discovery is made available through ontology [15].

The work of Maddi *et al.*, presents a way to extract ontologies for text documents using singular value decomposition (SVD) to obtain the concepts from terms. It represents the results using bipartite graphs [48]. A domain specific ontology is constructed by reading the collection of related text documents, and extracting ontological information statistically. This system can determine word frequencies, which are formed into a frequency matrix. Singular Value Decomposition is performed on the frequency matrix, and the resulting matrices are used to determine statistical relationships between documents and terms. These relationships are then used to build an ontological graph. The ontology graph can be viewed and manipulated with graphical user interface (GUI), or loaded into user code. Thus this system provides a procedure of generating richer domain specific knowledge, with minimal user interaction.

Considering the solutions that generate ontologies for applications and specific document types, there is the work of Sanchez and Moreno *et al.,* that presents a methodology for automatic construction of domain ontologies in which concepts are obtained as keywords from Web pages [49].

Fortuna *et al.*, present a process for obtaining concepts semi-automatically, because the solution suggests only the terms sets. From this suggestion, the user chooses the concepts and makes connections between them [50] .

The creation of ontologies for lecture notes in distance education systems is presented by N. Gantayatin in his work [51]. He uses natural language processing to extract keywords, algorithms based on frequency to select the concepts from the keywords, and association rules algorithms to define the semantic relations.

Gillam and Ahmad propose the acquisition of concepts using statistical methods for comparison between a vocabulary created by domain experts and the general vocabulary words from the text [52]. For the hierarchy creation, it uses solutions from literature, such as smoothing, extraction and placement techniques. The most popularly used data sets for ontology learning in various research studies are listed in Table 2.6.

Table 2.6: The most popularly used data sets for ontology learning

Sl. No	Corpus	No. of documents	Domain
1	Mecklenburg Vorpommern [53]	1047	Tourism
2	Lonely Planet [53]	1801	Traveling
3	British National Corpus [54]	4124	News
4	Reuters-21578 [55][56]	21578	News
5	Genia Corpus [57]	2000	Biological
6	Planet Stories [58]	307	Stories

Assessing the quality of ontology acquisition is a crucial aspect of smart web technology as it allows the researchers and practitioners to assess the correctness at the lexical level, coverage at the concept level, wellness at taxonomic level, and adequacy at the non-taxonomic level of yielded ontologies. Evaluation of ontology acquisition makes it possible to renew and remodel the entire ontology learning process in case of unexpected resultant ontologies, which do not support the specific requirements of a user. An overview of the ontology evaluation approaches is given in Table 2.7.

Table 2.7: Overview of ontology evaluation approaches

Evaluation Approach	**Applicatio n Based**	**Data level**	**Assessment by human**
Lexical, vocabulary, concept, and data	X	X	X
Hierarchy and taxonomy	X	X	X
Other semantic relations	X	X	X
Context and application	X		X
Syntactic			X
Structure, architecture, and design			X

Several ontologies are being developed so far, but they have only considered certain functionalities of the University; *viz*, some of them consider only examination system of the University [59], a few of them considered only courses offered [60], some of them taken into account scientific research management [61], some have proposed part of a data warehouse of the University [62], some has developed an ontology to guide internship process [63], some of them included few of subdomains [64][65]. But the present ontology brought together all the necessary concepts, their object and data properties to help the University and their expert and inexpert staff in all their related concerns.

2.4. Review on Text Preprocessing Techniques

Manipulation of texts for information retrieval and automatic indexing and abstracting for producing text in a desired format has been an essential area of research in NLP. There are mainly two classifications in the area of natural language text processing that allows the structuring of large bodies of textual information intending to retrieve particular information or to derive knowledge structures that may be used for a specific purpose. Automatic text processing systems generally take some form of text input and translate it into an output of some different forms. The natural language text processing systems convert ambiguous natural language queries and texts into unambiguous internal representations in which matching and retrieval can take place.

Liddy, a natural language text processing system, begins with morphological analyses. Stemming of terms, in both the queries and documents, is done to get the morphological features of the words involved. The lexical and syntactic processing includes the utilization of lexicons for determining the characteristics of the words, recognition of their parts-of-speech, determining the words and phrases, and for parsing of the sentences [66].

Past research works concentrating on natural language text processing systems have been reviewed by Haas [67], Mani and Maybury [68], Smeaton [69], and Warner [70]. Some NLP systems have been built to process texts using particular small sub-languages to reduce the size of the operations and the complexity level. These domain-specific analyses are primarily named as Sublanguage analyses by Grishman and Kittredge [71]. Some of these studies are limited to a particular subject area as medical science; others deal with a specific type of document such as patent texts.

In their work, Johannes Leveling *et.al.,* tried to find the effects of stop-word removal in different phases of a system for SMS-based Frequently Asked Query retrieval [72]. The experiments are conducted on the FIRE 2011 monolingual English data [73]. The steps of FAQ systems are normalization and correction of SMS, retrieval of FAQs comprising answers using the BM25 retrieval model, and recognition of out-of-domain queries.

In this work, Charniak *et al.*, achieved 90% accuracy in assigning the part-of-speech tag to a word by applying simple statistical measures [74].

Fuchun Peng, Nawaaz Ahmed, Xin Li, and Yumao Lu propose a context-sensitive stemming method [75]. They perform context-sensitive analysis on the query then predict which of its morphological variants is useful to expand a user query before submitting the query to the search engine. They also perform a context-sensitive document matching for those expanded terms.

2.5. Review on Vector-based Similarity Indexing

One of the most challenging problems in information retrieval is to retrieve relevant documents based on a query given by the user. Studies have shown, however, that users appreciate receiving more information than only the exact match to a query. Depending on the word(s) given in the user's query, and with an option to choose more relevant terms which narrow the request, retrieval will be more efficient.

The similarity of the document with the query is to be determined, and they are to be ranked in the similarity index to find the best matched document. Vector Space Model (VSM) is considered as the most popular and efficient model owing to its simplicity and fast processing. It can handle weighted terms and produces a ranked list as output. Vector Space Model, sometimes referred to as the Term Vector Model, is an algebraic model of representing text documents as vectors such as index terms. It has found its applications in information retrieval, information filtering, indexing, and ranking. In the information retrieval process, both documents and queries are represented as vectors.

Each dimension of the vector corresponds to a separate term. If a term occurs in the document, its value in the vector is non zero. Different ways of computing these values called weights are being developed. One of the best schemes is known as tf-idf weighting.

The basic idea of the VSM is to represent text as a Bag of Words (BoW). To have a compact representation, the word ordering in a document is ignored, and the document is represented by a vector of word frequencies. The definition of the *term* varies with the application. Usually, single words, keywords, or longer phrases are denoted by terms. If words are chosen to be the terms, then dimensionality of the vector is the number of words in the vocabulary [76].

Text having different lengths, namely a word, sentence, paragraph, or document, is transformed into a numeric vector called Vector Space model to be used in applications like similarity detection or machine learning algorithms. The most basic text vectorization method is TFIDF, which defines a space where a separate and orthogonal dimension represents each term in the vocabulary. Despite its acceptance and simplicity, basic TFIDF may suffer from some problems [77]. They can be listed out as

- Ignorance of n-gram phrases
- Complications with incremental updates upon addition of new documents
- A large number of dimensions.

To deal with the two former issues, variants of TFIDF have been proposed:

(i) To incorporate n-grams as new terms, and/or

(ii) To adjust for the timing of the use of vocabulary across the timeline of the corpus.

To deal with the latter issue, text embedding methods are used, which attempt to address the high-number of dimensions by transforming each text into a low-dimensional vector [78]. Text embedding techniques can be classified into count-based models and prediction-based models. Count-based models, also referred to as topic models, create a document term matrix where the weight of each cell is based on the number of times a term appears in the focal document. Prediction-based models, otherwise known as neural models, can predict the occurrence of a term/document based on surrounding terms to learn a vectorization for each term/document.

2.5.1 TFIDF Models

Term Frequency–Inverse Document Frequency (TFIDF) is one of the most common vectorization techniques for textual data with many possible variations [79]. Yang and Huang, proposed an improved TF-IDF algorithm based on the traditional TFIDF for calculating weight according to the location and length of the keyword and proved it produces better precision and recall value when compared to the traditional TFIDF [80]. In his work, J Ramos provided evidence that TF-IDF efficiently categorizes relevant words that can enhance query retrieval [81]. J Sang *et.al.,* used an improved TFIDF model for feature selection and weighting methods for classification in the Internet of things [82]. The experiment results show considerable improvement in the precision and recall of classification and a decrease in the amount of information to be transferred on the network by text classification which helps in the efficient information access. Various interpretations of TF-IDF—based on binary independence retrieval, Poisson, information theory, and language modelling are reviewed by Roelleke and Wang [83]. This study uncovers components such as divergence from randomness and pivoted document length to be inherent parts of document-query independence (DQI) measure, and an integral of the DQI over the term occurrence probability leads to TF-IDF. TFIDF considers two documents as similar if they share rare but informative words.

2.5.2 Word2Vec, Doc2Vec and Bag of Words Models

Unlike models that simply count terms, neural models capture information from the context of other words that surround a given word, hence taking order into account. The most well-known model for predicting word context is W2V (Word to Vector), where the authors propose an algorithm based on a shallow neural network of three layers to learn word vectors [84]. Prior research has shown the W2V model to perform well with analogy and similarity relationships. Given a context window size, the W2V algorithm comes in two forms. In the first form, known as CBOW (Continuous Bag of Words), the model predicts the probability of a target word given a context word. In the second form, known as Skip-Gram, the model predicts the probability of a context word given a target word.

Alternatively, an extension of W2V, known as D2V (Document to Vector) or paragraph vectors, has been proposed to obtain document vectors. This directly by considering a document as a particular context token that can be added to the training data, such that the model can learn token vectors and consider them as document vectors [84]. Building on W2V, the

D2V algorithm also comes in two flavours: Distributed Memory (DM) and Distributed Bag of Words (DBOW). D2V can be implemented incrementally and showed a better performance compared to the previous approaches in some similarity detection benchmarks. However, on the negative side, it has numerous hyper-parameters that should be tuned to harvest its power to the full extent.

2.5.3 Topic Models

Topic models transform a text into a fixed size vector, equal to a given number of latent topics. The vector represents the probability distribution that the focal text relates to each of the different topics. In practice, each topic is a weighted average of a subset of terms. Similar to TFIDF, topic models treat the text as a bag of words where the order of words is ignored. On the down side, interpretation of each topic can be subjective, and determining the right number of topics requires tuning of the model. The use of topic modelling does not provide a full meaning of the text but provides a good overview of the themes, which could not have been obtained otherwise [85]. DiMaggio et al., found a key distinction in the use of topic modelling is that its use is more of utility than accuracy, where the model should simplify the data in an interpretable and valid way to be used for further analysis [86]. They noted that a subject-matter expert is required to interpret the outcome and that the analysis is formed by the data. Table 2.8 gives an overview of the applications of Topic modelling as described by various researchers.

Table 2.8: Applications of Topic Modelling

Reference	Data	Method	Proposed use	Size
Grimmer [86]	Press releases	Own developed method	Development of a Bayesian topic model	24,000
Quinn et al. [87]	Speeches (Text)	Own developed method	Development of a statistical learning model	118,000
Baum [88]	Speeches (Video)	LDA	Identify topics of German politicians	2581
DiMaggio et al. [89]	Newspapers	LDA	Identify central concepts in news coverage	8000
Jockers and Mimno [90]	Books	LDA	Identify literature themes	3346
Ghosh and Guha [91]	Tweets	LDA	Identify tweets related to obesity	2,581,283
Evans [92]	Newspapers	LDA	Identify subjects of discussion	14,952
Guo et al. [93]	Tweets	Dictionary-based analysis and LDA	Compare dictionary based analysis vs. LDA	77,000,000
Elgesem et al. [94]	Blogs	LDA	Identify topics in blogs regarding the arrest of Edward Snowden	15,000
Maier et al. [95]	Web pages	LDA	Investigate the validity and reliability of LDA	344,456

Topic modelling is used extensively by researchers to simplify data interpretation for further processing. Most of them used Latent Dirichlet Allocation (LDA) method for topic modelling which depicts document-topic density, number of topics and the number of terms in a single topic.

2.5.4 Existing Comparative Studies

Several recent studies compare different vector space models concerning their similarity detection power for texts. Very few of them, however, target similarity detection for longer texts (e.g., documents).

Baldwin et l., compared D2V variants against an averaging W2V, as well as a probabilistic n-gram model for two similarity tasks [96]. In the first task, the goal is to detect the similarity of forum questions; the second task aims to detect similarity between pairs of sentences. The authors find that D2V is superior for most cases and that training the model on large corpora can improve their results.

In their work, N. Marwa *et al.*, attempt to detect similarity between sentences and compare several neural models against a baseline that simply averages together word vectors[97]. They found that more complex neural models work best for in-domain scenarios (where both the training data set and the testing data set are from the same domain). At the same time, a baseline of averaging word vectors is hard to beat for out-of-domain cases.

S. Arora *et al.*, proposed a method for sentence embedding through the weighted averaging of word vectors as transformed by a dimensionality reduction technique [98]. They show that their text vectors outperform well-known methods to detect sentence to sentence similarity.

M. Pagliardin *et al.*, use an unsupervised method to vectorise sentences and show that their method outperforms other state-of-the-art techniques to detect the similarity of short sequences of words [99]. In each of these studies, however, the objective was to determine the performance of similarity detection algorithms on relatively short sections of text. There has been much less research on the performance of similarity comparisons for longer text.

In another study, Andrew M. Dai *et al*. compared D2V to a weighted W2V, LDA, and TFIDF to detect the similarity of documents in Wikipedia and arXiv corpus [100]. They found that D2V can outperform other models on Wikipedia, but that D2V could barely beat a simple TFIDF baseline on arXiv.

Alvarez *et al*. compared several algorithms to detect similarity between biomedical papers of PubMed and found that advanced embedding techniques again can hardly beat simpler baselines such as TFIDF [101]. This work adds to the stream of research comparing text vectorization methods for longer text. In particular, we focus on a real-world problem with an objective standard for determining similarity. In contrast, prior research had to rely on broad categorizations from repositories such as Wikipedia and arXiv. This work provides a comparative study of semantic similarity methods.

2.5.5. Hybridization of Similarity Measures

Generally, there are two approaches for searching relevant documents in the literature. The first one is the keyword-based approach. The second approach uses the whole document as a vector to compute a similarity measure, and the query is also presented as a vector using VSM. Similarity measures such as Cosine, Jaccard, Euclidean, and Okapi detailed below are extensively used in VSM to enhance the performance of IR systems.

Cosine Coefficient: In this method the cosine of the angle between the query and document vector is computed, as shown in equation (2.1). The numerator represents the dot product of the vectors q and d, while the denominator is the product of their Euclidean lengths:

$$Cos(Q, D_i) = \frac{\sum_{j=1}^{t} W_{qj}\, d_{ij}}{\sum_{j=1}^{t}(d_{ij})^2 \sum_{j=1}^{t}(W_{qj})^2} \qquad (2.1)$$

Jaccard Coefficient: The Jaccard coefficient is defined as the size of the intersection divided by the size of the union of the document and query vectors, as shown in equation (2):

$$Jaccard(Q, D_i) = \frac{\sum_{j=1}^{t} W_{qj} d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 + \sum_{j=1}^{t} (W_{qj})^2 - \sum_{j=1}^{t} W_{qj} d_{ij}} \qquad (2.2)$$

Okapi-BM25: This similarity measure considers not only the occurrence of the query terms but also the average length of the whole collection and the length of the document under evaluation. The mathematical formula for Okapi-BM25 is shown in equation (2.3):

$$Okpi - BM25(Q, D_i) = \sum_{T \in Q} W \frac{(k_1 + 1) tf}{K + tf} \times \frac{(k_3 + 1) qtf}{k_3 + qtf} \qquad (2.3)$$

Where Q is a query that contains the words T,

D is the document set, K is $(k_1 (1 - b) + b.dl / avdl)$ k_1, b and k_3 are constant parameters. dl and $avdl$ are the document length and average document length, respectively. tf is the term frequency of the term within a document and qtf is the term frequency in the query. W is $\log(N - n + 0.5) / (n + 0.5)$ where N is the number of documents, n is the number of documents containing the term.

Pathak *et al.* also tried to design a composite similarity measure by finding a linear combination of these similarity measures to overcome its limitations [102]. In their approach, the weights for different similarity measures are determined using GA, but it suffers two drawbacks. First, finding the best possible combination of weights through GA is a time-consuming process. Second, it is not able to capture vagueness and uncertainty as the documents and queries are written in natural language.

Pathak *et al.*, treated an overall matching function as a weighted sum of the scores returned by different matching functions. The weights are determined by using GA [102]. The fitness function used in this approach is given by equation (2.4)

$$E = 1 - 1 \Big/ \left[\frac{\alpha}{P} + \frac{1-\alpha}{R} \right] \qquad (2.4)$$

where α is a parameter used to express the degree of user preference for precision *(P)* or recall *(R)* components. A higher value of α characterizes a user with less preference for recall, while a lower value of α characterizes one with less preference for precision. For each similarity measure, a randomly chosen weight (in the range 0.0–1.0) is assigned. The weights are encoded using the actual real numbers between 0.0 and 1.0 (inclusive). The experiments are conducted for a population size of 50, run for 75 generations; α is equal to 1 and crossover, and mutation rates equal to 0.6 and 0.1, respectively.

Gupta *et al*., proposed a fuzzy logic-based approach for hybrid similarity measure [103]. They presented two different fuzzy hybrid similarity measures in their work. The first one is a combination of cosine, Euclidean, and Jaccard, that is, FBHSM1, and the second one is a combination of cosine, Jaccard, and Okapi-BM25, that is, FBHSM2. The Corresponding equations are given in (2.5) and (2.6).

$$FBHSM1(D,Q) = W_1 \times Co\sin e + W_2 \times Jaccard + W_3 \times Euclidean \quad (2.5)$$

$$FBHSM2(D,q) = W_1 \times Co\sin e + W_2 \times Jaccard + W_3 \times Okapi - BM25 \quad (2.6)$$

The following section provides a review on short query processing used for semantic web.

2.5. Review on Short Query Processing for Semantic web

The Information Retrieval community has investigated many different techniques to retrieve documents from extensive collections of documents for query processing.

Zhu *et al*. designed and realized a framework of constructing a natural language interface to a graph-based bibliographic information retrieval system [104]. This framework permitted users to query bibliographic information by expressing and responding to the queries specified in natural language. While interpreting natural language queries, one of the important things was to recognize the bibliographic named entities in natural language queries. They tested the framework using a large empirical dataset, and the experimental results indicated that the method correctly interpreted 39 out of 40 natural language queries with different levels of complexities.

The work discussed by Mahboob Alam Khalid and Susan Verberne quantitatively compares the impact of sliding windows and disjoint windows on the document retrieval for query processing [105]. For the document retrieval to why-questions from Wikipedia, the best retrieval model is Term Frequency Inverse Document Frequency (TF/IDF), and sliding windows give significantly better results than disjoint windows.

In the NTUBROWS system, the first phase is query processing [106]. During this phase produces a set of keywords. They are passed to the retrieval model, which outputs a list of relevant documents. The ranking module adjusts the ranking of these relevant documents by considering various features such as frequency, term's distribution, position in a paragraph, etc.

Kothari *et al*. presented a FAQ-based query processing system over an SMS interface [107]. They handled the noise in an SMS query by framing the query similarity over queries as a combinatorial search problem, where the search space consists of a grouping of all possible dictionary variations of tokens in the noisy query.

Mukul Rawat *et al*. have presented an “Improved version of SMS Based FAQ Retrieval System" [108]. They have mainly added three features to the system. They are Proximity score, Length score, and an answer matching system. The experiments show that the accuracy of the system gives the accuracy of the current state- of- the-art systems reported in the literature.

Mihalcea and Moldovan described the reduced application of statistical methods in word sense disambiguation, basically due to the lack of widely available semantically tagged corpora [109]. They report research that enables the automatic acquisition of sense tagged corpus and based on the information provided in WordNet and the information gathered from the Internet using existing search engines.

Johannes Leveling presents results for DCU's second participation in the SMS-based FAQ Retrieval task [110]. In this work, the SMS queries are converted into a normalized form and submitted to a retrieval engine to obtain a list. Then classify the training data then determine which queries are out of the domain and which are not. Then translate the full Hindi FAQ documents into English and retrieve results from the translated Hindi FAQ documents.

Myoung-Cheol Kim and Key-Sun Choi present a comparison of similarity measures, including Jaccard, Dice, and Cosine similarity measures for the proper selection of additional search terms in query expansion [111]. Besides, they also compare the performance of two more similarity measures, Average Conditional Probability (ACP) and Normalized Mutual Information (NMI). ACP is the mean value of two conditional probabilities between a query term and an additional search term. NMI is a normalized value of the two terms' mutual information.

Deirdre Hogan submitted a system using monolingual Information retrieval in FIRE (Forum of Information Retrieval Evaluation) 2011, where they submitted their second system using both Monolingual and cross lingual FAQ retrieval in FIRE 2012 [112].

2.6. Review on Fuzzy Ontology

Lee *et al.*, present a solution for creating ontologies from text documents in Chinese language using fuzzy logic, similarity, and clustering to obtain the taxonomy of the ontology [113].

Fuzzy ontologies have been used in many domains. For example, Rodríguez *et al.* proposed a Fuzzy Ontology (FO) to model human behavior [114]. Lee and Wang proposed a five-layer fuzzy ontology and utilized it in a fuzzy expert system for diabetes management [115]. Ali *et al.*, proposed this for opinion mining. Moreover, it has been managed in some AI systems as rule-based systems [116]. The usage of FO with rule-based systems is less applicable than other systems because these systems require collecting explicit models of domains. FO extends the capabilities of the crisp one and improves the accuracy and applicability of Case-Based Reasoning (CBR) in the medical domain.

Park *et al.*, utilized crisp ontology in a fuzzy CBR system for the prevention of ship collision [117]. Alexopoulos *et al.*, tried to build an FO for CBR using fuzzy algebra [118]. Their proposed ontology has been designed for electronic libraries. However, utilizing relational databases and conceptual data models for building fuzzy ontologies is the most suitable form to build CB fuzzy ontology.

Sheker *et al.,* presents a fuzzy case base reason for the diabetes mellitus domain to enhance the semantic and storage of CBR knowledge-base [119]. The combination of ontology and fuzzy logic reasoning is critical in the medical domain. Case-base representation based on fuzzy ontology is expected.

Fuzzy ontology building can be based on several aspects. The different approaches and the critical works are given in table 2.9.

Table 2.9: Fuzzy Ontology building approaches

Aspect	Author
Textual databases	Lau *et al.*, [120], Ghorbel *et al.*, [121]
Crisp ontologies and the use of fuzzy membership functions	Kaur and Kaur, [122], Baazaoui-Zghal and Ben Ghezala [123], Chien *et al.*, [124]
XML model by transforming the fuzzy XML document and DTD in fuzzy ontology	Zhang *et al.*, [125]
WordNet to compute the distance between concepts or to extract concepts or hierarchy of concepts	Truong *et al.*, [126]

From the review of previous works, it is concluded that fuzzy ontologies can be used to represent domain information consisting of fuzziness or vagueness in ontology terms by fuzzifying them with membership functions.

2.7. Conclusion

The studies on the development of various information retrieval techniques are reviewed in this chapter. Ontology based semantic information retrieval, ontology developing tools and learning techniques are reviewed and studied in detail. The vector space models for representing queries and documents with different approaches are also reviewed. Similarity measures used to find the similarity between documents and queries, are also considered and reviewed. A quick review on fuzzy ontology is also done in this chapter. It is observed that not much research work has been carried out in semantic information retrieval models incorporating vector representation with ontology.

Chapter 3

Dataset Acquisition and Preprocessing

3.1. Introduction

The word semantic is used to indicate meaning in a language. Extraction of appropriate information based on the query, just like a human does, is one of the most challenging tasks in machine learning. In semantic-based searches, the key features considered are frequency of words, syntactic structure of the natural language, and other linguistic elements [127]. The domain of interest for this study is any state University in India, and hence the documents and queries containing the University related information are collected. Since this specific domain consists of a limited amount of University related data sets, its own data set is prepared for experimental purposes.

The choice of a precise sample dataset plays a crucial role in assessing the performance of the study. A uniform set of data with a reasonable amount of representative samples helps to reduce the initial preprocessing work and increases the reliability of the algorithms proposed. The prominent research groups typically share their database with the public and allow free access from their web portals. The web portal of the

Universities and various academic and government web sites are other sources of data required for this study.

The first part of this chapter explains the creation of the document data set that is used for the conduct of experiments. The preparation and preprocessing techniques used in this data set are defined in sections 3.3 and 3.4. The creation and preprocessing of the query data set are described in sections 3.5 and 3.6, respectively. WordNet is used in this study for finding synonyms, and hence an overview of WordNet is given in section 3.7. Finally, section 3.8 concludes the chapter.

3.2. Data Set Creation

The domain considered for the current study is University and, in particular, the information related to Administration, Academics, Examination, Finance, and Planning & Development branches of Universities. The experimental arrangement of proposed work is operated on the enquiries on different Departments under the University and the general enquiries related to the University system. It includes a Query set and corresponding Document set.

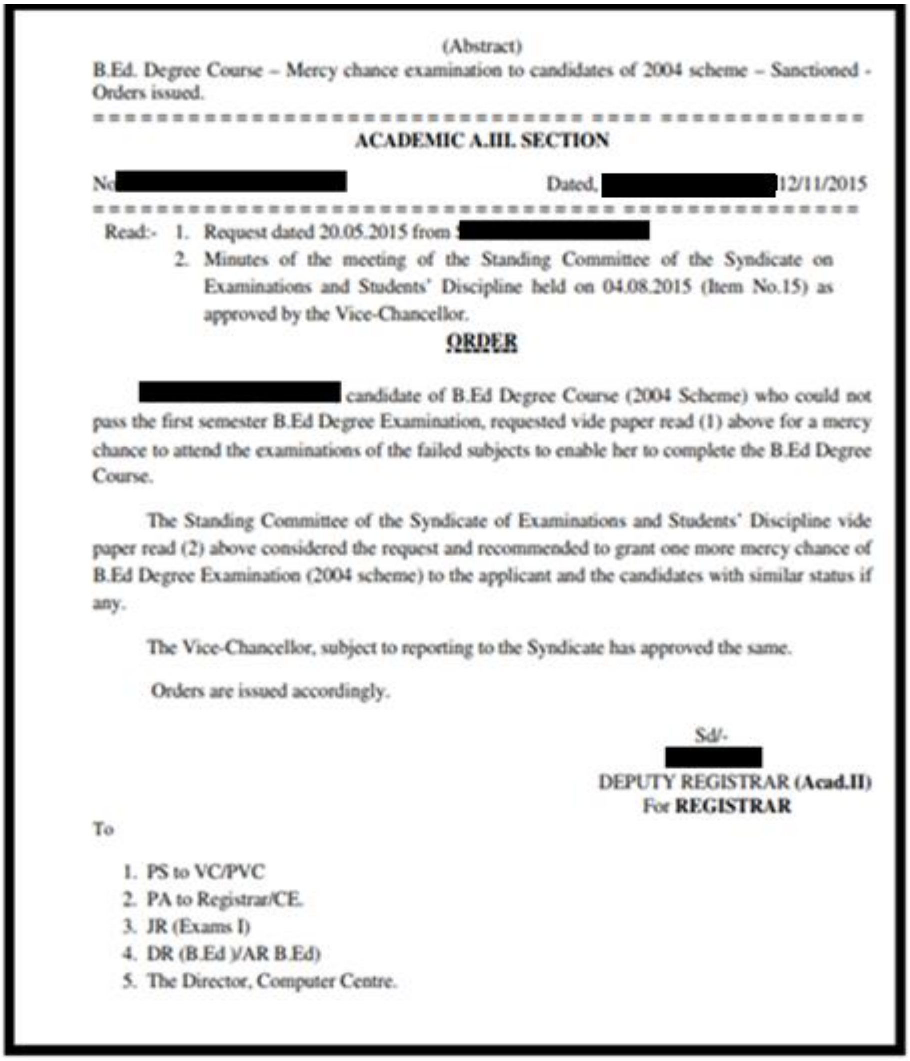
The dataset for this study consists of documents, which are the orders released by the different Universities or other officials related to higher education all over India, for the wellbeing of students and staff on academic affairs. These documents are available at the University website, Government websites, which are orders, and notices generally in Portable Document Format (pdf). The dataset also consists of natural language queries from various stakeholders related with academic affairs. Both Query and Document set is subdivided into five separate categories, related to Academics, Administration, Examination, Finance, and Planning and Development branches of Universities.

3.3. Document Data Set Preparation

For the data extraction process, a web crawler was built which scrapes the content from any university or educational websites. The documents collected are PDF files of the University and / or Government orders and notices issued by the authorities concerned.

These documents' content varies both in structure and size according to the subject or topic handled. Examples of the documents are given in Figure 3.1 and 3.2 (a, b).

To excerpt the relevant information from these documents, the different sections separated by specific characters and certain keywords are considered. Each of these documents will have a section called an abstract, which indicates the topic of the document, then a section name, order number, and date. The actual content of the document will be followed. Finally there is a signature part with designation of the issuing authority and the list of officers to which the copy of the order to be forwarded is marked.



(Abstract)

B.Ed. Degree Course – Mercy chance examination to candidates of 2004 scheme – Sanctioned - Orders issued.

ACADEMIC A.III. SECTION

No[redacted] Dated,[redacted]12/11/2015

Read:- 1. Request dated 20.05.2015 from [redacted]

2. Minutes of the meeting of the Standing Committee of the Syndicate on Examinations and Students' Discipline held on 04.08.2015 (Item No.15) as approved by the Vice-Chancellor.

ORDER

[redacted] candidate of B.Ed Degree Course (2004 Scheme) who could not pass the first semester B.Ed Degree Examination, requested vide paper read (1) above for a mercy chance to attend the examinations of the failed subjects to enable her to complete the B.Ed Degree Course.

The Standing Committee of the Syndicate of Examinations and Students' Discipline vide paper read (2) above considered the request and recommended to grant one more mercy chance of B.Ed Degree Examination (2004 scheme) to the applicant and the candidates with similar status if any.

The Vice-Chancellor, subject to reporting to the Syndicate has approved the same.

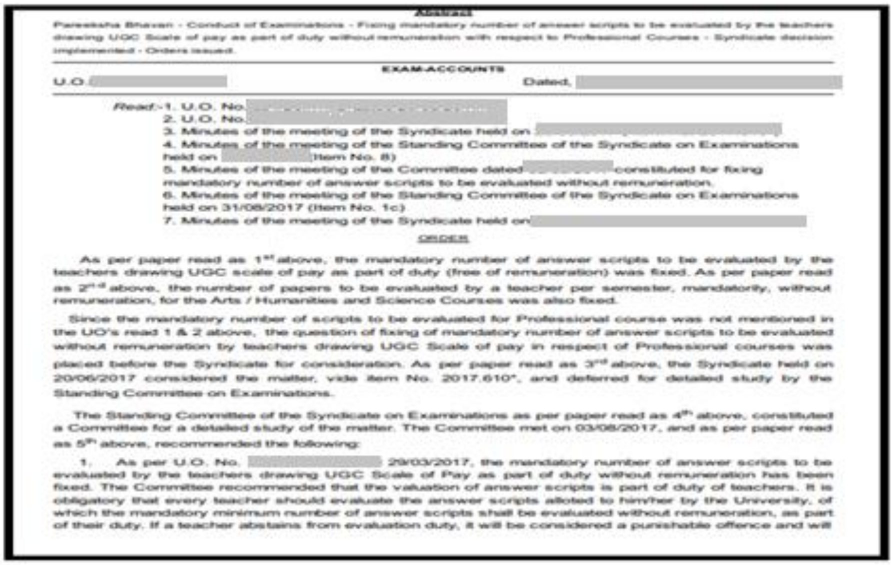
Orders are issued accordingly.

Sd/-
[redacted]
DEPUTY REGISTRAR (**Acad.II**)
For **REGISTRAR**

To

1. PS to VC/PVC
2. PA to Registrar/CE.
3. JR (Exams I)
4. DR (B.Ed)/AR B.Ed)
5. The Director, Computer Centre.

Figure 3.1 Document related to B.Ed Degree course – mercy chance



Abstract

Pareeksha Bhavan - Conduct of Examinations - Fixing mandatory number of answer scripts to be evaluated by the teachers drawing UGC Scale of pay as part of duty without remuneration with respect to Professional Courses - Syndicate decision implemented - Orders issued.

EXAM-ACCOUNTS

U.O. [redacted] Dated, [redacted]

Read:-1. U.O. No. [redacted]
2. U.O. No.
3. Minutes of the meeting of the Syndicate held on [redacted]
4. Minutes of the meeting of the Standing Committee of the Syndicate on Examinations held on [redacted] (Item No. 8)
5. Minutes of the meeting of the Committee dated [redacted] constituted for fixing mandatory number of answer scripts to be evaluated without remuneration.
6. Minutes of the meeting of the Standing Committee of the Syndicate on Examinations held on 31/08/2017 (Item No. 1c)
7. Minutes of the meeting of the Syndicate held on [redacted]

ORDER

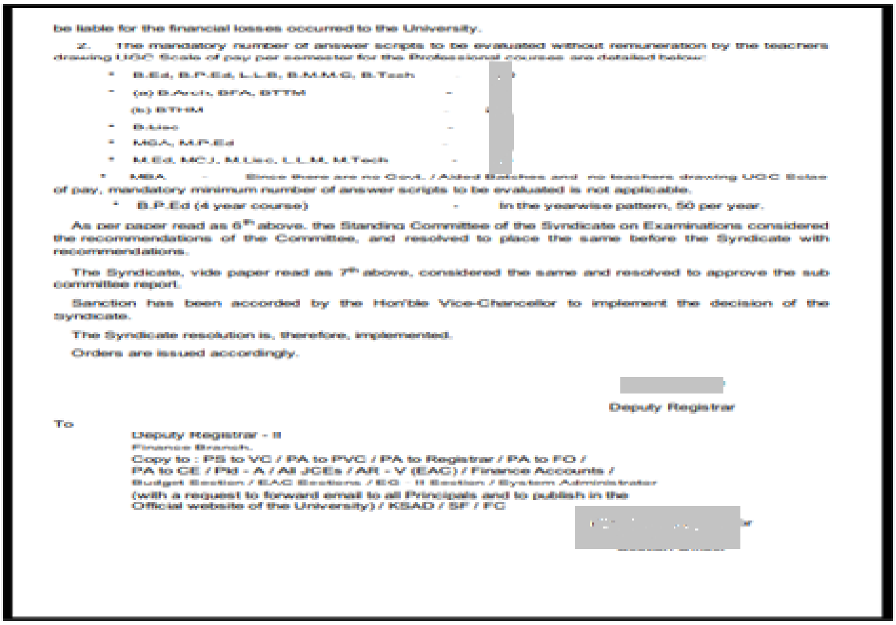
As per paper read as 1st above, the mandatory number of answer scripts to be evaluated by the teachers drawing UGC scale of pay as part of duty (free of remuneration) was fixed. As per paper read as 2nd above, the number of papers to be evaluated by a teacher per semester, mandatorily, without remuneration, for the Arts / Humanities and Science Courses was also fixed.

Since the mandatory number of scripts to be evaluated for Professional course was not mentioned in the UO's read 1 & 2 above, the question of fixing of mandatory number of answer scripts to be evaluated without remuneration by teachers drawing UGC Scale of pay in respect of Professional courses was placed before the Syndicate for consideration. As per paper read as 3rd above, the Syndicate held on 20/06/2017 considered the matter, vide item No. 2017.610", and deferred for detailed study by the Standing Committee on Examinations.

The Standing Committee of the Syndicate on Examinations as per paper read as 4th above, constituted a Committee for a detailed study of the matter. The Committee met on 03/08/2017, and as per paper read as 5th above, recommended the following:

1. As per U.O. No. [redacted] 29/03/2017, the mandatory number of answer scripts to be evaluated by the teachers drawing UGC Scale of Pay as part of duty without remuneration has been fixed. The Committee recommended that the valuation of answer scripts is part of duty of teachers. It is obligatory that every teacher should evaluate the answer scripts alloted to him/her by the University, of which the mandatory minimum number of answer scripts shall be evaluated without remuneration, as part of their duty. If a teacher abstains from evaluation duty, it will be considered a punishable offence and will

(a)



be liable for the financial losses occurred to the University.

2. The mandatory number of answer scripts to be evaluated without remuneration by the teachers drawing UGC Scale of pay per semester for the Professional courses are detailed below:

- B.Ed, B.P.Ed, L.L.B, B.M.M.C, B.Tech - [redacted]
- (a) B.Arch, BFA, BTTM -
  (b) BTHM -
- B.Lisc -
- MSA, M.P.Ed -
- M.Ed, MCJ, M.Lisc, L.L.M, M.Tech -
- MBA - Since there are no Govt. / Aided Batches and no teachers drawing UGC Scale of pay, mandatory minimum number of answer scripts to be evaluated is not applicable.
- B.P.Ed (4 year course) - In the yearwise pattern, 50 per year.

As per paper read as 6th above, the Standing Committee of the Syndicate on Examinations considered the recommendations of the Committee, and resolved to place the same before the Syndicate with recommendations.

The Syndicate, vide paper read as 7th above, considered the same and resolved to approve the sub committee report.

Sanction has been accorded by the Hon'ble Vice-Chancellor to implement the decision of the syndicate.

The Syndicate resolution is, therefore, implemented.

Orders are issued accordingly.

[redacted]
Deputy Registrar

To
Deputy Registrar - II
Finance Branch.
Copy to : PS to VC / PA to PVC / PA to Registrar / PA to FO /
PA to CE / Pld - A / All JCEs / AR - V (EAC) / Finance Accounts /
Budget Section / EAC Sections / EG - II Section / System Administrator
(with a request to forward email to all Principals and to publish in the
Official website of the University) / KSAD / SF / FC

[redacted]

(b)

Figure 3.2 (a, b) Document related to Examination remuneration

For the ease of processing the data set, the documents are converted into text files (.txt files). Therefore, as a whole, the entire corpus consists of two sets of documents, namely, Document Set and Query Set. The number of documents and queries in the different categories used in this study includes:

- Academics: It includes 1500 queries and associated 550 document files.
- Administration: This consists of 1800 queries and associated 660 document files.
- Examination: Includes 1800 queries files and associated 690 document files.
- Finance: Includes 1500 queries and associated 540 document files.
- Planning and Development: Includes 900 queries and associated 360 document files.

Graphical representation of the number and percentage of documents used for experimental purposes under each category is depicted in figures 3.3 and 3.4, respectively.

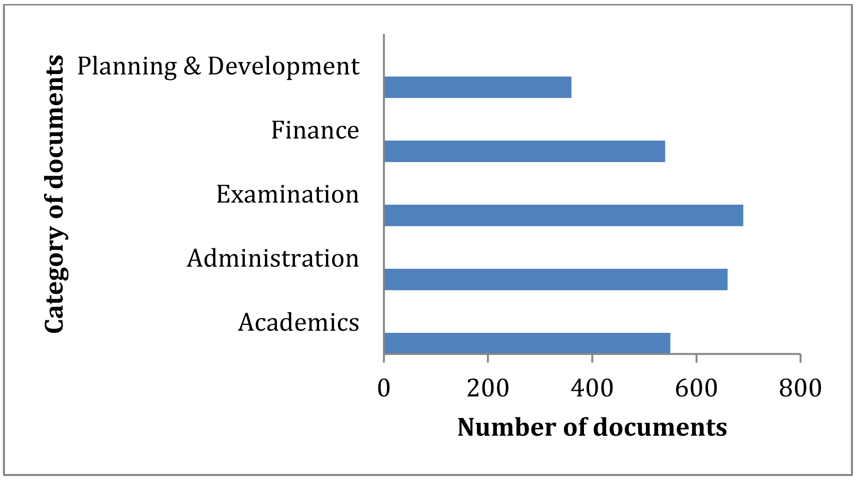


Figure 3.3: The number of documents under different categories

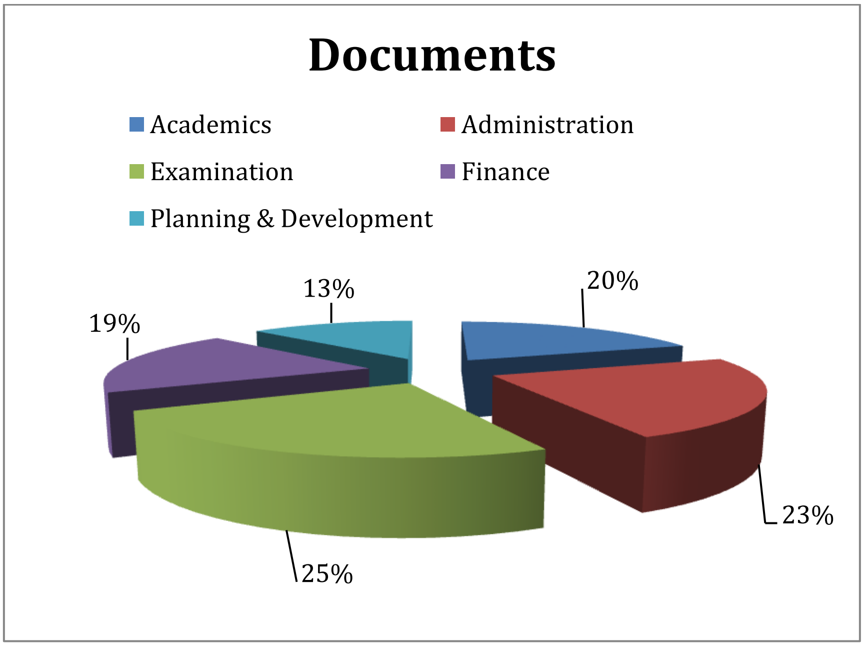


Figure 3.4: The percentage of documents under different categories

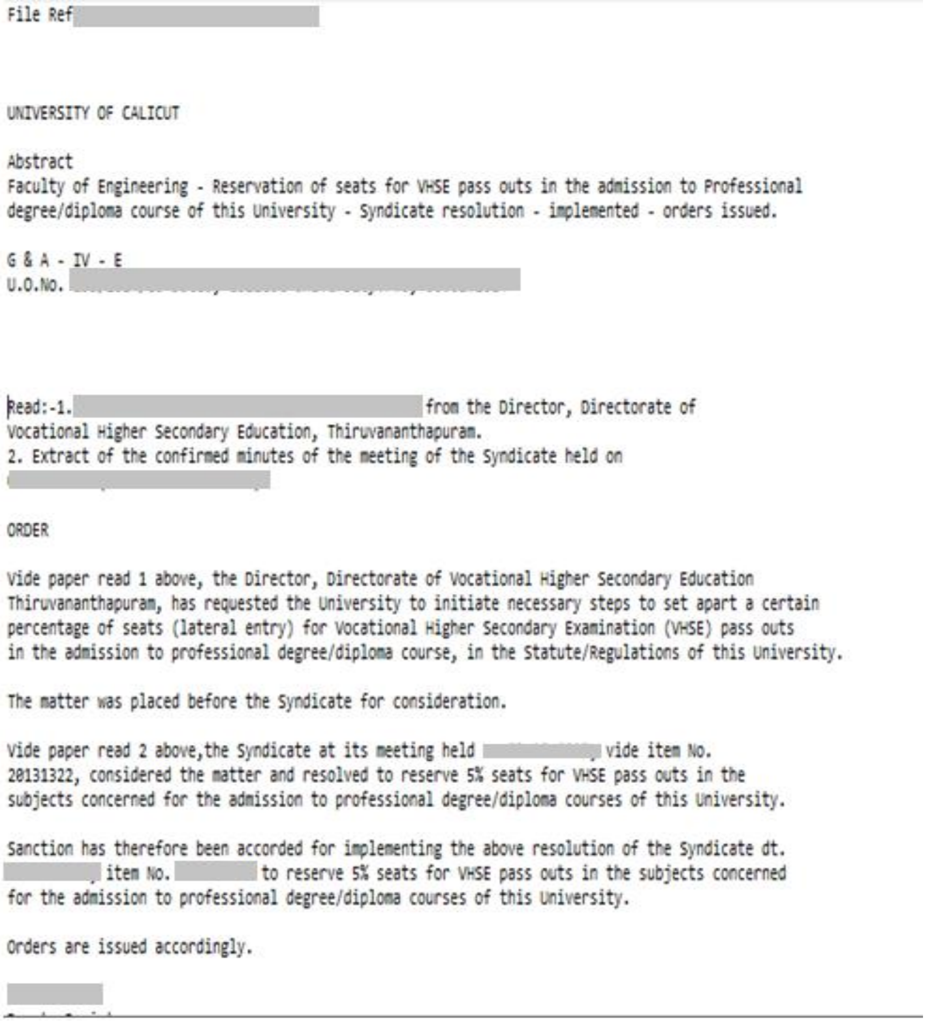
This document dataset is named University Document Data Set in English (UDDSE) and made available publically for research purposes [128].

3.4. Preprocessing on Document Data set

All data to be processed must first become encoded to be understandable by the computer. The quality of data affects the mining results. To improve the quality of data as well as the output, raw data is preprocessed to enhance efficiency. Therefore preprocessing plays a crucial role that deals with the preparation and transformation of the initial dataset and to resolve ambiguities in data.

There exist numerous preprocessing methods in data mining [129][130]. The most common approach is to reduce everything to lowercase for ease. Duly the entire data is converted into lower case letters, which are considered as the standard format in preprocessing text data. Natural language ToolKit (NLTK), is used as a tool for performing text preprocessing. It is a collection of codes and libraries for symbolic and statistical Natural Language Processing (NLP) for English. which is developed in Python programming language [131]. A sample document converted into a text file for further preprocessing is shown in figure 3.5.

The text preprocessing is used to prepare the input text for further processing, including tokenization, parsing, stop word removal, stemming, lemmatization is discussed in the following subsections.



```
File Ref[redacted]


UNIVERSITY OF CALICUT

Abstract
Faculty of Engineering - Reservation of seats for VHSE pass outs in the admission to Professional
degree/diploma course of this University - Syndicate resolution - implemented - orders issued.

G & A - IV - E
U.O.No. [redacted]



Read:-1.[redacted]from the Director, Directorate of
Vocational Higher Secondary Education, Thiruvananthapuram.
2. Extract of the confirmed minutes of the meeting of the Syndicate held on
[redacted]

ORDER

Vide paper read 1 above, the Director, Directorate of Vocational Higher Secondary Education
Thiruvananthapuram, has requested the University to initiate necessary steps to set apart a certain
percentage of seats (lateral entry) for Vocational Higher Secondary Examination (VHSE) pass outs
in the admission to professional degree/diploma course, in the Statute/Regulations of this University.

The matter was placed before the Syndicate for consideration.

Vide paper read 2 above,the Syndicate at its meeting held [redacted] vide item No.
20131322, considered the matter and resolved to reserve 5% seats for VHSE pass outs in the
subjects concerned for the admission to professional degree/diploma courses of this University.

Sanction has therefore been accorded for implementing the above resolution of the Syndicate dt.
[redacted] item No. [redacted] to reserve 5% seats for VHSE pass outs in the subjects concerned
for the admission to professional degree/diploma courses of this University.

Orders are issued accordingly.

[redacted]
```

Figure 3.5: A sample text file corresponding to an input document

3.4.1. Tokenization

Tokenization is almost always the first step in any NLP pipeline. Tokenization is the task of breaking the text into a more meaningful character sequence, groups, usually words or sentences. It is the process of breaking up the sequence of strings into pieces such as words, keywords, phrases, symbols, and other elements called tokens. The special symbols and punctuations are discarded at the earliest. Characters are difficult to interpret on their own by a computer, but words are typically self-contained semantic units. This provides a comfortable level of abstraction to work since a lot of NLP operations act directly on words. But among text, a majority of words are connecting parts of the sentence, rather than the subject or content of the text. The process of filtering out those unwanted words that do not contribute anything to the context of the text is called Stopword removal. NLTK has a list of stopwords stored in 16 different languages, including English [132].

The first step in tokenization would be simply to fragment by spaces and remove the punctuation, if any. Though it is a good baseline, it is not ideal. There are a number of tricky cases, even for everyday English. For example, the use of the apostrophe for contractions. “Shouldn’t” may be a single token or two ("should", "n’t"), meaning "should not". There may be different cases of hyphenation like "co-curricular" or "semester-based". There may be composite borrowed names like "Paper name" or “Grade point” or any mathematical notation. The handling of such corner cases is also ambiguous, depending on the desired use-case. In practice, tokenization is considered as a closed problem in academia [133].

It is common to have some unavoidable mess in language, but there are good efficient tokenizers that output relatively clean results. The current approach is relatively pragmatic. NLP libraries like NLTK offer a variety of language-specific tokenizers that consider most of the corner cases. Every implementation has different priorities, and one can choose the most suitable one for each use-case.

Implementation details of preprocessing in this study is detailed in table 3.1 with a sample input and output.

3.4.2. Normalization

Another step in data preprocessing is normalization. Often isolation of words is not enough for specific use-cases. Formatting and morphological variations can hinder matching, similarity tasks, or indexing. A linguistic model should be able to identify that "fee" and "fees" are the same substantive or "have" and "had" are the equal verbs. Normalization, in common, is the exercise of removing undesired variation from the text, so that trivial linguistic variances do not get in the way of matching what are effectively the same concepts. This is frequently done by eliminating or transforming some parts of the word so that only a common root is kept. However, normalization inherently leads to information loss, which is not always desirable.

It is more challenging to remove morphological variation. However, there are two distinct types of techniques for doing so. The goal of normalization is to ensure a standard format and property to the entire set of values, which helps in the prediction of data. Two methods in normalization are stemming and lemmatization.

3.4.3. Stemming

Stemming removes word suffixes, possibly recursively in layer after layer of processing. In terms of efficiency, stemming lessens the number of unique words in the index, which causes a reduction in the storage space essential for the index and accelerates the search process. In terms of effectiveness, stemming increases recall by reducing different word forms to a single stem form [134][135]

3.4.4. Lemmatization

An alternative approach from stemming to remove inflection is lemmatization, which is considered as a more intensive and slow process but more accurate. Lemmatization is an alternative method that uses human-curated dictionaries to extract the correct lemmas or roots from known words. A shared vocabulary used for this task is WordNet2 [136]. This approach is more accurate than stemming, but such dictionaries can be relatively limited, especially in domain-specific text. Lemmatization is also considerably more substantial than simple rule-based stemming. For maximum accuracy, it is not uncommon to use both methods in conjunction. Lemmatization works better to determine the text. Python NLTK provides WordNet Lemmatizer that uses the WordNet database to lookup lemmas of words.

3.4.5. Parsing

After isolating and cleaning words, usually, another layer of processing is added to work at higher abstraction levels. Parsing encompasses a set of annotation tasks to identify the role of each word with respect to its context in some text, usually a sentence. Parsing is usually the core of NLP tasks that require deep understanding. Probabilistic language parsers were one of the significant breakthroughs in NLP. Such parsers were one of the main components that enabled high-quality machine translation for the first time.

Part-Of-Speech tagging is considered as the first step in parsing which is the computational equivalent of morphological linguistic analysis. It annotates individual tokens with a morphological class, say substantive, adjective, or verb. Some taggers also go more in-depth and specify substantive properties, such as plurality, or tenses of the verb. POS tagging is usually fulfilled using a reference dictionary, a few morphological heuristics, and a disambiguation component. Even in dictionaries, words may have ambiguous morphologic types. An example: "she became the head of the team" and "he was injured on his head". Disambiguation is the most challenging part. Generally, disambiguation has been performed probabilistically by estimating co-occurrence probabilities. More recently, the state-of-the-art in disambiguation has been beaten by models using word embedding or end-to-end deep models [137].

The syntactical analysis is the second step of parsing. Much like in linguistic analysis, the syntax follows after morphological analysis. Here the role of each word in the sentence is determined and captures dependencies between each component. This is an even tougher challenge than POS tagging. Probabilistic methods were a good baseline, but high-quality results have not been achieved until recently before starting to use deep supervised models. Nevertheless, complete syntactical analysis is not required for many use-cases. For example, probabilistic models that extract subject-verb-object triplets have been operative for a while. For most information extraction tasks, such triples are already extremely useful.

3.4.6. Stop word Removal

A stop word list consists typically of those word classes known to convey any meaning, such as articles, conjunctions, interjections, prepositions, pronouns, and forms of the verb. The removal of stop words helps to increase processing speed, resource utilization by avoiding further processing, as well as possible matching, those terms that have little value in finding useful documents in response to a user query [134][135]. The converted text file is used for further process.

A sentence from the document is shown with major preprocessing steps in table 3.1.

Table 3.1: Preprocessing of a text

Preprocessing stages applied to the dataset			
Sl. No	**Preprocessing Stage**	**sample Input Text**	**Output**
1	Remove Punctuation	Reservation of seats for admission to professional-degree course	Reservation of seats for admission to professional degree course
2	Tokenization	Reservation of seats for admission to professional degree course	"Reservation", "of", "seats", "for", "admission", "to", "professional", "degree", "course"
3	Normalization	Reservation of seats for admission to professional degree course	Reserve of seat for admit to profession degree course
4	Lemmatization	Reservation of seats for admission to professional degree course	Reserve of seat for admission to professional degree course
5	Stop word removal	Reservation of seats for admission to professional degree course	Reserve, seats, admission, professional, degree, course

3.5. Query Data Acquisition and Dataset Preparation

There may be many doubts and queries from the students as well as employees on the process followed in a University in the natural language. The queries are normally answered based on the information provided in the doubts. That means these queries can be of Natural Language Queries, which are fabricated manually by different individuals. Therefore a Query data set is created, which consists of a set of document queries raised by each individual in the natural language. These query documents are also categorized into five categories: such as Academics, Administration, Examination, Finance, and Planning & Development. The query documents are converted into text files (.txt files) for the purpose of training the data set.

The query dataset is created by including the commonly asked general questions collected with the support of the University enquiry section. The short queries related to the University services are framed with the assistance of the staff and authorities working in the University enquiry branch. The queries are also framed based on the information collected directly from the students at the University campus. Some queries are also generated directly from the available documents. Hence a list of frequently asked queries related to University services is generated accordingly. Graphical representation of the number and the percentage of queries used for experimental purposes under each category is shown in figure 3.6 and 3.7 respectively.

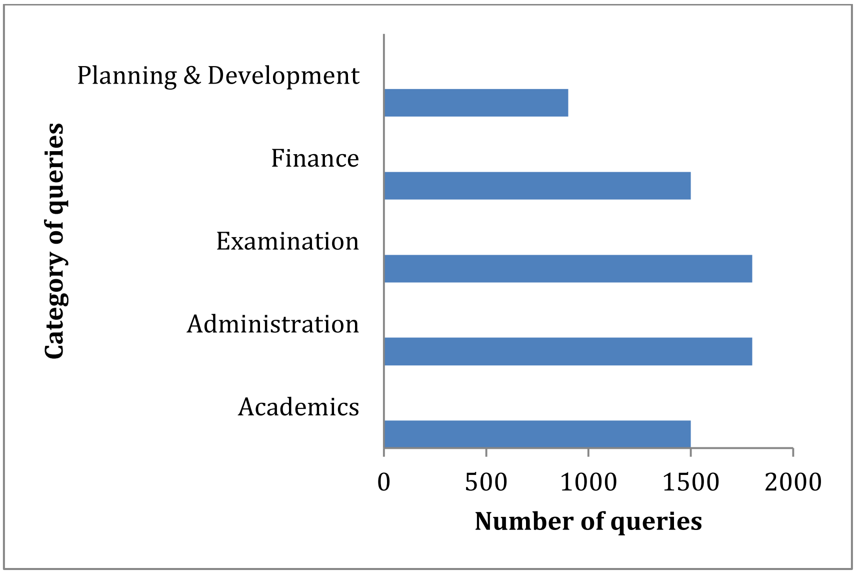


Figure 3.6: The number of queries under different categories

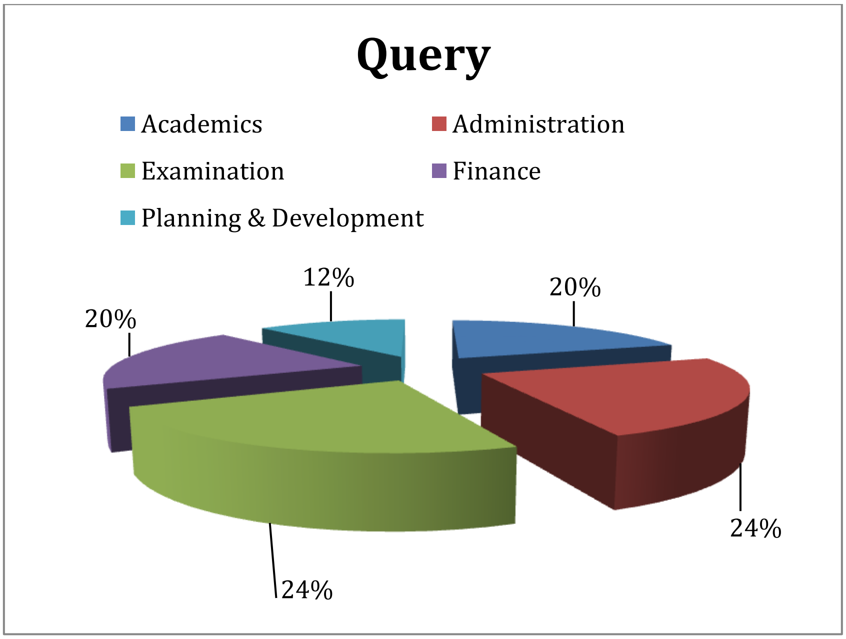


Figure 3.7: The percentage of queries under different categories

These queries are then digitized to text format. This dataset is named Short Queries for University Services in English (SQUSE) and made available publically for research purposes [138]. Sample queries taken from the own created short query dataset are given in Table 3.2.

Table 3.2: Sample queries taken from the SQUSE data set

Sl.No.	Natural language queries
1	How to apply for an open degree
2	How to apply for TC from the distance education office
3	Cana student under regular mode continue through distance education during the course
4	What is the fee for confidential mark lists
5	What is an additional degree
6	What is the difference between private registration and distance education
7	Finance order regarding the expenditure of the lecture program of the Department of Sanskrit.
8	Venue for training employees of the University.
9	If there is name correction necessitated, how can it be done
10	Who are the UG students of Nirmala College of Engineering, ChaIakudy avail condonation?
11	Purchased items by the Department of Geology.
12	Who is the Head of the Department of Commerce and Management Studies during the period from 23.02.2015?
13	What is the payment of Guest Lecturers with NET/Ph.D. qualification engaged on an hourly basis in the Department of Journalism and Mass Communication?
14	Amount sanctioned by Vice-Chancellor for C.H. Mohammed Koya Chair for Developmental studies.
15	Who are the approved research guides in the Dept of Computer Science

The five categories identified for the domain of interest include Academics, Administration, Examinations, Finance, Planning & Development. Thus the queries are also created related to these domains. The SQUSE data set consists of a total of 7500 queries that are to be fired against the 2800 documents

3.6. Preprocessing on Query Data Set

Preprocessing methods play a significant role in short query processing applications. It is the first step in Natural Language Processing (NLP) applications [139]. Data preprocessing refers to any processing performed on raw data to prepare it for another processing. The information retrieval process will be more difficult if there is a large amount of irrelevant and redundant information presents in the data set. Data preparation and filtering steps require a considerable amount of processing time. The data preprocessing includes tokenization, stop word removal, part of speech tagging, stemming, etc. The output of data preprocessing is the final training dataset. The significant preprocessing steps are done on the query data set, including tokenization, stop word removal, and stemming used as part of this study are described in the following sections

3.6.1. Tokenization

When a user inputs a query, the system must tokenize the query stream. In this process, the input text is broken down into understandable fragments. Typically a token is defined as an alpha-numeric string that occurs between white spaces and/or punctuations [140].

For example, the results of tokenization applied to the query *"**How to apply for an open degree?**"* is given in Table 3.3

Table 3.3: Result of tokenization

Tokens
How
to
apply
for
an
open
degree
?

3.6.2. Stop Word Removal

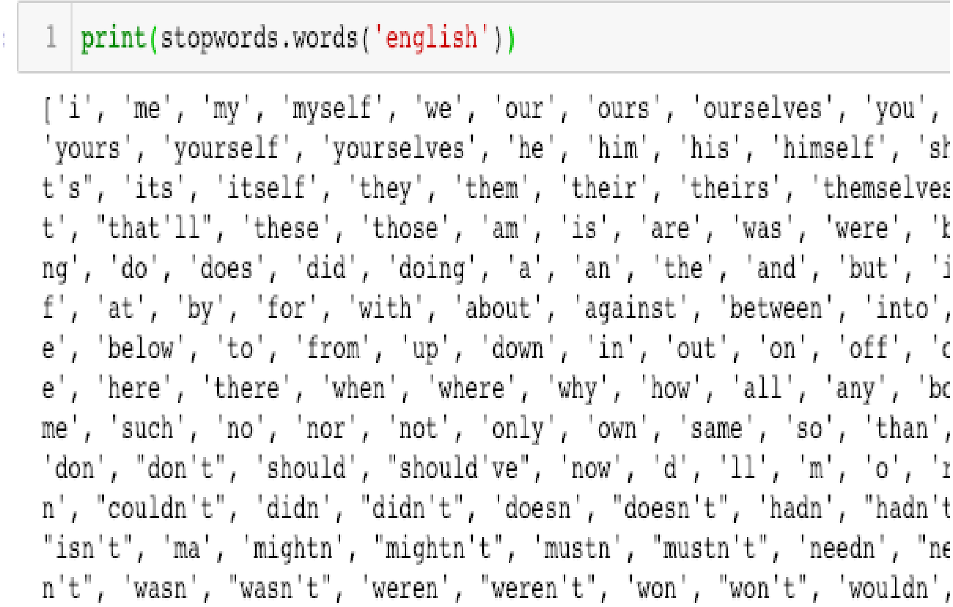
Stop words are not crucial for analyzing the content of a document as they have little meaning. Removing the stop words reduces the dimension of term space. In general, commonly found stop words in a text document are articles, prepositions, pronouns, etc. Stop words are removed from documents because those are not considered as keywords in the short query processing application. NLTK in Python has lists of stop words stored in different languages. Since the dataset considered in this study is in English language, the stop words used in English are only considered in this study. Figure 3.8 shows the list of major stop words used in this study.

The following functions in Python are used to set the stop words.

fromnltk.corpus import stopwords

stopWords = stopwords.words('english')

Now the stopWords is a list of 179 words.



```
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'sh
t's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves
t', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'b
ng', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'i
f', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into',
e', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'o
e', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'bo
me', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than',
'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 'r
n', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't
"isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "ne
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn',
```

Figure 3.8: List of Stop words

After setting the stop words, they can easily be filtered from the query dataset and hence from the query processing steps. In the given sample query, it can be seen that *how, to* and *for* are stop words and will be filtered out for further processing. Hence the tokens in Table 3.3 are reduced to *apply, open,* and *degree* which are considered for further processing.

3.6.3. Stemming

Stemming is the process of getting the stem for the given word. The stem of a word will get after the removal of prefixes and suffixes called affixes. Stemming reduces the variant word form to a single stem form. For example, the result of the stemming applied on the word "apply" is given in table 3.4

Table 3.4: The result of stemming applied to the word "Apply."

Sl. No	Variant forms of the word Apply
1	Apply
2	Applied
3	Applying
4	Applies

When the user query includes contiguity, closeness, or Boolean operators, the system needs to parse the query into operators and query terms. These operators may present in the form of reserved punctuation. The system will identify the operators implicitly in the language used and not bother how the operators might be expressed, for example, prepositions, conjunctions, ordering, etc. [141].

In the information retrieval process to enhance the performance, queries are modified by query expansion which consists of finding synonyms of words, fixing spelling mistakes, etc. [142]. These synonyms were chosen using WordNet, a lexical database for English which is briefly explained in the next section.

3.7. WordNet

WordNet is used to build the lexicon for our query-document semantic information retrieval. WordNet is a lexical database for English created at Princeton University [23]. In WordNet, words are the basic units. But it also contains phrasal verbs, collections, and compounds as well as idiomatic phrases. The primary uses of WordNet are computational linguistics research. A massive amount of lexical knowledge is available in the WordNet that could support Natural Language Processing (NLP) research better than other traditional dictionaries.

There is an excellent variety of semantic relations that can be defined between word forms and between word meanings, but only a small collection of them is used in WordNet. WordNet contains different types of relations, such as Synonymy, Antonymy, Hyponymy, Hypernymy, Meronymy Holonymy, Troponymy, and Entailment.

Synonymy is used to form the sets of synonyms to represent meanings of words. Then the word forms will have a symmetric relation between them. Antonymy is also a symmetrical relation between word forms and is essential while organizing the meanings of adjectives and adverbs.

Hyponymy is a specification relation, whereas its inverse hypernymy is a generalization relation. Meronymy represents part of a relation while holonymy represents the whole of a relation. Just as hyponymy is used with nouns, troponymy is used with verbs for a manner name.

3.8. Conclusion

Real-world text is cumbersome to deal with. Natural Languages are highly inconsistent, even in the most formal settings. All data to be processed must first become encoded to be understandable by the computer. Real-world data is often incomplete, inconsistent, and or lacking in certain behaviors or trends and is more prone to ambiguity and errors. Therefore, the following preprocessing steps are done on the document and query dataset used in the experimental purpose.

- Conversion of pdf files into text files
- Conversion of the content of text files into lowercase letters
- Removal of punctuations and regular expressions
- Tokenization
- Stopword removal
- Normalization

The first part of this chapter discusses the creation of a document data set. The created data set consists of 2800 documents, which contains information regarding the domain of interest of this study. These unstructured documents are preprocessed for improved performance. The next part of this chapter explains the acquisition and preprocessing of the query dataset. The various preprocessing techniques applied in the query corpus are explained in detail. As a contribution of this study, two dataset named University Document Data Set in English (UDDSE) and Short Queries for University Services in English (SQUSE) are developed and are presently available publically for research purposes.

Chapter 4

Document and Query Similarity Indexing using TF-IDF and Fuzzy Hybridization

4.1. Introduction

Text retrieval is the core of Information Retrieval. Most of the Natural Language applications deal with the automatic detection of semantic text similarity between documents. There are two wide categories of techniques used in text retrieval such as structure based and structure agnostic. In a structure based approach, to do the comparisons, the logical structure of the text is transformed into an intermediate structure like trees. But it may not always be predefined as to which structure is suitable for a particular context and for a certain comparison. In the other category, the text is represented using a Vector Space Model (VSM) by ignoring its structure [143].

In this study unstructured documents and queries are used for the conduct of information retrieval experiments. . Hence to retrieve the most similar document for the given user query, the documents are retrieved in the order of similarity ranking. In the Vector Space Model, the text corresponding to documents and queries are converted into a numeric vector. Definition and number of dimensions are the key aspects of

VSM. The objective of the work prescribed in this chapter is to find out the most similar document from the set for the given user query.

The rest of the chapter is organized as follows. Section 4.2 gives an overview of vector space representation of the documents and queries used in this study and methods used for measuring the similarity. The studies on Document and Query representation using TF-IDF, Word2Vec, and Doc2Vec are given in the sub sections. Topic modelling using GloVe and Latent Dirichlet Allocation (LDA) for query classification is described in section 4.3. A comparative analysis is done in section 4.4. Fuzzy logic is incorporated to deal with the vagueness and is described in section 4.5. Section 4.6 presents the experimental analysis and results. Finally section 4.7 gives the concluding remarks.

4.2. Vector Space Representation and Similarity Measures

Computing weights for the terms in a query referred to as Query Term Weighting is the final step in query processing [144]. Occasionally the user controls this step by indicating either how much to weight each term or simply which term or concept in the query matters most and must appear in each retrieved document to ensure efficiency. Few systems implement system-based query weighting, but some do an implicit weighting by considering the first terms in a query as having higher priority. The search engines use this information to provide a list of documents to the user. After this step, the weighted query is searched against the inverted file of documents.

More number of documents and greater length of the documents make the process more expensive for processing in terms of computational resources and responsiveness. However, the longer the wait for results, the higher the quality of results. Thus, search system designers must choose what is most important between their user's time or quality [144].

Similarity measure is a function used to compute the degree of similarity between a pair of vectors or documents. Since both queries and documents are represented as vectors considering the terms used in them, the similarity between the queries and documents can be represented with a similarity measure [145].

The different steps in the development of query processing systems are text pre-processing, vector space representation and similarity measures. The pre-processing methods applied on document data set and query data set has already been described in detail in section 3.4 and 3.6 of Chapter 3.

The following section describes the vector space representation methods and the use of similarity measures applied on these documents and queries. Cosine and Jaccard similarity measures are used to rank the similar document-query pairs.

Experiments are conducted using the data set consisting of the queries related to different Departments/Branches under the University and the general queries related to the University system. The dataset includes a Query and corresponding Document set. Both Query and Document set is subdivided into five separate categories corresponding to the various branches of the University including Academics, Administration, Examination, Finance and Planning & Development as explained in section 3.2 of Chapter 3.

Since the machines cannot work on the text as it is, texts are represented as numbers such as a set of independent units like unigrams, bigrams or multi-grams which acts as a building block of the feature space. An *n*-gram represents a contiguous sequence of *n* items from a given text. An *n*-gram of size 1 is referred to as a "unigram"; size 2 is a bigram and so on.

The database used for this study consists of documents and queries related to Universities. The document set UDDSE consists of orders released by different universities or other officials related to higher education all over India, for the wellbeing of students and employees on academic and administrative affairs. These documents are normally available at the University websites, Government websites, which are orders, and notices generally in Portable Document Format (pdf).

There may arise many doubts and queries from the students as well as employees, which can be answered based on the information available in the related documents mentioned above. Usually people ask their doubts in their regional language or natural language. That means these queries can be of Natural Language Queries, which are fabricated manually by different individuals. Therefore a Query set is created, which consists of a set of document queries in natural language made by each individual.

4.2.1. Document and Query Representation using TF-IDF

Each term in the vocabulary is signified by a distinct and orthogonal dimension in the simple TFIDF (Term Frequency-Inverse Document frequency) approach. In TFIDF the term frequency of each term in a text is measured and multiplies it by the logged inverse document frequency of that term across the entire corpus.

The TFIDF suffers from an ignorance of n-gram phrases, difficulties with the modification like addition of new documents and a large number of dimensions. Term frequency *tf(t,d)* can be defined as the number of times the term ***t*** occurs in document ***d***. It is a measure of how many times the terms present in our vocabulary are present in the documents. So it can be defined as a counting function.

$$tf(t,d)=\sum_{x\in d} fr(x,t) \tag{4.1}$$

Where $fr(x,t)$ is a simple function defined as :

$$fr(x,t)=\begin{cases}1, \; if \; x=t\\ 0, otherwise\end{cases} \tag{4.2}$$

Since the document length varies, to normalize, usually term frequency is divided by document length.

$$TF(t,d)=\frac{\text{Number of times term } t \text{ apper in a document}}{\text{Total number of terms in the document}} \tag{4.3}$$

Inverse document frequency *idf (t, D)* is the measure of how much information a word gives in the document collection *D*, indicating whether the word is rare or shared across the document corpus [146]. Mathematically this is computed as the logarithmically scaled inverse fraction of the documents that contain the word.

$$IDF(t,d)=\log_e \frac{\text{Total Number of documents}}{\text{Number of documents } D \text{ with term } t \text{ in it}} \tag{4.4}$$

More precisely,

$$idf(t,d) = \log \frac{|D|}{1+|\{d : t \in d\}|} \qquad (4.5)$$

,

where

$|\{d : t \in d\}|$ represents the number of documents with the term t appears and 1 is added to avoid the division by zero error.

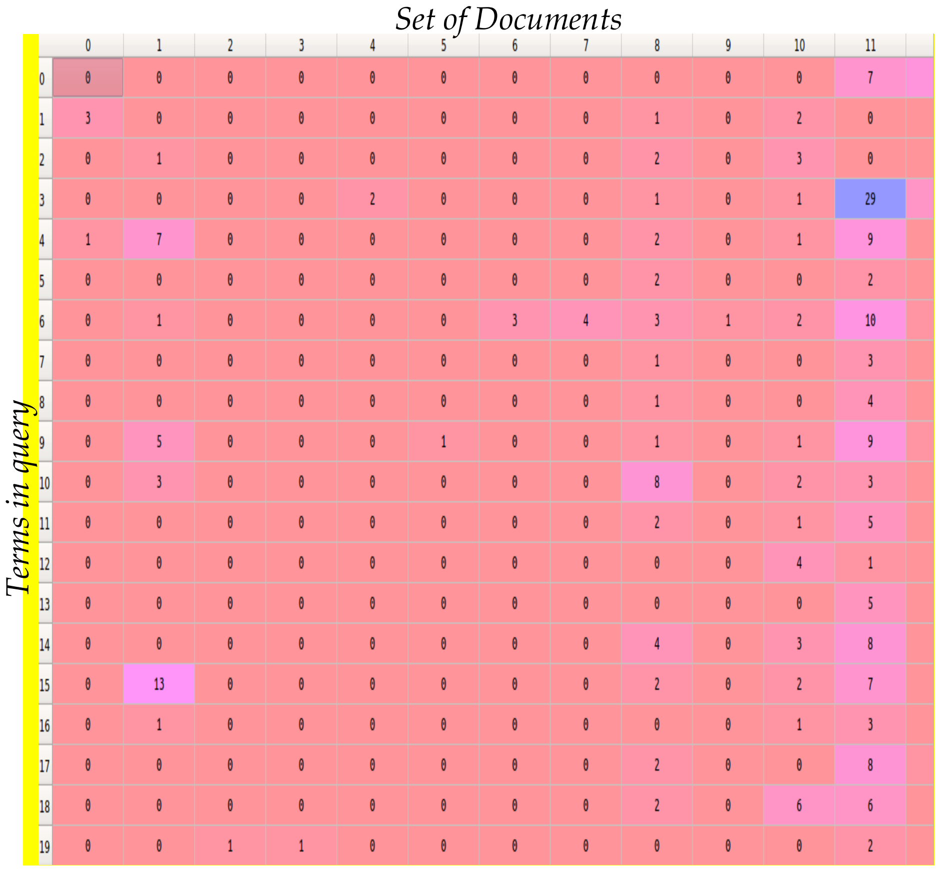
Thus the formula for $tf - idf(t,d)$ is

$$tf - idf(t,d) = tf(t,d) \times idf(t,d) \qquad (4.6)$$

Bag of Words or BoW is a traditional term frequency – inverse document frequency (*tf-idf*) approach to model the documents. *tf-idf* is a numerical statistics which focuses on how a word is important to the document in a corpus. In general, *tf-idf* is the product of term frequency (*tf*) weights and inverse document frequency (*idf*) weights [146].

A document can be represented by a bag of words in the form of a word-document matrix. Initially, the documents are preprocessed to remove the punctuation and stopwords. Preprocessing the documents also include stemming and lemmatization. As a result, some words are formed which are called dictionaries or vocabulary. Then it models each of the documents by computing the number of times each term occurs. BoW represents text by ignoring its order and grammar.

Figure 4.1 shows the BoW representation of certain terms in each document in the data set. In this figure, each row represents a term while each column denotes a document. The value $W_{i,j}$ in the matrix shows the number of times term i appear in document j.


Set of Documents
Terms in query

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	7
1	3	0	0	0	0	0	0	0	1	0	2	0
2	0	1	0	0	0	0	0	0	2	0	3	0
3	0	0	0	0	2	0	0	0	1	0	1	29
4	1	7	0	0	0	0	0	0	2	0	1	9
5	0	0	0	0	0	0	0	0	2	0	0	2
6	0	1	0	0	0	0	3	4	3	1	2	10
7	0	0	0	0	0	0	0	0	1	0	0	3
8	0	0	0	0	0	0	0	0	1	0	0	4
9	0	5	0	0	0	1	0	0	1	0	1	9
10	0	3	0	0	0	0	0	0	8	0	2	3
11	0	0	0	0	0	0	0	0	2	0	1	5
12	0	0	0	0	0	0	0	0	0	0	4	1
13	0	0	0	0	0	0	0	0	0	0	0	5
14	0	0	0	0	0	0	0	0	4	0	3	8
15	0	13	0	0	0	0	0	0	2	0	2	7
16	0	1	0	0	0	0	0	0	0	0	1	3
17	0	0	0	0	0	0	0	0	2	0	0	8
18	0	0	0	0	0	0	0	0	2	0	6	6
19	0	0	1	1	0	0	0	0	0	0	0	2

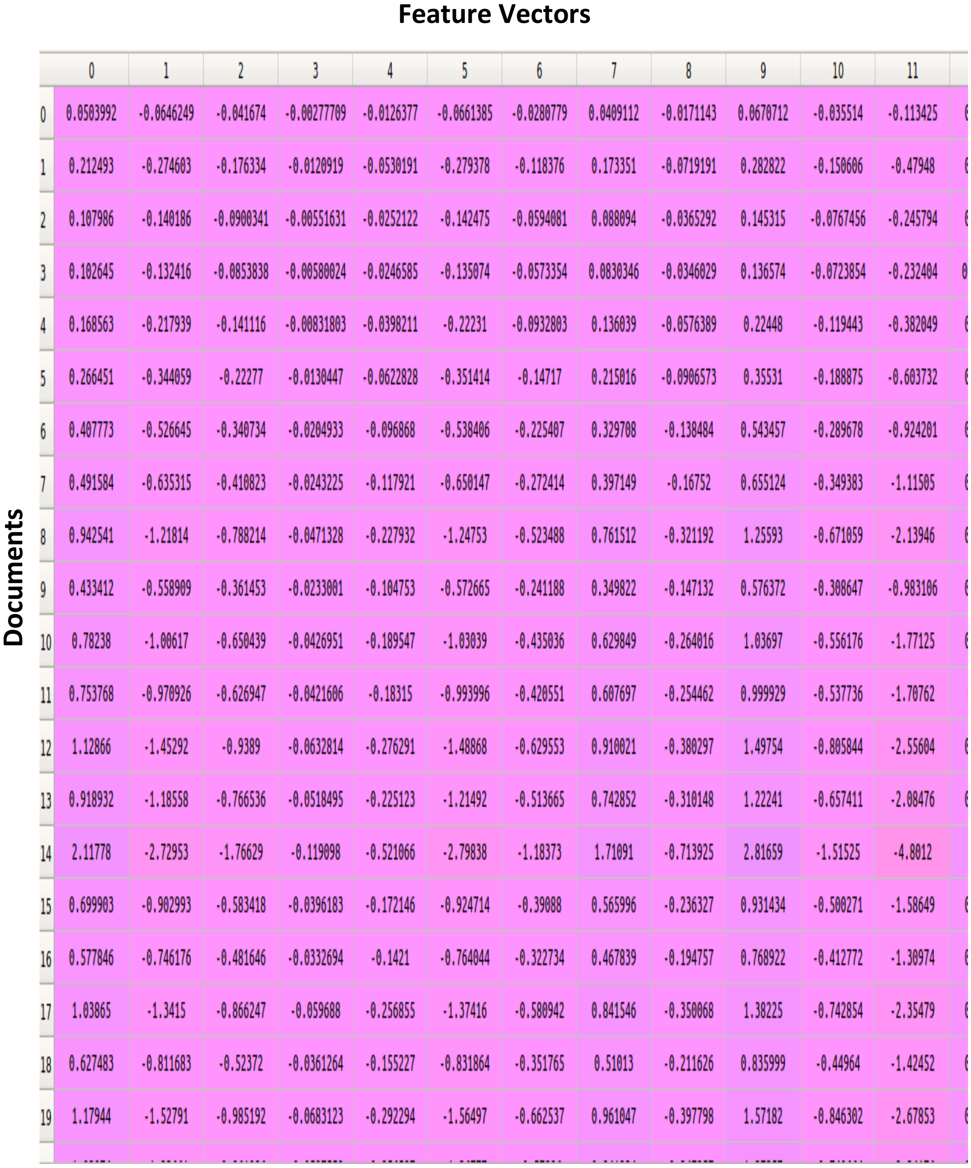
Figure 4.1: BoW representation of terms in query against document set

For example, in the figure 4.1, $W_{3,11} = 29$ represents the number of times the word represented by 3 appears in document 11 of the set which is 29. Term document matrix is considered as a simplified representation of documents as the input to the topic model. If there are w words and n documents in an input then BoW should be a $n \times w$ matrix.

4.2.2. Document and Query Representation Using Word2Vec

Word2Vec is a group of related models that are used to produce word embedding. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2Vec is probably the most popular learning model for the generation of dense embedding based on Distributional Hypothesis by Harris *et.al,.* [147]. It states that words occurring in similar contexts tend to have similar meanings. It also allows learning vector representations of words referred to as word embedding. The main aim of using the Word2Vec model is to learn the distributed representation for each target word by defining the context. The main advantage is that each dimension of the embedding represents a latent feature of the word and simple vector similarity operations can be computed using cosine similarity. The model is initialized with a minimum count for the input words such as the terms whose frequency was lower than 20 were discarded. The model was trained using the Continuous Bag of Words (CBOW) algorithm since it is more suitable for larger datasets. The way CBOW works is that it is likely to predict the probability of a word given a context [148]. A context may be a single word or a group of words. The dimensionality of feature vectors is fixed at 200. Once constructed the vocabulary and trained the input data, the learned word vector representations were performed on the unseen test set documents. The centroid *c* for each document *d*, is computed where ed_i is the i th embedding in *d*, so as to obtain a meaningful topic representation for each document [149]. Figure 4.2 shows the Word2Vec representation of the documents in the data set with feature vectors fixed as 200.

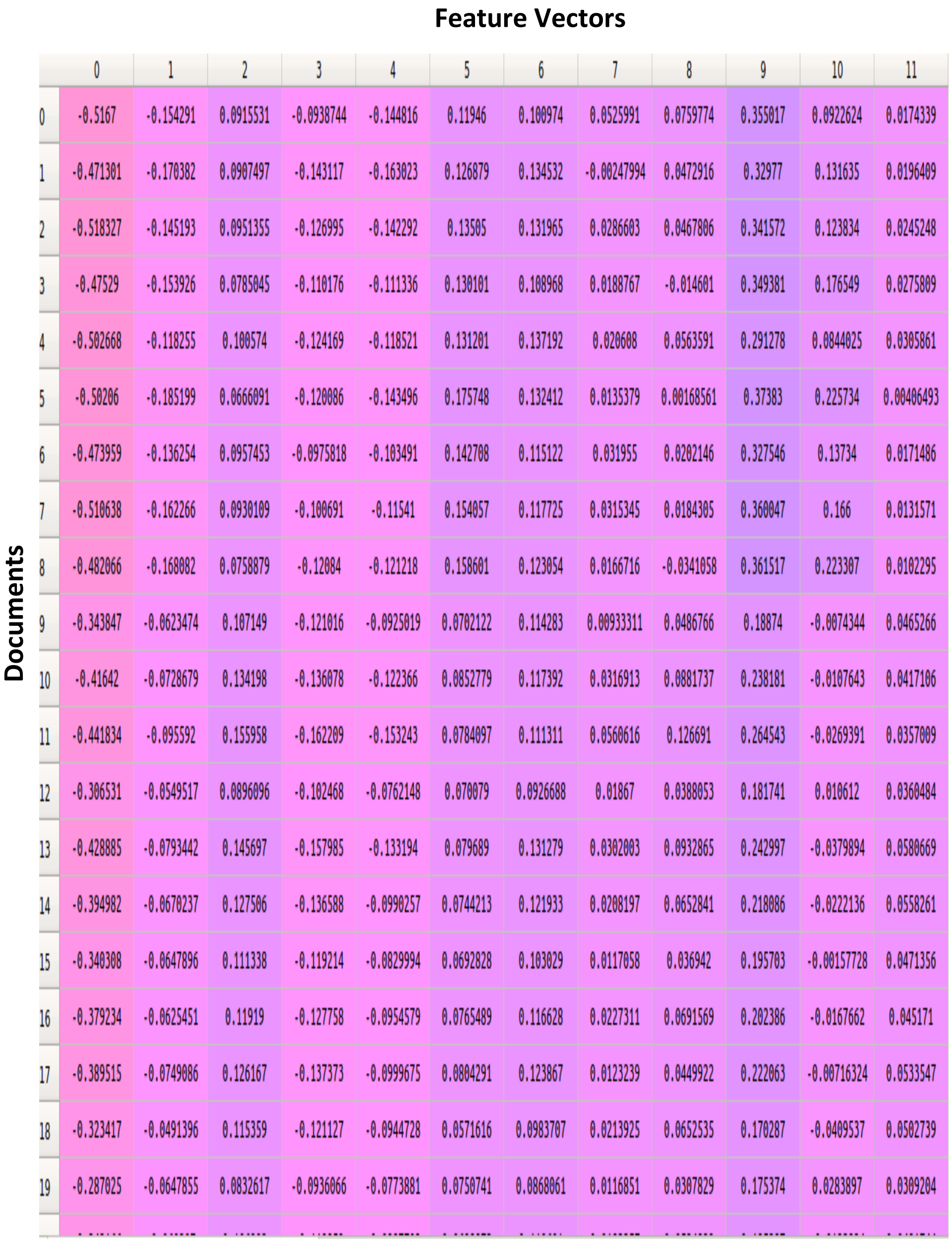


	0	1	2	3	4	5	6	7	8	9	10	11
0	0.0503992	-0.0646249	-0.041674	-0.00277709	-0.0126377	-0.0661385	-0.0280779	0.0409112	-0.0171143	0.0670712	-0.035514	-0.113425
1	0.212493	-0.274603	-0.176334	-0.0120919	-0.0530191	-0.279378	-0.118376	0.173351	-0.0719191	0.282822	-0.150606	-0.47948
2	0.107986	-0.140186	-0.0900341	-0.00551631	-0.0252122	-0.142475	-0.0594081	0.088094	-0.0365292	0.145315	-0.0767456	-0.245794
3	0.102645	-0.132416	-0.0853838	-0.00580024	-0.0246585	-0.135074	-0.0573354	0.0830346	-0.0346029	0.136574	-0.0723854	-0.232404
4	0.168563	-0.217939	-0.141116	-0.00831803	-0.0398211	-0.22231	-0.0932803	0.136039	-0.0576389	0.22448	-0.119443	-0.382049
5	0.266451	-0.344059	-0.22277	-0.0130447	-0.0622828	-0.351414	-0.14717	0.215016	-0.0906573	0.35531	-0.188875	-0.603732
6	0.407773	-0.526645	-0.340734	-0.0204933	-0.096868	-0.538406	-0.225407	0.329708	-0.138484	0.543457	-0.289678	-0.924201
7	0.491584	-0.635315	-0.410823	-0.0243225	-0.117921	-0.650147	-0.272414	0.397149	-0.16752	0.655124	-0.349383	-1.11505
8	0.942541	-1.21814	-0.788214	-0.0471328	-0.227932	-1.24753	-0.523488	0.761512	-0.321192	1.25593	-0.671059	-2.13946
9	0.433412	-0.558909	-0.361453	-0.0233001	-0.104753	-0.572665	-0.241188	0.349822	-0.147132	0.576372	-0.308647	-0.983106
10	0.78238	-1.00617	-0.650439	-0.0426951	-0.189547	-1.03039	-0.435036	0.629849	-0.264016	1.03697	-0.556176	-1.77125
11	0.753768	-0.970926	-0.626947	-0.0421606	-0.18315	-0.993996	-0.420551	0.607697	-0.254462	0.999929	-0.537736	-1.70762
12	1.12866	-1.45292	-0.9389	-0.0632814	-0.276291	-1.48868	-0.629553	0.910021	-0.380297	1.49754	-0.805844	-2.55604
13	0.918932	-1.18558	-0.766536	-0.0518495	-0.225123	-1.21492	-0.513665	0.742852	-0.310148	1.22241	-0.657411	-2.08476
14	2.11778	-2.72953	-1.76629	-0.119098	-0.521066	-2.79838	-1.18373	1.71091	-0.713925	2.81659	-1.51525	-4.8012
15	0.699903	-0.902993	-0.583418	-0.0396183	-0.172146	-0.924714	-0.39088	0.565996	-0.236327	0.931434	-0.500271	-1.58649
16	0.577846	-0.746176	-0.481646	-0.0332694	-0.1421	-0.764044	-0.322734	0.467839	-0.194757	0.768922	-0.412772	-1.30974
17	1.03865	-1.3415	-0.866247	-0.059688	-0.256855	-1.37416	-0.580942	0.841546	-0.350068	1.38225	-0.742854	-2.35479
18	0.627483	-0.811683	-0.52372	-0.0361264	-0.155227	-0.831864	-0.351765	0.51013	-0.211626	0.835999	-0.44964	-1.42452
19	1.17944	-1.52791	-0.985192	-0.0683123	-0.292294	-1.56497	-0.662537	0.961047	-0.397798	1.57182	-0.846302	-2.67853

Figure 4.2: Word2Vec representation with Feature Vectors

4.2.3. Document and Query Representation using Doc2Vec

Doc2Vec is an unsupervised algorithm to generate vectors for documents. This algorithm is an adaptation of Word2Vec which can generate vectors for words. The vectors generated by Doc2Vec can be used for tasks like finding the similarity between documents. This model randomly sample consecutive words from a paragraph and *predict a centre word* from the randomly sampled set of words by taking as *input the context words and a paragraph id from the desired documents.* The Doc2Vec representation of documents from the dataset with feature vectors is shown in Figure 4.3



	0	1	2	3	4	5	6	7	8	9	10	11
0	-0.5167	-0.154291	0.0915531	-0.0938744	-0.144816	0.11946	0.100974	0.0525991	0.0759774	0.355017	0.0922624	0.0174339
1	-0.471301	-0.170382	0.0907497	-0.143117	-0.163023	0.126879	0.134532	-0.00247994	0.0472916	0.32977	0.131635	0.0196409
2	-0.518327	-0.145193	0.0951355	-0.126995	-0.142292	0.13505	0.131965	0.0286603	0.0467806	0.341572	0.123834	0.0245248
3	-0.47529	-0.153926	0.0785045	-0.110176	-0.111336	0.130101	0.108968	0.0188767	-0.014601	0.349381	0.176549	0.0275809
4	-0.502668	-0.118255	0.100574	-0.124169	-0.118521	0.131201	0.137192	0.020608	0.0563591	0.291278	0.0844025	0.0305861
5	-0.50206	-0.185199	0.0666091	-0.120086	-0.143496	0.175748	0.132412	0.0135379	0.00168561	0.37383	0.225734	0.00406493
6	-0.473959	-0.136254	0.0957453	-0.0975818	-0.103491	0.142708	0.115122	0.031955	0.0202146	0.327546	0.13734	0.0171486
7	-0.510638	-0.162266	0.0930109	-0.100691	-0.11541	0.154057	0.117725	0.0315345	0.0184305	0.360047	0.166	0.0131571
8	-0.482066	-0.168082	0.0758879	-0.12084	-0.121218	0.158601	0.123054	0.0166716	-0.0341058	0.361517	0.223307	0.0102295
9	-0.343847	-0.0623474	0.107149	-0.121016	-0.0925019	0.0702122	0.114283	0.00933311	0.0486766	0.18874	-0.0074344	0.0465266
10	-0.41642	-0.0728679	0.134198	-0.136078	-0.122366	0.0852779	0.117392	0.0316913	0.0881737	0.238181	-0.0107643	0.0417106
11	-0.441834	-0.095592	0.155958	-0.162209	-0.153243	0.0784097	0.111311	0.0560616	0.126691	0.264543	-0.0269391	0.0357009
12	-0.306531	-0.0549517	0.0896096	-0.102468	-0.0762148	0.070079	0.0926688	0.01867	0.0388053	0.181741	0.010612	0.0360484
13	-0.428885	-0.0793442	0.145697	-0.157985	-0.133194	0.079689	0.131279	0.0302003	0.0932865	0.242997	-0.0379894	0.0580669
14	-0.394982	-0.0670237	0.127506	-0.136588	-0.0990257	0.0744213	0.121933	0.0208197	0.0652841	0.218086	-0.0222136	0.0558261
15	-0.340308	-0.0647896	0.111338	-0.119214	-0.0829994	0.0692828	0.103029	0.0117058	0.036942	0.195703	-0.00157728	0.0471356
16	-0.379234	-0.0625451	0.11919	-0.127758	-0.0954579	0.0765489	0.116628	0.0227311	0.0691569	0.202386	-0.0167662	0.045171
17	-0.389515	-0.0749086	0.126167	-0.137373	-0.0999675	0.0804291	0.123867	0.0123239	0.0449922	0.222063	-0.00716324	0.0533547
18	-0.323417	-0.0491396	0.115359	-0.121127	-0.0944728	0.0571616	0.0983707	0.0213925	0.0652535	0.170287	-0.0409537	0.0502739
19	-0.287025	-0.0647855	0.0832617	-0.0936066	-0.0773881	0.0750741	0.0868061	0.0116851	0.0307829	0.175374	0.0283897	0.0309204

Figure 4.3: Doc2Vec representation

4.3 Topic Modelling using GloVe and LDA for Query Classification

Query classification (QC) is the task of automatically labeling user queries into a given target taxonomy which is needed to identify the type of document and to get an insight on the contents of the documents [150]. Query classification can help the information providers understand users' needs based on the categories that the users are searching for. Word embedding using GloVe algorithm and Topic modeling using Latent Dirichlet Allocation (LDA) algorithm on the dataset is carried out in this study. The natural language queries from the five different categories of the University domain are considered for experimental purposes. The preprocessed queries are brought into a Python platform in Anaconda and Spyder IDE for further experimentation using the Glob module. Anaconda is free, an open-source distribution of Python and R programming languages for scientific computing that aims to simplify package management and deployment. Spyder stands for 'Scientific PYthon Development EnviRonment' and is included in the Anaconda distribution [151].

The Glob module in Python finds all the path names matching a specified pattern according to the rules used by the Unix shell. The text data in the files are tokenized using WordNet Tokenizer, a tool from Natural Language Toolkit

(NLTK). Topic Modelling is a technique used to extract the underlying topics from a large volume of text. Gensim package provides support for implementing the Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing (LSI) algorithms and the necessary sophistication to build high-quality topic models. The plain text files obtained as a result of preprocessing which is discussed in chapter 3 are used for further processing. GloVe tool and LDA algorithm are applied on these files in order to do the iterations when the sliding window is used to determine the context of each word in vocabulary as well as each topic from the corpus.

To train the model on GloVe, the parameters such as window size, the number of components, learning rate, epochs and number of threads are set initially with values 10, 100, 0.05, 5 and 10 respectively and with the parameter *'verbose'* set to '*true*'. The process is repeated on each five categories of queries and documents with a different number of iterations.

Another approach used is to train the preprocessed corpus named SQUSE in the LDA model. The model has trained on all the query sets of the five categories of the corpus. The text files obtained after preprocessing act as the input for training the model in LDA while parameters are quite different from GloVe models. Soon after text preprocessing, the 'bag-of-words' model and dictionary are created which are the fundamental parameters of the LDA model. To train the model, the parameters such as the bag-of-words representation of the corpus, chunk size, random state, and the number of iterations are set as *query_corpus, 10000, 100, 10* respectively. The most vital tuning parameter for LDA models is the number of topics which is initially set to 10 for the study. Some topics will share common keywords which results in the reduced number of distinct topics. Soon after each iteration, the topic coherence and perplexity of topics are evaluated. Higher the topic coherence, more human interpretable the model will be and lower the perplexity value, better the model.

The model is trained on each category with a varying number of iterations. The number of topics that should be extracted from the dataset is set to 10 as the major topics to be identified in this context are 5. But not all the document consists of exactly the 5 topics identified for this study. Therefore the top ten topics from the query files are extracted. To classify a query as fitting to a specific topic, a logical approach is to find out which topic has the highest influence on that query and assign it. The topic weight is the method to find out which topic has the highest contribution to the document. The weight of each keyword in each document can be obtained as a two dimensional matrix. Table 4.1 gives the topic weights for each query for a sample of 15 queries.

Table 4.1: Query Topic Weights

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	Dominant_topic
Q0	0	0	0	0	0	0	0.27	0.1	0.69	0	8
Q1	0	0	0	0.87	0	0	0	0	0.49	0.14	3
Q2	0	0.28	0.94	0.21	0	0	0	0	0	0.35	2
Q3	0	0	0	0	0	0.53	0	0.13	0.55	0	8
Q4	0	0	0.53	0	0	0.83	0	0.88	0.22	0	7
Q5	0.15	0	0	0	0.49	0	0	0	0	0	4
Q6	0	0.71	0	0.08	0.06	0	0	0	0	0	1
Q7	0	0	0.06	0	0	0	0.22	0	0.71	0	8
Q8	0.18	0.24	0	0	0	0	0	0	0	0	1
Q9	0	0.99	0	0	0	0.08	0	0	0.2	0	1
Q10	0	0.3	0	0	0.58	0	0	0	0	0	4
Q11	0.58	0	0	0.88	0.09	0	0	0	0	0	3
Q12	0	0	0	0	0	0	0.77	0	0	0	6
Q13	0	0	0	0.71	0	0	0	0.29	0	0	3
Q14	0.25	0.57	0.96	0	0	0	0	0	0	0.4	2

In the given table, each row represents a query and each column represents a topic. The topic weight for each topic in a given query is given in the cell value. The dominant topic is identified as the topic with highest value.

A graphical representation of the topic weights for each query is shown in figure 4.4.

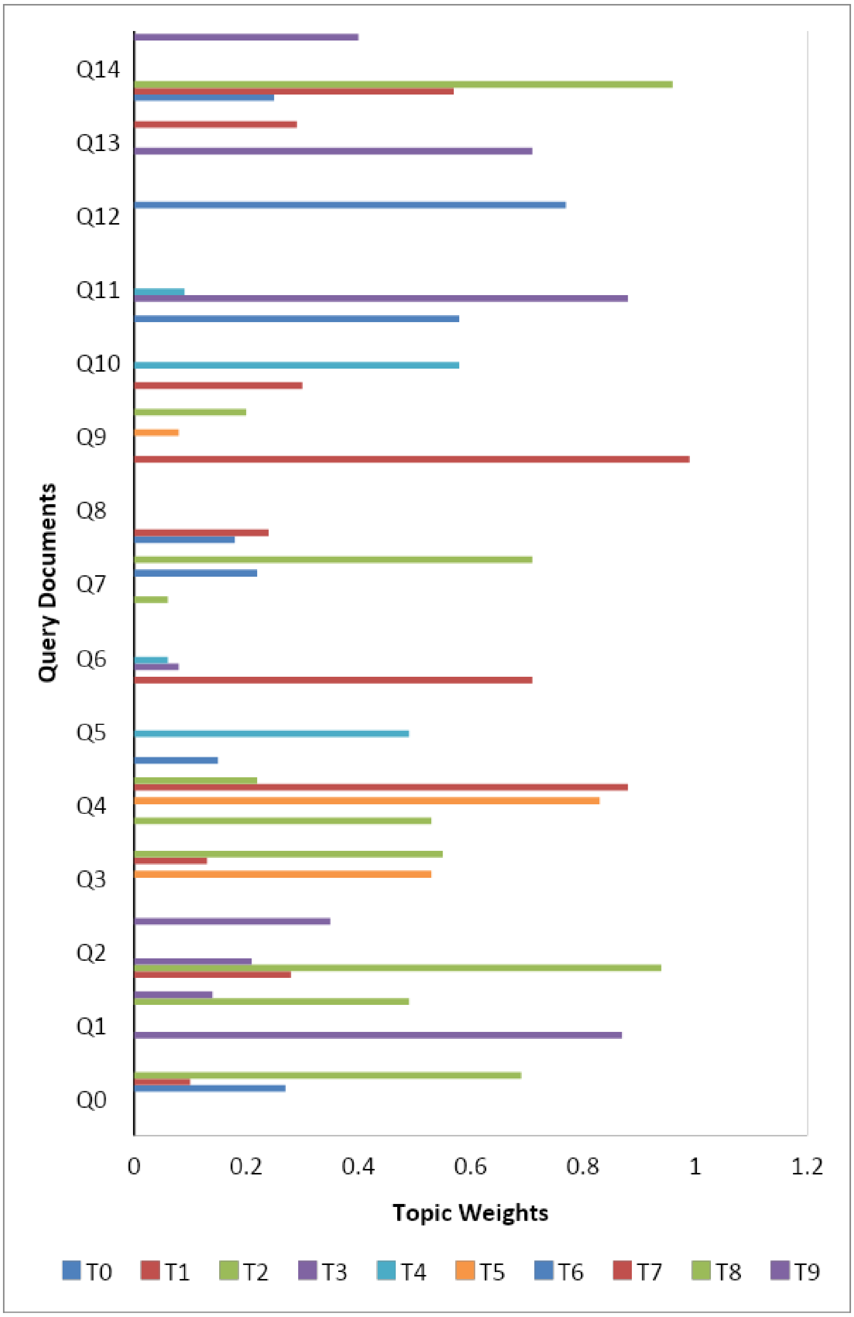


Figure 4.4: Topic Weights

A graphical representation of the dominant topics and numbers of queries belonging to each topic is given in the figure 4.5.

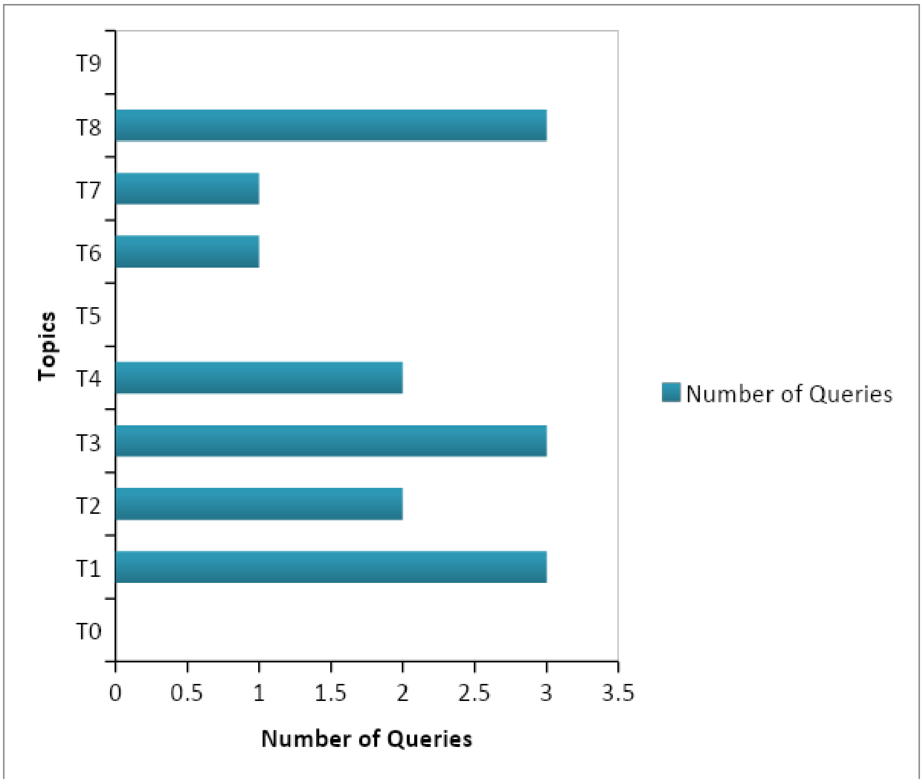


Figure 4.5: Number of queries belonging to each topic

Initially the number of topics is set to 10. But from the experimental results it is evident that not all topics are dominant by examining the number of queries containing the topics. Topics 0, 5 and 9 are not significant as no queries considered here contain the topics 0, 5 and 9. The five most dominant topics identified are 1, 2, 3, 4 and 8.

4.3.1. Experimental Results and Analysis of Topic Modelling

The most straightforward method for obtaining better results in topic modelling using GloVe is by increasing the training iterations when estimating the word vectors from the corpus. This is done on the GloVe model. Instead of training the model with the same parameters on different categories of documents on the Document Set, the model is trained with the same parameters but by changing the number of iterations. Initially, the value of MAX_ITER (parameter for maximum number of iterations) was set as 10. And after each complete execution, the model is trained with an increasing number of iterations like 10, 20, 30, 50. It should be considered that all methods benefit from increasing the number of training iterations. However the proportion of the increased similarity is minimal, which shows that the model already reached a local optimum and that extra 50 iterations were beyond the point where the phenomenon of diminishing returns started affecting the GloVe model. Hence the value of maximum iteration becomes MAX_ITER is 50.

The same is in the case of Topic modeling on LDA. Training iterations are increased while estimating the latent features from the corpus to obtain better results using LDA algorithm for topic modeling. This is performed on the LDA model. Instead of training the model with the same parameters on each different category of documents on the Document Set, we trained the model with the same parameters but with PASSES=50. Initially, the value of PASSES was 10. And after each complete execution, the model is trained with an increasing number of iterations, likely 10,20,30,50. It should be considered that all methods benefit from increasing the number of training iterations. However, the proportion of the increased similarity is minimal, which shows that the model already reached a local optimum and that extra 50 iterations were beyond the point where the phenomenon of diminishing returns started affecting the LDA model.

4.4 Comparison and Analysis of Similarity Measures

The cosine and Jaccard similarities are measured for the set of documents against the set of query.

Cosine Similarity measure computes the cosine of the angle between the query and document vector, as shown in Equation 4.7.

$$Cos(Q, D_i) = \frac{\sum_{j=1}^{t} W_{qj} d_{ij}}{\sum_{j=1}^{t} \left(d_{ij}\right)^2 \sum_{j=1}^{t} \left(W_{qj}\right)^2} \qquad (4.7)$$

The numerator represents the dot product of the query vector q and document vector d, while the denominator is the product of their Euclidean lengths.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle.

The Cosine similarity between two vectors (or two documents on the Vector Space) is a measure that computes the cosine of the angle between them. This metric is a measurement of orientation and not magnitude. It can be seen as a comparison between documents on a normalized space because the magnitude of the count of each word on each document (*tf-idf*) and the angle between the documents are considered for this study. Hence the cosine similarity equation is used to solve the equation of the dot product for the cosine of the angle.

The cosine similarity measurement for the query and documents is represented in figure 4.6.

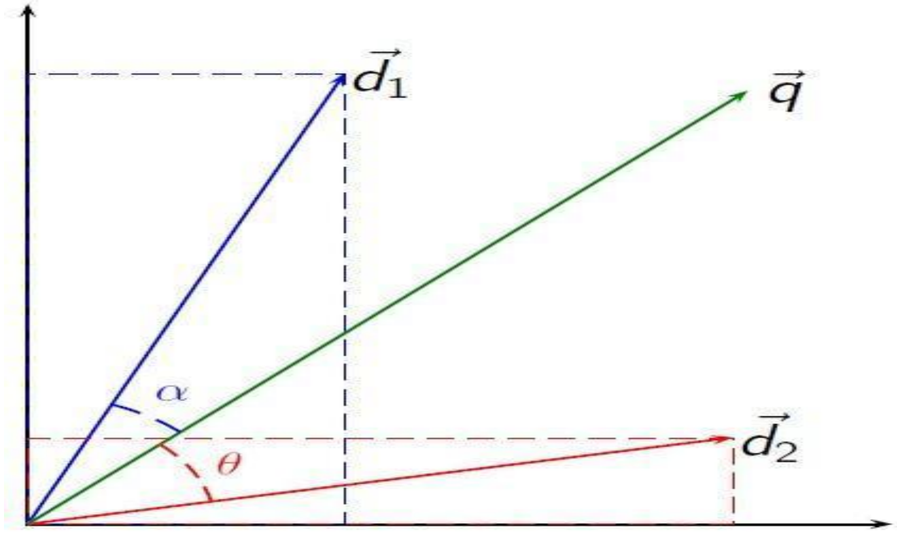


Figure 4.6: Cosine of the angle between query and documents

From figure 4.6, finding the deviation of angles between each document vector and the original query vector where the query is represented as the same kind of vector as the documents makes it easier to compute the cosine of the angle between the vectors, instead of the angle itself.

Jaccard Coefficient is the size of the intersection divided by the size of the union of the document and query vectors as given in Equation 4.8.

$$Jaccard(Q, D_i) = \frac{\sum_{j=1}^{t} W_{qj} d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 + \sum_{j=1}^{t} (W_{qj})^2 - \sum_{j=1}^{t} W_{qj} d_{ij}} \qquad (4.8)$$

The similarity of the documents and queries are measured using the cosine similarity and Jaccard similarity. For the ease of verifying the similarity, the first 5 queries from each category of the query set are compared with all documents in the particular document set.

As there are many documents in each category, the Cosine and Jaccard values of the three randomly selected documents for the first 2 queries in each set of documents is given in the tables 4.1, 4.2, 4.3, 4.4, and 4.5 respectively.

Table 4.2 Similarity of query related to Administration category with Documents

Query file name	Document file name	Cosine Similarity	Jaccard Similarity
Adq1.txt1	Ad9.txt	0.8901	0.7381
Adq1.txt1	Ad3.txt	0.9012	0.8394
Adq1.txt1	Ad56.txt	0.1585	0.1008
Adq1.txt2	Ad9.txt	0.3263	0.1976
Adq1.txt2	Ad17.txt	0.7102	0.6423
Adq1.txt2	Ad23.txt	0.0053	0.0

Table 4.3 Similarity of query related to Academics category with Documents

Query file name	Document file name	Cosine Similarity	Jaccard Similarity
Acq1.txt1	Ac67.txt	0.2247	0.001
Acq1.txt1	Ac3.txt	0.7488	0.6254
Acq1.txt1	Ac108.txt	0.2842	0.108
Acq1.txt2	Ac395.txt	0.5662	0.4976
Acq1.txt2	Ac179.txt	0.1574	0.003
Acq1.txt2	Ac23.txt	0.8247	0.7756

Table 4.4 Similarity of query related to Examination category with Documents

Query file name	Document file name	Cosine Similarity	Jaccard Similarity
Exq1.txt1	Ex79.txt	0.6472	0.4582
Exq1.txt1	Ex38.txt	0.4525	0.2984
Exq1.txt1	Ex564.txt	0.7824	0.6977
Exq1.txt2	Ex93.txt	0.3620	0.2356
Exq1.txt2	Ex177.txt	0.5940	0.5121
Exq1.txt2	Ex3.txt	0.4068	0.396

Table 4.5 Similarity of query related to Finance with Documents

Query file name	Document file name	Cosine Similarity	Jaccard Similarity
Fnq1.txt1	Fn9.txt	0.8247	0.7752
Fnq1.txt1	Fn3.txt	0.4255	0.3652
Fnq1.txt1	Fn56.txt	0.7824	0.7123
Fnq1.txt2	Fn9.txt	0.0620	0.042
Fnq1.txt2	Fn17.txt	0.5940	0.462
Fnq1.txt2	Fn23.txt	0.7168	0.6352

Table 4.6 Similarity of query related to Planning & Development with Documents

Query file name	Document file name	Cosine Similarity	Jaccard Similarity
Pdq1.txt1	Pd9.txt	0.8972	0.812
Pdq1.txt1	Pd3.txt	0.1225	0.1
Pdq1.txt1	Pd56.txt	0.4258	0.3879
Pdq1.txt2	Pd9.txt	0.5657	0.4852
Pdq1.txt2	Pd17.txt	0.0009	0.0
Pdq1.txt2	Pd23.txt	0.7426	0.6943

Figure 4.7 shows the average measure of similarity based on Cosine and Jaccard similarities.

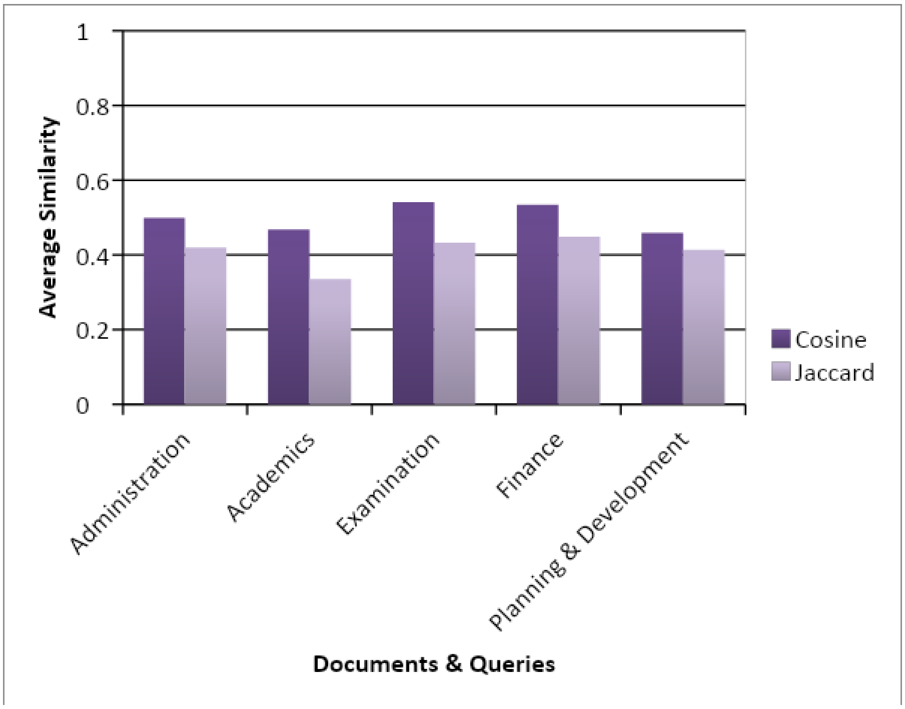


Figure 4.7: Graphical view of the average measures of similarity

The cosine and Jaccard measures for similarity vary between 0 and 1, hence a threshold or cut off value is to be determined for further analysis. Since these similarity measures vary depending on the domain structure, manual verification for a sample set is required. For this analysis, 25 queries are randomly selected from each category and the similarity measures are computed with 100 documents from the five categories identified for the study. From this the top 100 documents on the basis of cosine similarity are selected *ie.,* a set of 5 x 100=500 documents. The same is repeated for Jaccard similarity measure. Each document is labeled with a 1 or 0 as relevant or not respectively. The accuracy obtained is 91.6% for Cosine and 88.4% for Jaccard similarity measures.

Similarity measures such as Cosine, Jaccard, Euclidean and Okapi are extensively used in VSM. Hybrid models by using Fuzzy logic are also presented by Y. Gupta *et.al.,* to improve the performance of IR systems [152][153]. Pathak *et.al.,* suggested some hybrid models based on Genetic Algorithm (GA) [154].

The following section describes the proposed similarity measure which applies Fuzzy logic to deal with the vagueness present in the query.

4.5 Applying Fuzzy Logic for Dealing with Vagueness to form Hybrid Similarity Measures

A fuzzy logic system is a formal framework well suited for modelling vagueness and uncertainty [155]. In this work, a hybrid similarity measure is proposed which incorporates the features of standard/conventional similarity measures with the fuzzy logic.

The fuzzy weights are determined on the basis of the individual similarity scores of different similarity measures.

This approach ensures that a higher value of weight is assigned in a fuzzified manner to the similarity measure having a high similarity score [114]. Diverse similarity measures are possible by a suitable weighted combination of scores produced by different similarity measures which improves the results obtained by individual measures.

Similarity measures such as Cosine and Jaccard are widely used in VSM. The simplest method to design a hybrid similarity measure is by finding a linear combination of these similarity measures which overcome the limitations of different similarity measures. But, the weights for different similarity measures are to be determined to combine these measures. Since the dataset considered for this study consists of natural language queries there should be a mechanism to deal with the vagueness and uncertainty present in them.

So the development of hybrid similarity measures is focused on finding appropriate weights to be used for Cosine and Jaccard similarity measures. Fuzzy logic is employed in the proposed approach to deal with the vagueness present in natural language queries. It can be named as Fuzzy-Jaccard-Cosine Similarity Measure (FJCSM). The mathematical formulation of FJCSM and the reasoning based on fuzzy logic is explained in the following sections.

4.5.1 Proposed Fuzzy – Jaccard - Cosine Similarity Measure (FJCSM)

The mathematical formulation for a Fuzzy-Jaccard-Cosine Similarity measure (FJCSM) is expressed as in Equation 4.9.

$$FJCSM = \sum_{i=1}^{n} \left(wt_i X \; sm_i(D, Q) \right) \qquad (4.9)$$

where sm_i *(D,Q)* represents the similarity measure that is used to compute the matching score of document *D* for a given query *Q;* wt_i denotes the weight of the *i*th similarity measure and *n* is the total number of similarity measures considered. The appropriate values of weights are determined using a fuzzy logic system to maximize retrieval efficiency.

Generally, three types of fuzzy inference methods are proposed in literature: Mamdani fuzzy inference, Sugeno fuzzy inference, and Tsukamoto fuzzy inference. All of these three methods can be divided into two processes. The first process is fuzzifying the crisp values of input variables into membership values according to appropriate fuzzy sets, and these three methods are exactly the same in this process. While the differences occur in the second process then the results of all rules are integrated into a single precise value for output. In

Mamdani inference, the consequent of If-Then rule is defined by a fuzzy set. The output fuzzy set of each rule will be reshaped by a matching number, and defuzzification is required after aggregating all of these reshaped fuzzy sets. But in Sugeno inference, the consequent of If-Then rule is explained by a polynomial with respect to input variables, thus the output of each rule is a single number. Then a weighting mechanism is implemented to work out the final crisp output. Although Sugeno inference avoids the complex defuzzification, the work of determining the parameters of polynomials is inefficient and less straightforward than defining the output fuzzy sets for Mamdani inference. A Mamdani-type Fuzzy Inference System is used here as it is the most popular one [156].

Mamdani fuzzy inference was first presented as a technique to create a control system by synthesizing a set of linguistic control rules obtained from experienced human operators [157]. In a Mamdani system, the output of each rule is a fuzzy set. Since Mamdani systems have more natural and easier to understand rule bases, they are compatible with expert system applications where the rules are created from human expert knowledge, such as medical diagnostics. The block diagram of a fuzzy logic system is given in figure 4.8 that incorporates three major blocks.

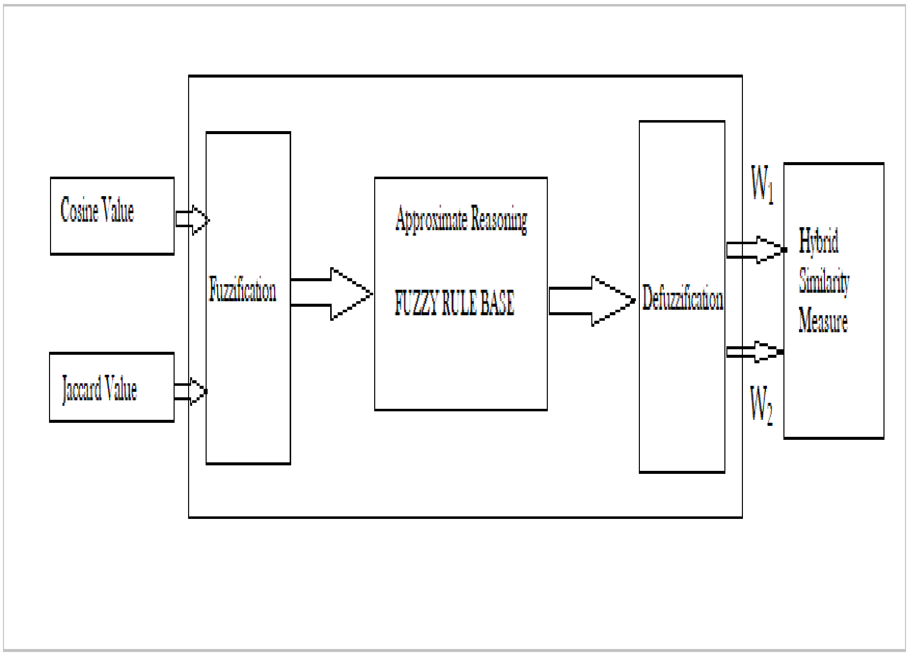


Figure 4.8: Fuzzy logic system

The first is the input block, the second is the main fuzzy logic system block and the third is the output block. The working of a fuzzy logic controller block, which contains three processes – fuzzification, approximate reasoning and defuzzification, is illustrated.

4.5.2 Fuzzification

In the proposed approach, two input fuzzy variables such as the scores or values of Cosine and Jaccard for each document are used for new measures. W_1 and W_2 are the output variables, which represent the weights of Cosine and Jaccard similarity measures, respectively. The range of all input and output variables is represented by three linguistic terms as low, medium and high.

A membership function (MF) is a curve that defines the feature of a fuzzy set by assigning to each element the corresponding membership value, or degree of membership. It maps each point in the input space to a membership value in a closed unit interval [0, 1].

A triangular membership function is used to map input space to a degree of membership of a fuzzy set. Figure 4.9 shows the triangular membership function.

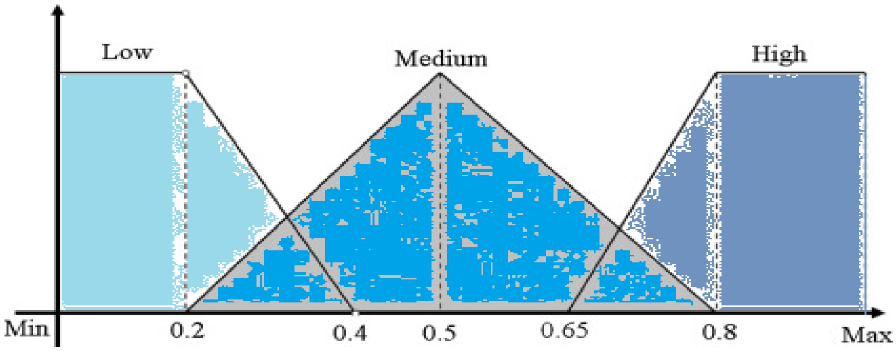


Figure 4.9: Input and Output Membership function

4.5.3 Approximate Reasoning Based on Fuzzy Rule base

An approximate reasoning is established for inference in order to deal with uncertainty and vagueness. The Mamdani-type fuzzy rule toolbox is used to formulate the conditional statements that comprise fuzzy logic. Common knowledge of IR signifies that a higher value of weight should be assigned to the associated similarity measure that has a higher similarity value/score as compared with other similarity measures that have lower similarity values/scores. Therefore, the fuzzy rules are framed on the basis of domain knowledge. A total of 9 Mamdani-type fuzzy rules are framed as conditional statements, and are listed in Table 4. 7.

Table 4.7: Fuzzy Rule Base

Sl. No	Fuzzy Rules
1	If (cosine is Low) and (jaccard is Low) then (W1 is Low) (W2 is Low)
2	If (cosine is Low) and (jaccard is Medium) then (W1 is Low) (W2 is Medium)
3	If (cosine is Low) and (jaccard is High) then (W1 is Low) (W2 is High)
4	If (cosine is Medium) and (jaccard is Low) then (W1 is Medium) (W2 is Low)
5	If (cosine is Medium) and (jaccard is Medium) then (W1 is Medium) (W2 is Medium)
6	If (cosine is Medium) and (jaccard is High) then (W1 is Medium) (W2 is High)
7	If (cosine is High) and (jaccard is Low) then (W1 is High) (W2 is Low)
8	If (cosine is High) and (jaccard is Medium) then (W1 is High) (W2 is Medium)
9	If (cosine is High) and (jaccard is High) then (W1 is High) (W2 is High)

4.5.4 Defuzzification

In the proposed system, the centroid method (centre of sums) is used for defuzzification. The defuzzified output value represents the weights for values of Cosine and Jaccard as in equation 4.10.

$$Y = \int_{y} \sum_{i=1}^{n} y.\mu_{Bi}(y)dy \Bigg/ \int_{y} \sum_{i=1}^{n} \mu_{Bi}(y)dy \qquad (4.10)$$

4.6 Experimental Analysis and Results

The experiments have been performed on the dataset developed as part of this study consisting of 2800 document files and 7500 queries in English. The 25 queries are selected randomly from each dataset. The analysis of results obtained is presented in the form of precision-query curves, recall-query curves, average precision rate and average recall rate. The analysis is done for the top 10, top 25 and top 50 documents. The results are compared with Cosine, Jaccard, and the proposed system.

The performance of the proposed fuzzy-hybridized approach is evaluated on the basis of precision and recall in this study. The precision and the recall are defined as in equation (4.11) and (4.12).

$$\text{Precision} = |R_d| / |A| \qquad (4.11)$$

$$\text{Recall} = |R_d| / |R| \qquad (4.12)$$

Where $|R_d|$ is the set of relevant documents retrieved, $|A|$ is the set of total documents retrieved and $|R|$ is the set of all relevant documents.

Figures 4.10 shows the Precision vs Query plot obtained for precision of top 10 retrieved documents for 25 different queries.

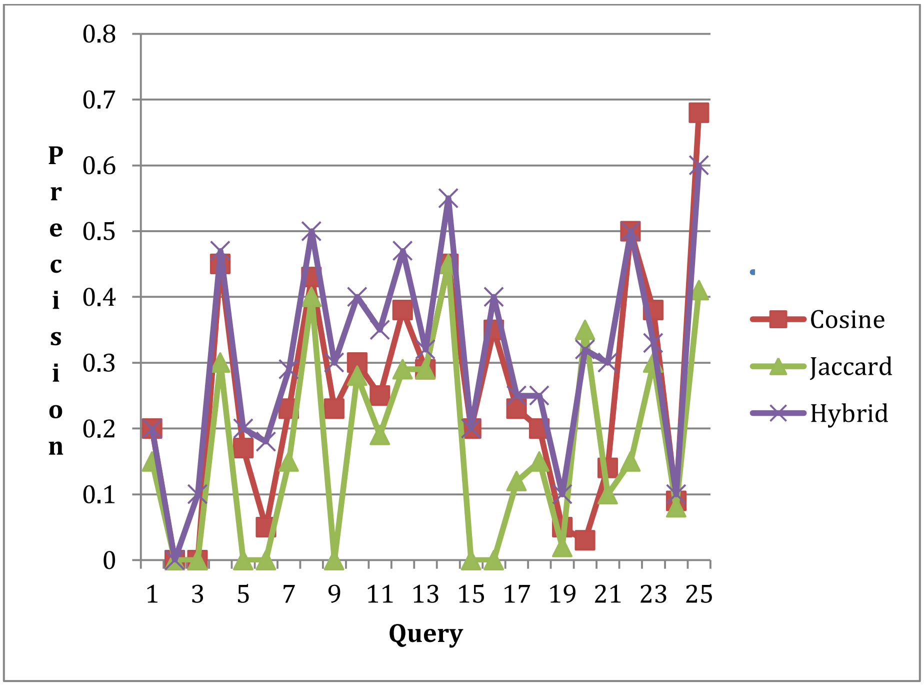


Figure 4.10: Precision of top 10 retrieved documents for 25 different queries

Figure 4.11 shows the Recall vs Query plot obtained for recall of top 10 retrieved documents for 25 different queries

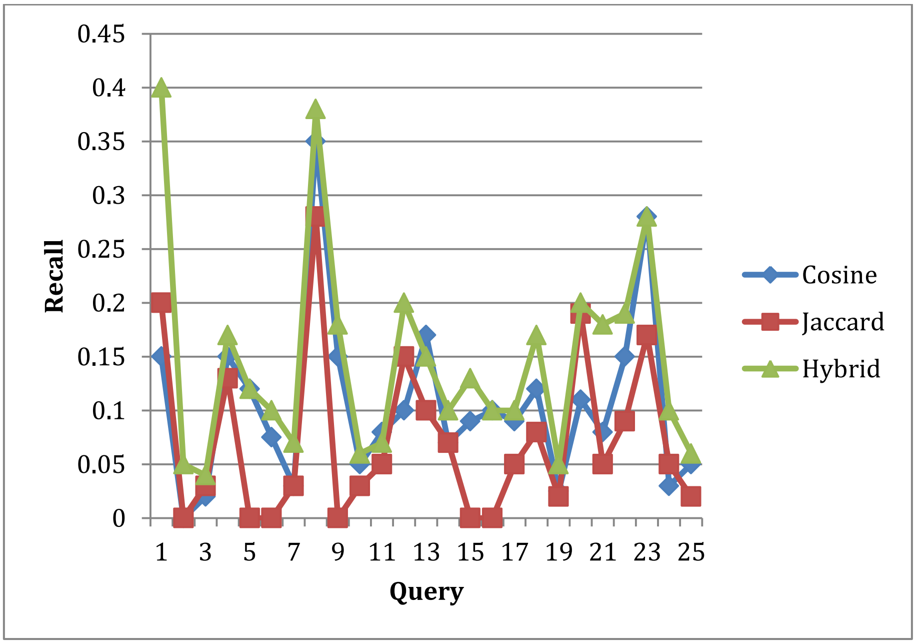


Figure 4.11: Recall of top 10 retrieved documents for 25 different queries

From Figure 4.10, it is evident that the hybrid method gives better precision values for 19 queries and lags for only three queries out of 25 as compared with Cosine, and Jaccard. Figure 4.11 shows that better recall values for 20 queries and lags for only 4 queries out of 25 as compared with Cosine, and Jaccard

Table 4.8 shows a comparison of the average precision and the average recall obtained for the proposed hybrid similarity measure with Cosine and Jaccard similarity measure for the SQUSE (Short Queries for University Services in English) with corresponding documents in the UDDSE (University Document Data Set in English). The analysis is done for the top 10 retrieved documents, the top 25 retrieved documents and the top 50 retrieved documents.

Table 4.8 Average precision and recall for retrieved documents

Method	Top 10 retrieved documents		Top 25 retrieved documents		Top 50 retrieved documents	
	Average precision	Averag e recall	Average precision	Average recall	Average precision	Average recall
Jaccard	0.1346	0.0287	0.1099	0.0718	0.0918	0.1178
Cosine	0.1977	0.0638	0.1137	0.0904	0.1036	0.1407
FJCSM	0.2491	0.0902	0.1741	0.1268	0.1398	0.1862

It is evident that the proposed method increases average precision and average recall of the IR system for dealing with document retrieval.

4.7 Conclusion

In this chapter efforts are made to measure the similarity of document and query for improving the results of information retrieval. When a natural language query is fired for retrieving information, the best similar document is to be retrieved. Since machines cannot work with the text as it is, texts are represented as numbers such as a set of independent units like unigrams, bigrams or multi-grams which acts as a building block of the feature space. Usually, the text is represented by the assigned values such as binary, Term Frequency (*TF*) or Term Frequency-Inverse Document Frequency (*TF-IDF*). In this study both the documents and queries are represented as vectors and their similarity is measured to find the best match for a given query. The Cosine and Jaccard coefficients are computed for the purpose. Then to improve the similarity measure, a hybridized model of fuzzy logic with Cosine and Jaccard is also proposed. The average precision and recall values are computed for the similarities obtained for the queries based on SQUSE (Short Query for University Services in English) dataset with the corresponding document data set in English UDDSE (University Document Data Set in English). The values obtained for average precision and recall using Jaccard, Cosine and the proposed FJCSM measures are compared. The proposed method shows an improvement in precision by more than 25% and in recall by more than 40% which is comparable with the existing methods for measuring similarities. Hence it is evident that the results are promising and can be effectively used for finding the best matched document against natural language query.

Chapter 5

An Ontology-Based Semantic Web Model for the University Domain

5.1 Introduction

Over the years, the volume of information available through the WWW has been increasing continuously and never has so much information been so readily available and shared among so many people. The role of searching information has therefore changed radically from systems designed for special purposes with a well-defined target group to general systems for almost everyone. But the unstructured nature and massive volume of information accessible over a network have made it increasingly difficult for users to filter out and find relevant information. Numerous Information Retrieval (IR) techniques have been developed for this purpose [158].

The commonly used IR techniques are related to keyword-based searching. These techniques use keyword lists to describe the content of information, but one problem with such lists is that they do not consider any semantic relationships between keywords, or they do not take into account the meaning of words and phrases. It is often demanding for ordinary users to use information retrieval systems based on these commonly used keyword-based techniques. Users usually express their information needs in natural language rather than system-specific query languages. These information requirements need to be converted into a query by the system that declines the performance of the system.

The World Wide Web, generally based on HTML, which cannot be fully utilized by the information retrieval techniques, and thus processing of information on the web is usually restricted to manual keyword searches, which results in extraneous information retrieval. To overcome this limitation, Tim Berners Lee proposed a new web architecture known as a semantic web, which is intelligent [15]. The central concept used in the design of semantic web is ontology. It is used for the exchange of information, use, and reuse of knowledge and to capture knowledge about any domain of interest to integrate the machine-understandable data on the current human-readable web. Web Ontology Language (OWL) is a semantic markup language for sharing ontologies on the web. There are different methods and tools for ontology development. This chapter describes proposed methods used to develop an ontology for University domain.

From the literature, it is found that a tool that can provide a friendly and easy interface to users for handling complex natural language queries is very much required. The capabilities of such tools to perform the retrieval of domain-specific information from the ontology are also investigated by many researchers [18][20][22]. In semantic search, emphasis is given to the frequency of words, syntactic structure of the natural language and other linguistic elements. This research work proposes an information retrieval system for the University domain. The proposed method offers advanced querying in which the search results are automatically aggregated and extracted directly in a reliable user interface by decreasing manual efforts. The most important objective of this research work that aims to design and develop a semantic web-based system that incorporates ontology towards domain-specific information retrieval is explained in detail.

In the first phase, a framework for the semantic web based system is designed. The second phase consists of building the prototype for the proposed framework with the help of Protégé Tool [38]. Dealing with the conversion of natural language questions (query) to SPARQL (SPARQL Protocol And Query Language) is done in the third phase. Then in the fourth phase, the converted SPARQL queries are fired into the ontology to get the results. Finally, the evaluation of the prototype has been carried out to ensure efficiency and usability.

The rest of this chapter is organized as follows. The significance of semantic search is noted in section 5.2. Section 5.3 describes the development of ontology for the University domain. Architecture of the proposed system for the University ontology is given in section 5.4. An overview of Protégé, the ontology development tool used for this study and the method used to retrieve information from the developed ontology for University domain is described in section 5.5 with its subsections. The crucial challenge faced in this study is manually identifying and conceptually mapping the general organizational hierarchies and functionalities generally found in a University system to corresponding ontology representation. To get over this tedious task which directly affects the retrieval of information, a semi-automatic process is suggested which will automatically generate the concepts of the ontology based on the document and query processing and is described in section 5.5.3. An analysis of the experimental result is carried out in section 5.6 and finally, section 5.7 draws the concluding remarks.

5.2 Significance of Semantic search

To search for a document in a traditional keyword based information retrieval system, users are provided with too many results among which most results are irrelevant. To overcome the limitation of keyword based systems, technique of conceptual search is implemented. Conceptual search implies search by meaning instead of matching keywords. For example, meaning of the word “Key” - taken either as “a small piece of shaped metal with incisions cut to fit the wards of a particular lock” or as “each of several buttons on a panel for operating a computer, typewriter, or telephone” or “of crucial importance” [159]. The keyword based system retrieves data for all these meanings. On the other hand, the ontological concept retrieves concepts from the domain concerned. This conceptual search technique is implemented by using the concept of ontology. In the field of Artificial Intelligence, ontology is defined as “the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary” [160]. The following section describes the process adopted for developing ontology for the University domain.

5.3 Development of Ontology for University Domain

Ontology has a good conceptual structure representation which can be combined with knowledge representation. This model makes use of annotation and indexing. The ontology model depends on the semantic index terms, whereas the Vector Space Model described in Chapter 4 depends on the keyword index. The semantics of the concepts are used to build a concept term representation.

The ontology similarity measure improves the concept of relevance score. Ontology provides a vocabulary comprising unambiguous definitions for terms that can fundamentally serve as formal support for communication among software agents [161]. A software agent is any computer software that acts for a user. To compare conceptual information across two knowledge bases on the web, a program must have a way to discover common meanings. In this case, the knowledge bases may be document and query. Ontology is a solution to this problem. Ontology formally describes a list of terms that represent essential concepts, such as classes of objects and the relationships between them to serve an area of knowledge. Ontologies provide a formal semantics that can be employed to process and integrate information on the web. Gruber *et.al.*, describe ontology as an explicit specification of conceptualization [162]. In their work, they define the semantics for different domains for interactions on the web and help in creating a knowledge base that will enable people to work on a particular domain. World Wide Web Consortium (W3C) recommends Web Ontology Language (OWL) for developing ontologies and is widely used to construct domain ontology. To perform useful automatic reasoning tasks on web data, it requires going beyond the underlying semantics of XML Schema and RDF Schema, and there is a need for a more expressive and reasoning language that enhances the RDF with more vocabulary [163].

Other important factors playing a significant role in the realization of efficient and intelligent retrieval of information on the web are XMLS, RDFS, URI, SPARQL, etc. XMLS (XML Schema) extends the capabilities of XML, where XML (Extensible Mark-up Language) is used for data exchange and to add meaning to data. RDFS (RDF schema) is to represent the web resource where RDF (Resource Description Framework) is for representing the knowledge resources on the web and uses the web identifier URI (Uniform Resource Identifier) to identify the resources. SPARQL (Standard Protocol for RDF Query language) is used to extract information from RDF graphs for machine understandable representation. In the context of the Semantic Web, social networks like twitter assist people in finding common relationships and discussion forums for exchange of information and are playing a crucial role in information credibility and trustworthiness [164].

The ontology-based information retrieval model for the University domain supports semantic search from the document repository. The problem here is to enhance the relevance in search and ranking of documents by considering the semantic information apart from mere keywords present in the user query. The search query may not be constant, since it is framed by a common man in natural language. Hence semantic techniques are applied to store the data and fetch the results based on the query.

Machine understandable information helps to have accurately detailed searching with less time and effort. This paves the way for knowledge management and agent-based processing [165]. In the context of semantic web, searching means searching by contextual meaning [166]. The following section describes the features of the generic University domain.

5.3.1 Description of University Domain

The domain ontology shows concepts, the association between concepts in a specific subject area rather than postulating the generic concepts. The known information about a particular subject is modelled through the ontology just as it is in a textbook of that subject. By modelling the University system as the domain of interest, an exclusive search service for the University-related information is possible. It helps people belonging to different domains or fields in retrieving University related information in an easier way. The general structure of State Universities in Kerala is considered for the development of domain ontology. The main branches of the University such as Administration, Academics, Examination, Finance, and Planning & Development with all the sub branches are represented with concepts in ontology. Figure 5.1 shows a part of the University domain describing the academic subdomain.

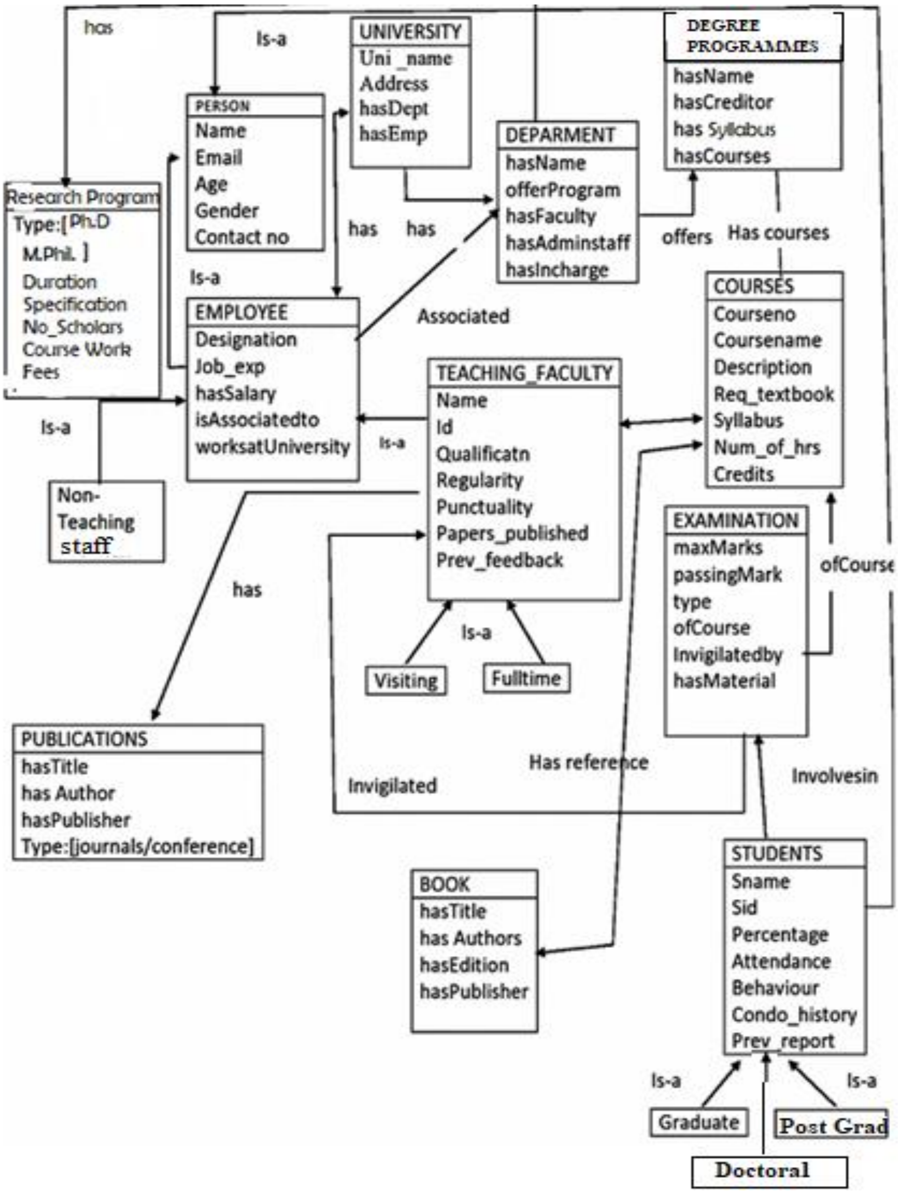


Figure 5.1 : A general layout of academic sub domain of a University system

The ontology based representation is an effective and efficient IR system for providing necessary information as quickly as possible with the appropriate reasoning for decision making in any issues related to the University. Various stakeholders including students, faculty members, non-teaching staff, government authorities, parents or common people may approach the University for information access through various means. The required information may be in the form of a document or just an answer depending on the query. Hence a domain ontology is proposed to provide the facility to search in the University domain without searching the entire web for the relevant information which enhances the efficiency of the retrieval system. The foundation for developing ontology is the conceptualization process, which starts with the preparation of the final list of keywords and descriptors.

In this work, five major concepts are identified to construct the ontology for the University domain as listed below.

- Academics
- Administration
- Examinations
- Finance
- Planning & Development

The University ontology is being developed for semantic web based information retrieval. It is accomplished by representing each component of the University as classes or subclasses and their properties in the ontology. The University orders issued by various branches are also considered as a source of information which describes the ontology.

The data structure for representing the concepts related to the domain with its relations and properties is predefined. The ontology is created using the concepts of ontology with Web Ontology Language (OWL) and is saved as .owl files. This information will be in XML form which is created with the Protégé tool. The main concepts identified for the development of University ontology is represented in figure 5.2. The root of all concepts is represented as University. Then the five main concepts identified for this study are represented as the child nodes of the University in the corresponding ontology.

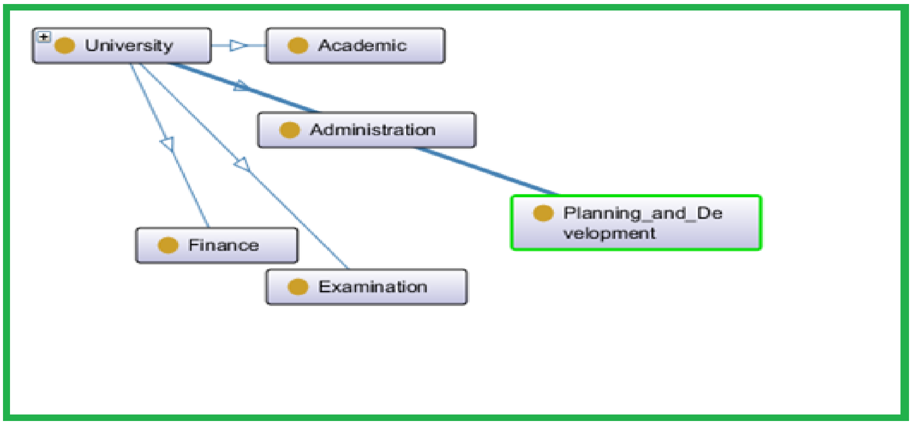


Figure 5.2: The main classes identified for developing the University ontology

The following section describes the proposed methods used for designing ontology, defining objects, properties and performing consistency checks in detail.

5.3.2 Ontology Design

Developing Ontology is a very challenging domain oriented process which has to be supported by an appropriate ontology development tool. Ontology is best presented in Resource Description Framework (RDF) data format which is a triplet form analogous to *subject-object-predicate* in a natural language. In this work, Protégé_4.2.8 tool is used to develop ontology for the University domain. Various topics from the University or the Government orders are used to create the proposed system. The ontology development process consists of various stages which are shown in figure 5.3.

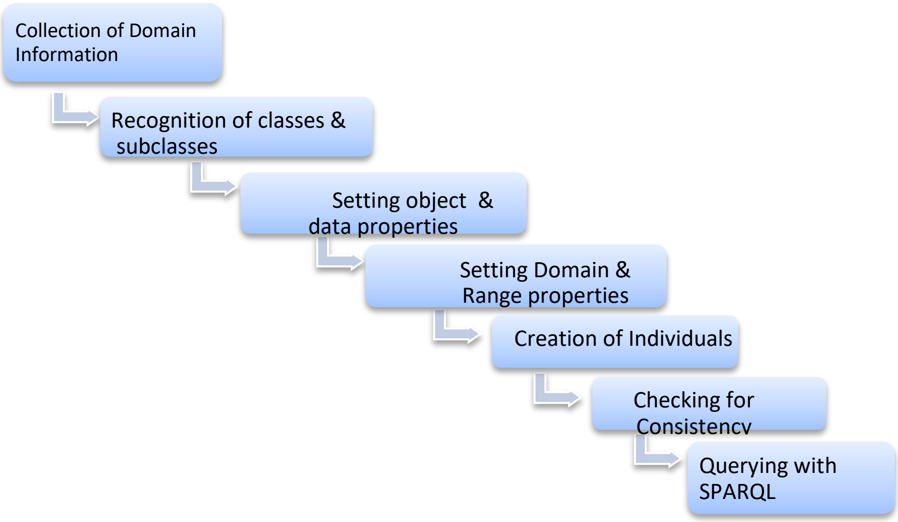

Collection of Domain Information
Recognition of classes & subclasses
Setting object & data properties
Setting Domain & Range properties
Creation of Individuals
Checking for Consistency
Querying with SPARQL

Figure 5.3: Stages of Ontology Development

First stage stands for collecting the domain information required for the ontology development. Next stage aims to recognize all the classes and subclasses for the ontology to be developed. Setting the properties between classes and subclasses is done in the third stage. There are two types of properties, such as object properties and data properties. Object properties are generally used to describe relationships between two individuals or instances of classes. Relationships between instances and data values are described using data properties. Each property has domain and range. Hence setting the domain and range of every property comes in the fourth stage.

Comments are added to give explanations to classes and properties defined for the domain. Fifth stage meant for creating instances of classes and to set their data and object properties to define relationships between the instances of various classes and subclasses. Consistency checking which performs the testing of the developed ontology is carried out in the next stage. To check the consistency on the ontology developed, either the Inbuilt Reasoner or the additional plug-ins of the Protégé can be used. Finally, in the seventh stage, the ontology is saved in RDF/OWL format. To finish, the ontology has to be exported in RDF or OWL data format to the required interface to execute the queries.

5.3.3 Defining Classes and Subclasses

In the University scenario, the core concept is considered as five main departments *viz* Academics, Administration, Examination, Finance and Planning & Development as discussed earlier. Even though there are various other sub branches in the University, they can be associated with any of these categories one way or another. While building the prototype for the proposed SIRSU (Semantic Information Retrieval System for University) system each of these classes are connected to several other classes and subclasses as shown in figure 5.4. The root node named '*University*' contains classes of the five major domains as already described in section 5.3.1. These classes have various members which act as their instances.

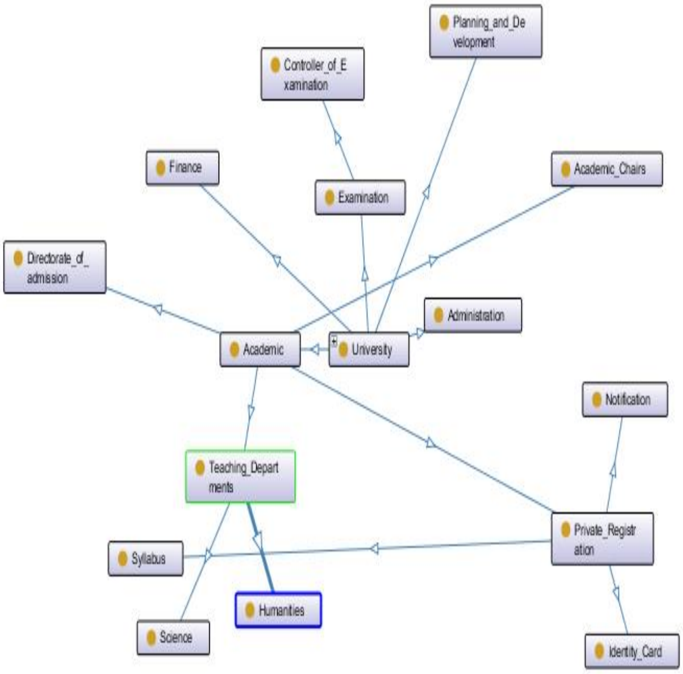
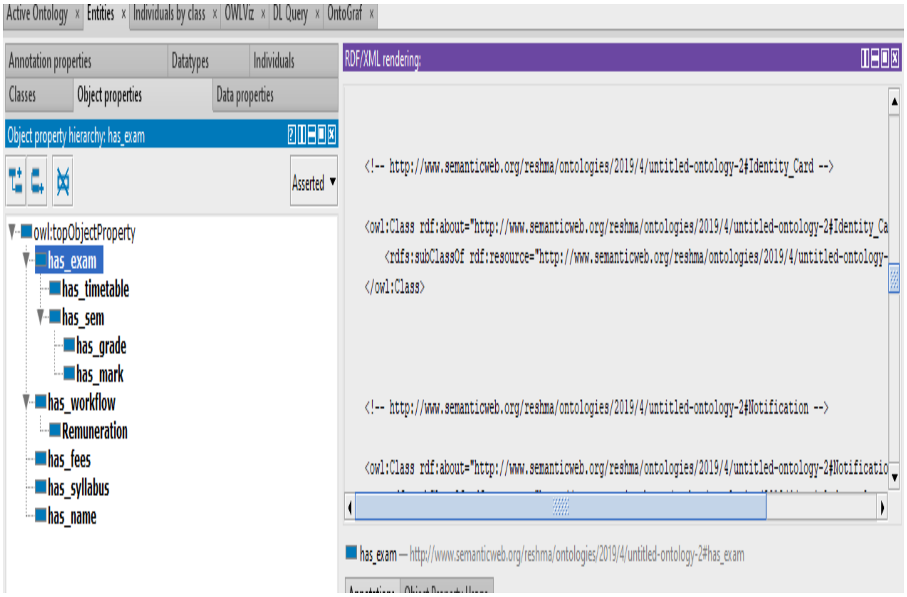


Figure 5.4: Classes and their sub classes defined in the proposed SIRSU

The following section describes the method used to identify object and data properties in this University domain.

5.3.4 Identification Object and Data Properties

Various classes and subclasses are defined in section 5.4.3. As the next stage the Properties of these classes also need to be specified. The relationship between two instances or individuals of classes is described with several object properties such as section, department, faculty, course_offered, course_duration, fees, timetable, etc. Every property has domain and range which needs to be set while developing ontology. Figure 5.5 shows the properties of different classes as defined with the help of the Protégé tool as part of this study.



- owl:topObjectProperty
  - has_exam
    - has_timetable
    - has_sem
      - has_grade
      - has_mark
  - has_workflow
    - Remuneration
  - has_fees
  - has_syllabus
  - has_name

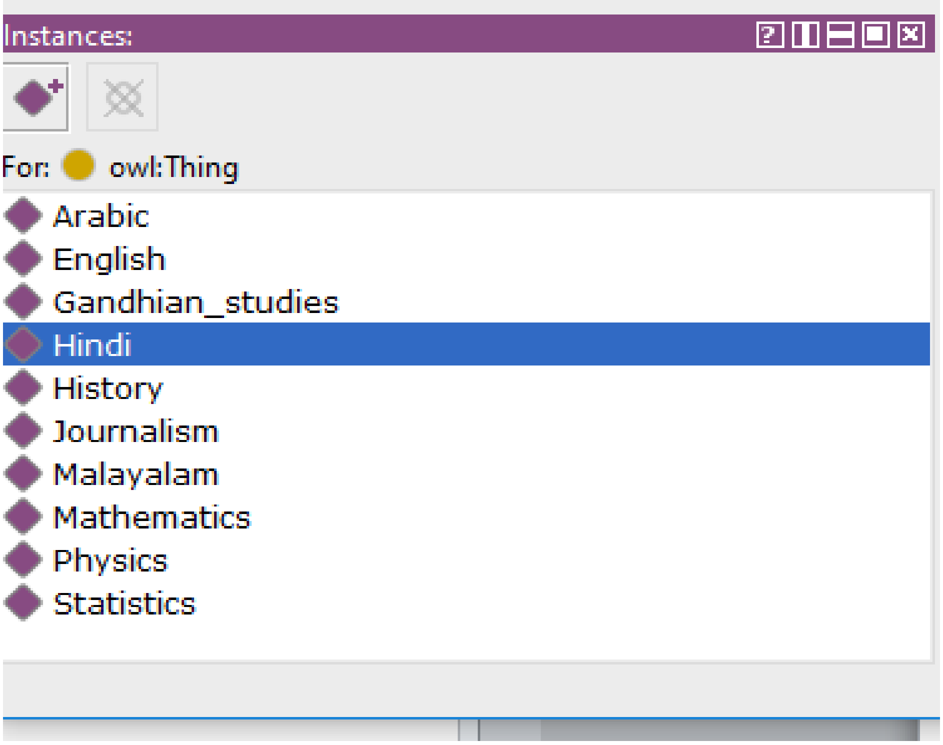
```
<!-- http://www.semanticweb.org/reshma/ontologies/2019/4/untitled-ontology-2#Identity_Card -->

<owl:Class rdf:about="http://www.semanticweb.org/reshma/ontologies/2019/4/untitled-ontology-2#Identity_Ca
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/reshma/ontologies/2019/4/untitled-ontology-
</owl:Class>



<!-- http://www.semanticweb.org/reshma/ontologies/2019/4/untitled-ontology-2#Notification -->

<owl:Class rdf:about="http://www.semanticweb.org/reshma/ontologies/2019/4/untitled-ontology-2#Notificatio
```

has_exam — http://www.semanticweb.org/reshma/ontologies/2019/4/untitled-ontology-2#has_exam

Figure 5.5: Properties of different classes defined using Protégé Tool

5.3.5 Creation of Individuals

Individuals act as an instance for classes and subclasses. Through *instances* or *individuals*, *relationships* are formed between all the entities of the respective ontology. For example, in the developed ontology, *Computer Science* is a value given to the data property named *Department* and another property called *course_offered* is given values *M.Sc, M.Phil*. Likewise all the individuals are set for all the classes and subclasses. Figure 5.6 shows certain *individuals* created in this University domain.



Instances:

For: owl:Thing

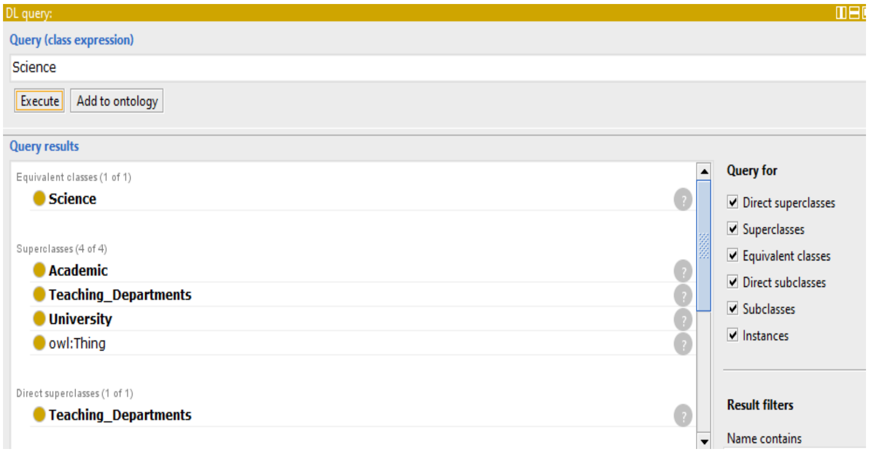
- Arabic
- English
- Gandhian_studies
- Hindi
- History
- Journalism
- Malayalam
- Mathematics
- Physics
- Statistics

Figure 5.6: Specifying individuals for the University domain

5.3.6 Consistency Check

Reasoner is an item in Protégé to check the consistency of the developed ontology. The *reasoner* for the consistency check needs to be started and when finished checking, it has to be stopped. The popular plugin available for consistency check in Protégé is *HermiT reasoner*. RDF validator helps to check the triples formed in the developed ontology for finding out the bugs.

To acquire relevant information about the concept of the University from the designed ontology, DL (Description Logic) query is used. The accuracy of the designed concepts and information in the developed ontology is checked with these queries. If the class or any property name is provided, the *reasoner* will display the related information about them. Thus it is possible to check whether the design of ontology is accurate in terms of mapping of super classes, subclasses and setting up relationships. For example, in order to retrieve a particular course under the science faculty of a University, it is required to provide the class name correctly (upper or lower case) as created in the ontology design. In this case, if the *class name* is given as *Science* and the result snippet will be obtained as shown in the figure 5.7.



Science

Equivalent classes (1 of 1)
- Science

Superclasses (4 of 4)
- Academic
- Teaching_Departments
- University
- owl:Thing

Direct superclasses (1 of 1)
- Teaching_Departments

☑ Direct superclasses
☑ Superclasses
☑ Equivalent classes
☑ Direct subclasses
☑ Subclasses
☑ Instances

Name contains

Figure 5.7: Results retrieved by DL query for the class name Science

5.3.7 Running SPARQL on Ontology

This section describes the process adopted to fetch the required information from Ontology using SPARQL. There are three approaches for running SPARQL queries on Ontology. The first approach is to run SPARQL queries through Protégé. In another approach SPARQL queries are executed through Apache's GUI based Jena Fuseki server. The third approach is to execute through Apache's ARQ which has a command line interface. Both ARQ and Jena Fuseki servers are SPARQL engines and are open source. These engines are a query engine for Jena that supports the SPARQL RDF Query language [167].

The following section describes the proposed architecture for the development of domain ontology for the University.

5.4 Architecture for the Development of Domain ontology for the University

As mentioned earlier the University ontology could be designed with various classes and subclasses required for the semantic representation of the concept of a University. The day to day business and the functions are guided and controlled by the different orders issued by the University or the Government authorities. The retrieval of information by processing the natural language query from various stakeholders is the major task carried out in this work.

The architecture proposed for the University Semantic Information Retrieval System based on ontology is shown in the figure 5.8. A GUI is developed using python to receive the query from the user. The input query is then refined using the extracted domain related keywords to be used as a SPARQL query. It can directly be used with the ontology designed for the University domain.

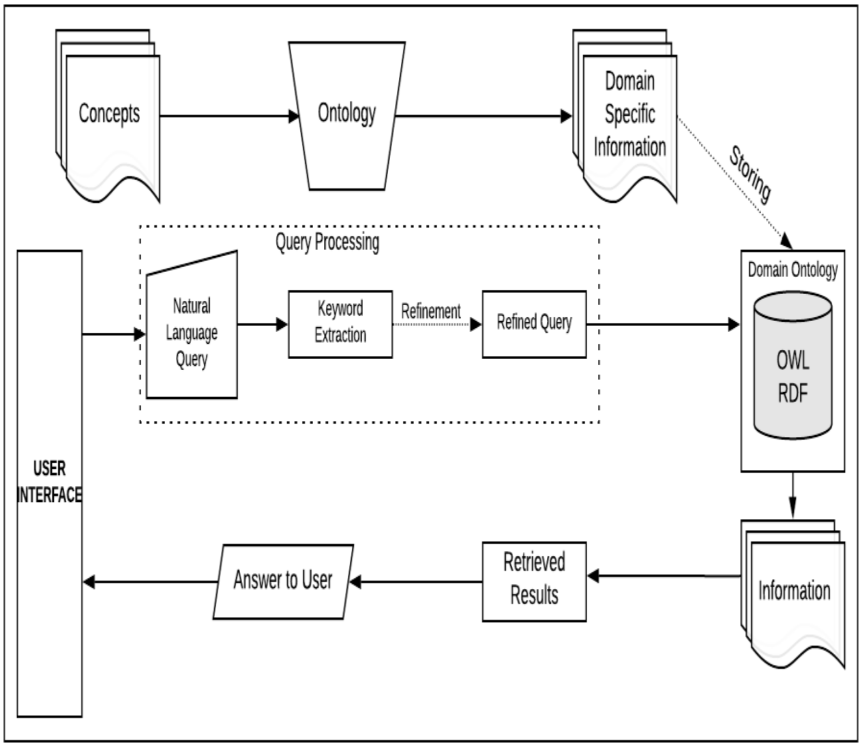


Figure 5.8 : Proposed architecture for the SIRSU

In the proposed SIRSU (Semantic Information Retrieval System for University) system architecture, the semantic information is extracted from the natural language query provided by the user. This semantic information is used to translate the query into a format which can be used to directly retrieve information from the given ontology. In the proposed system, the query input which is in natural language is converted into a SPARQL query which is a query language for RDF based databases. SPARQL query is then fired on to the RDF database and accesses the relevant information.

In the proposed architecture, there are mainly three phases. Initially a domain specific ontology has to be created. Next phase is the implementation of a user interface that accepts queries in natural language which need to be converted into SPARQL. This process is performed using the Python based QUEPY framework. The converted SPARQL queries are then fired to the ontology to fetch the results. Figure 5.9 shows these three phases.

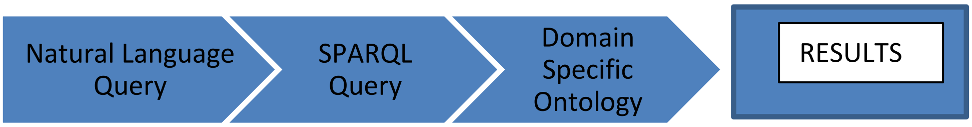


Figure 5.9: The three phases in the development of the proposed SIRSU system

After recognizing named entities from the natural language query, matching is performed in which the ontology *concepts* and *properties* are matched with the query terms to generate the triples (subject-predicate-object) generally referred to as (s-o-p). The SPARQL query is made by integrating these triples. The query generated in SPARQL is fired to the domain-specific ontology to get the results through the user interface. The experimental framework with various tools used for developing the University ontology and the experimental setup for retrieving information from the developed ontology is presented in the following section.

5.5 Experiment Framework

Retrieving information in a semantic manner requires the concepts to be represented in ontology. The commonly used tool used to design ontology is Protégé. An overview of the Protégé tool used for the system development is detailed in the following section.

5.5.1 Use of Protégé for the Development of University Ontology

Protégé is a free, open-source ontology editor and a knowledge management system. It was developed by the Biomedical informatics Research at Stanford University [38].

Protégé provides a graphic user interface to define OWL ontologies. OWL is the Web Ontology Language, which is used as the markup language for the semantic web. It also includes deductive classifiers used to validate those models for consistency and to infer new information based on the analysis of an ontology. Protégé is a framework for which various other projects suggest plugins. This application is written in Java and heavily uses Swing to create the user interface. It is also found that the Java based tool named Protégé is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development [38]. Protégé is an open- source, free ontology editor and framework for building intelligent systems.

Ontology is defined as a hierarchy of classes with instances and properties. The protégé assists to create the classes and subclasses and to describe the properties and associations of each class through interfaces. The Resource Description Framework (RDF) scheme used to represent the ontology on the web is automatically constructed. The visualization tools like OntoViz are also provided in Protégé. Certain plug-ins are available for extending the ontology with other applications or concepts.

5.5.2 Information Retrieval from the Ontology-Based Semantic Web for University Domain

The Internet was able to make significant advancements rapidly because of its convenience and the fact that it is easily accessible to anyone around the world. However, there emerged the problem of having too many search results on a single search for specific information requested by the user. The objective of an IR system is to retrieve relevant items which meet a user's information needs. Currently, there is a significant interest in personalized IR which seeks to improve IR effectiveness by incorporating a model of the user's interests [168].

Information storage space and retrieval has a certain level of difficulty, which has been considered from the 1940s. The challenge affirms that massive quantity of information to be stored and relevant information to be precise. Identifying the concept or effort of the user is the major complicated obsession for relevant document searching from a considerable amount of information. The domain information of documents and perception of the user are thus significant for the retrieval of relevant document information. The development of domain ontology for the University domain by identifying the *concepts* of the domain has been already discussed in section 5.3.

A detailed architecture of Semantic Information Retrieval System for University (**SIRSU)**, is proposed as part of this study. The natural language query from the user is passed to the query processing module. The two phases in the query processing module include a query handling phase and a semantic analysis phase. The proposed SIRSU architecture performs the following tasks to retrieve appropriate information.

- User gives the natural language input query to the IR system.
- Tokenizing the input query where key domain terms are extracted and prunes the irrelevant terms.
- Extraction of equivalent key terms from ontology.
- Expansions of extracted key terms with ontology query language, SPARQL.
- Mapping of SPARQL query with RDF database and retrieving the conceptual terms and identifying the context relation.
- Retrieving the result based on the query provided and the mapped concepts.

Figure 5.10 shows the block diagram of the proposed SIRSU architecture.

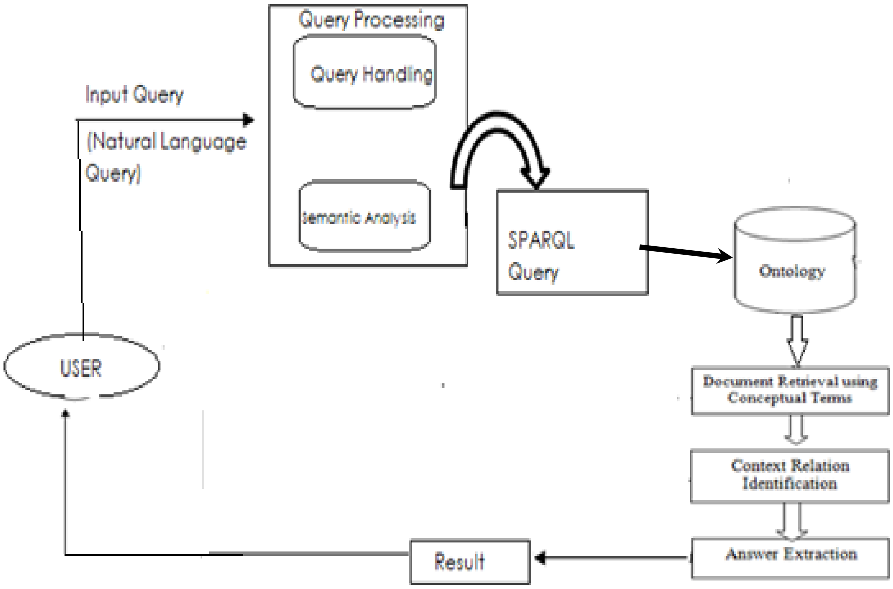


Figure 5.10: Block diagram of the Proposed SIRSU architecture

The query handling module processes the natural language query given by the user. Then key terms equivalent to the classes, subclasses, attributes etc are expanded through semantic query expansion using WordNet [169][170]. The natural language query given by the user is then semantically expanded to SPARQL query language by using a *Quepy* tool of *Python*. *Quepy* is a python framework to transform natural language questions into queries in a database query language. It can be easily customized to different kinds of questions in natural language and database queries. So, it is possible to build a system for providing natural language access to databases. Currently *Quepy* provides support for SPARQL query languages. The *Quepy* converts the query to SPARQL semantically by using NLTK_DATA which contains WordNet. WordNet is used for finding the synonyms of the words and an overview is given in chapter 3. The methods adopted for the conversion of natural language query to SPARQL is described in detail in chapter 6.

In the semantic search phase, meaningful concepts from ontology are extracted. The SPARQL matches the prefix i.e. namespace with RDF namespace then searches data from that URL properties. SPARQL query is a mapping with RDF and retrieves the answer. SPARQL analyses the answer but searches the data which is related to the input query in the knowledge base.

Ontology has been created for a particular domain and is used to model the knowledge for this domain in terms of Concepts (various terms of a specific domain) and relationships between concepts. Ontology shows the hierarchical relationship between different classes and their subclasses in graphical patterns.

Hierarchical view of the ontology designed for the SIRSU system using Protégé is given in Figure 5.11.

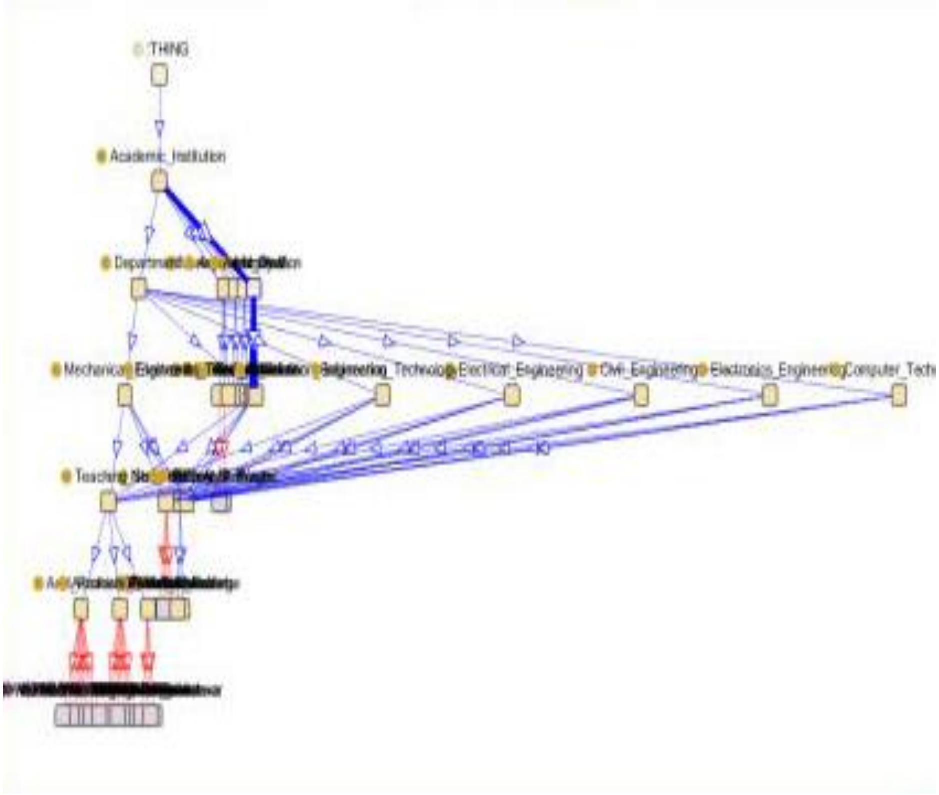


Figure 5.11: Hierarchical view of the domain ontology created for SIRSU

The ontology is then exported in RDF format using Protégé. The following section describes the automatic generation of domain ontology adopted for the University. The following section describes the processes followed for developing the ontology from the document.

5.5.3 Extending Ontology from the Contents in a Document

The domain ontology developed as part of the study is extended with the contents of the documents retrieved by processing the user query. The efficiency of information retrieval depends mostly on the retrieval of less irrelevant information while ensuring relevant information is not ignored.

The number of categories (classes) is fixed subject to the maximum number of *concepts* in the ontology. The lower level categories are defined based on the instances of the *concepts* which appear as the child node of the categories representing the concepts [171]. *Concept* is an idea or notion. Hence, instead of considering words that are present explicitly to represent the text documents, documents can be represented as a bag-of-concepts and the relation between the concepts[172].

Words represent *concepts* in natural language, but the mapping from words in the query to *concepts* is many-to-many. One *concept* may be represented with many different words (synonym), and one word may represent many different *concepts* (polysemy) [173]. Since concepts are abstract entities, WordNet is used as a knowledge base.

In the development of ontology, human involvement is extremely necessary. Even though tools are available to generate the ontology once the concepts are defined by experts, it will only be a semi-automatic process.

To automate the process of ontology generation, algorithms have to be designed to identify the *concepts* and then generate the hierarchy with associations among them. Vijayarajan *et.al.,* proposed a method of extracting Object-Attribute-Value triplets corresponding to the Subject-Predicate-Object of a clause for the purpose of generating the RDF [174].

The document will usually have a header part and body part. The official documents related to the University or Government orders are considered in the study for information extraction. General format of these orders are well defined, consisting of a file ref no, abstract, section, UO/GO number and dated information in the header part.

The body part of these order documents containing the matter related to the header part in detail. As an initial step, the document is prepossessed and stop words are removed as explained in Chapter 3. The ontology development from the document is carried out to extend the previously developed ontology, based on the algorithms discussed below.

Algorithm 5.1: Ontology taxonomy development from the content of a document

Input: doc, the document

Define : Clause: A typical clause consists of a subject and a predicate, where the predicate is naturally a verb phrase and any objects or other modifiers. It can be considered as a sentence in the doc.

```
      Subject (S), Object (O), Predicate (P)
Output: Ontology taxonomy of the S-P-O triples
Step 1. Extract clauses
Step 2. WHILE no more clauses left DO:
    (i)     Analyse the clause and obtain Noun Phrase (NP) and
            Verb Phrase (VP)
    (ii)    Obtain the last occurring Verb from the Verb Phrase
    (iii)   Extract compound entities from NP and VP(Algorithm 2)
    (iv)    Create S-P-O triplets between subjects and objects
    (v)     Semantically analyse the extracted triplets (Algorithm 3)
    (vi)    Create a semantic network by adding the triplets and
            individuals to the ontology
    (vii)   Develop taxonomy (Algorithm 4)
End WHILE
```

The extraction of nested relations from the document, such as X's Y's Z, the triplet extractor continuously checks for relationships and creates empty individuals, which can later be updated based on their future occurrence. The individuals are then classified based on the context where they are used. To extract compound entities from noun phrases (NP), Algorithm 2 is used.

Algorithm 2: Extraction of compound entities from NP

```
Input: NP, the noun phrase
Declare: Subject S, Object O, Predicate P, current token and next
 token
Output: Compound entities
Step 1 WHILE not end of NP DO:
    Step 2 IF next token $\notin$ NP then
        (i)     Create the current token as individual in ontology
    Step 3 ELSE
        (i)     Create S-P-O triplet between current token and next  token
                with O as a combination of both
        (ii)    Update current token with value of O
        (iii)   Set class of O with the value of class S
    End IF
End WHILE
```

To analyze direct relations, such as X is Y, the semantic analyzer determines the group that both individuals belong to, compares them, and accordingly updates the S-P-O triplet based on previous occurrences of both the object and its value. The algorithm for Semantic Analysis of Direct Relations can be formulated as given in Algorithm 3.

Algorithm 3: Semantic Analysis of Direct Relations

```
Input: Subject S-Predicate P-Object O
Output: Compound entities
Step 1 IF $S \in Ontology$ and $S \notin class$ of $O$ THEN
    (i)    O represents a property or a characteristic of $S$
Step 2 ELSE
    (i)    Set class of $S$ with the value of class of $O$ of $P$
END IF
```

Developing a hierarchy among the various identified groups needs hypernyms of all groups which are acquired using WordNet and common ancestors are determined for each entity going up the hierarchy level. This can be continued until the top (Thing) is reached in the ontology. The algorithm 4 is defined for developing the taxonomy as below.

Algorithm 4: Developing a Taxonomy

```
Input: Entity groups
Define: individuals, Taxonomy can be defined as the child class for a class
Output: Taxonomy
Step 1 WHILE no Individual left DO
    (i)   Extract hypernyms for each individual
    (ii)  Arrange the individual in order of appearance in their hierarchies.
    (iii) Find common ancestors between the individual up to their hierarchies
    (iv)  Add individual to a common class having common ancestor
    (v)   Remove individual and add the ancestor as another individual in the given set
END WHILE
```

5.6 Experimental Results and Analysis

In order to evaluate the performance of the information retrieval model based on ontology, the experiment is conducted by varying the number and category of documents. Queries are fired to the corresponding domain of documents and to the mixed domain. When 50 queries are fired to the same domain of documents, the precision and recall are noted. Then 50 queries are fired to the multi domain consisting of 2800 documents. Precision and Recall are the commonly used measures to indicate the quality of the Query processing system as defined in equations (1) and (2) .

$$\Pr ecision = |R_d| / |A| \qquad (1)$$

$$\operatorname{Re} call = |R_d| / |R| \qquad (2)$$

Where $|R_d|$ is the set of relevant documents retrieved, $|A|$ is the set of total documents retrieved and $|R|$ is the set of all relevant documents. The average precision and recall values of ontology based retrieval performed on single domain documents are given in Table 5.1

Table 5.1: Precision and Recall of single domain documents

Category Name	Precision	Recall
Academics	95.25	93.0
Administration	98.1	97.84
Examination	94.38	93.67
Finance	93.47	92.4
Planning and Development	99.24	99

The graphical representation of the precision and recall computed for each category is shown in figure 5.12.

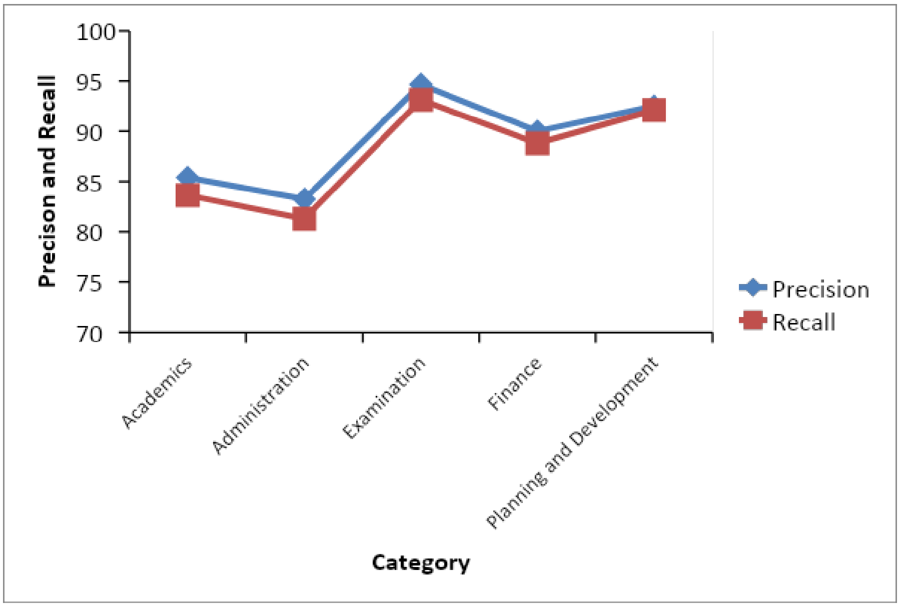


Figure 5.12: Precision and Recall of documents retrieved for single domain documents

Table 5.2 shows the average precision and recall values of ontology based retrieval performed on multi domain documents.

Table 5.2: Precision and Recall of multi domain documents

Category Name	Precision	Recall
Academics	85.4	83.63
Administration	83.23	81.28
Examination	94.65	93.12
Finance	90.0	88.79
Planning and Development	92.49	92.1

The graphical representation of the precision and recall for each category is shown in figure 5.13.

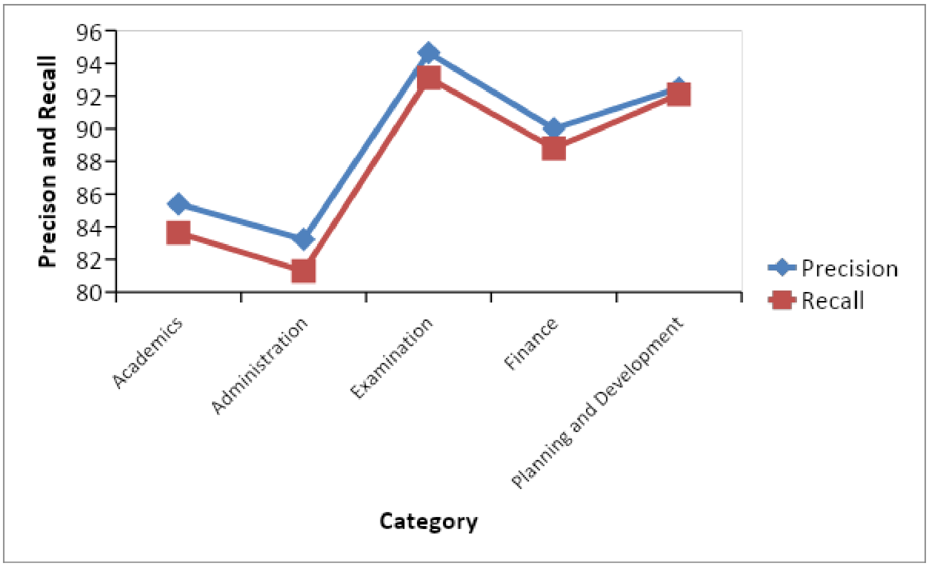


Figure 5.13: Precision and Recall of documents retrieved from multi domain documents

The average retrieval times in milliseconds taken to retrieve the document after the query is fired into the ontology for single domain and multi domain documents and are shown in table 5.3.

Table 5.3: Average retrieval time for single and multi-domain

Category Name	Average Retrieval Time (ms)	
	Single Domain	Multi Domain
Academics	200	245
Administration	194	230
Examination	195	225
Finance	180	230
Planning and Development	150	180

Figure 5.14 is the graphical representation of the average retrieval time for single domain and multi domain documents tabulated in table 5.3

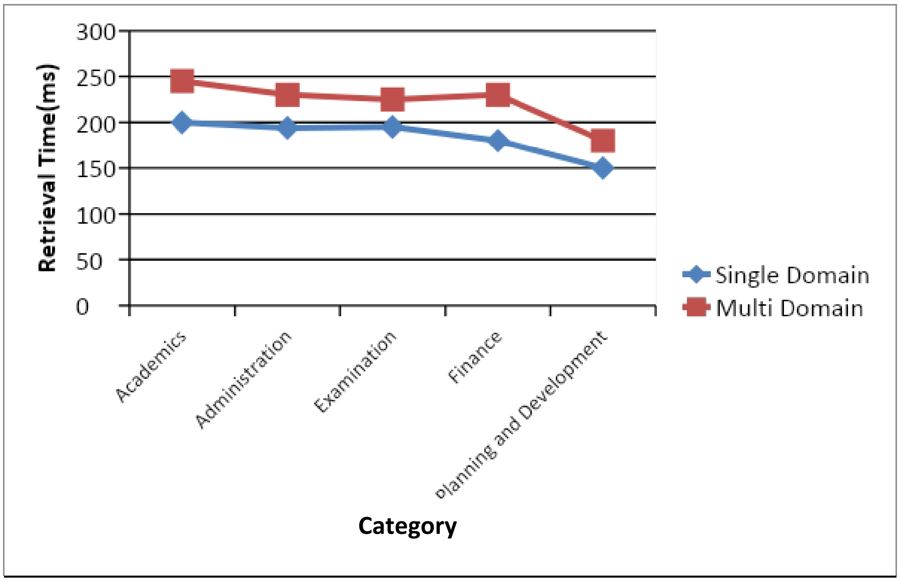


Figure 5.14: Average retrieval time for single and multi-domain documents

In order to better satisfy the users' information needs and to optimize the performance of the retrieval system, ontology is introduced to represent domain information. From the experimental results it is evident that when a query is fired against the domain specific documents, it gives better precision and recall values. When a mixed domain is considered, the performance declines because of the inadequate information available for processing. This may be due to the missing concepts or terms in different queries to indicate specific information in the related domains. This impreciseness and vagueness can be dealt with by incorporating Fuzzy logic with ontology as considered in chapter 7.

5.7 Conclusion

Developing a semantic information retrieval system for the University domain consists of the following steps:

(i) Crawl various University and Government websites for the documents and classify them based on the category they belong to.

(ii) Convert the unstructured text to structured RDF form (triple form) after necessary pre-processing.

(iii) Build the ontology for the University domain.

(iv) Convert the natural language query from various stakeholders to query specific to RDF

(v) Fire the SPARQL query to the developed ontology to retrieve information on the specific domain.

To assimilate machine-readable data on the current human-operable web for a particular domain of interest, the knowledge capturing is done with Ontology. The most crucial step involved here is the development of Ontology, which is illustrated with a model developed in the University domain.

The system proposed can overcome the limitations of keyword-based searching. Instead of providing a list of all the documents having related information, it can excerpt the pertinent information. SPARQL query is used to extract the RDF data with respect to the user input query. Protégé tool is used for creating ontology in RDF data format. By using the Protégé tool, ontology for the University domain is created. The proposed system retrieves answers to queries related to a specific domain of interest. To evaluate the proposed model precision and recall are measured for the queries against single and multi-domain documents. For the single domain documents, it gives precision values between 93.47 to 99.24 and the recall values between 92.4 and 99. When multi domain documents are considered the precision values obtained are between 83.23 and 94.65 and the recall value falls between 81.29 and 93.12. The average retrieval time for single and multi-domain documents are also considered as part of the study. For the single domain, the minimum time obtained is 150 ms and the maximum time is 200 ms. For the multi domain it is 180 ms and 245 ms respectively. The results obtained are comparable with the existing methods with promising accuracy. From the analysis, it is concluded that if the vagueness present in the query terms are modelled using Fuzzy Logic the results can be improved.

Chapter 6

Short Query Processing for the Semantic Web

6.1 Introduction

Human-Computer interaction is the most cherished dream of computer scientists from the development of the digital computer itself. For accurate and efficient communication, a computer must understand natural language and it must respond naturally. Natural Language Processing (NLP) deals with the problem related to Human-Computer interaction. Information Retrieval from a data repository with a natural language query is an application of NLP systems.

A system that takes user queries in natural language as input and translates it into a low-level expression to execute by the machine can be considered as a query processing system. It searches for the matching document in a set of documents to extract the precise answer. A query expressed in ordinary spoken language is a natural language query which is in a form severe for machines to process. Information Retrieval (IR) or Information Extraction (IE) is different from query processing. IR system presents the users with a set of documents that are related to user questions but do not precisely indicate the correct answers. The query processing system uses the techniques from

IR / IE and provides precise answers to open domain queries formulated naturally. IR systems can locate documents containing important information and leave it to the user to extract useful information from a collection of ranked documents.

One of the main components of the query processing system is the information retrieval engine located on the top of the document collection and handles retrieval requests [175]. Another component is a query interpretation system that converts natural language questions into keywords or queries for the search engines to fetch the vital documents from data sources that can potentially answer the queries. The third component is the answer extraction component, that will help to analyze documents and extract answers from them.

In this study, the Ontology-based system is used for information retrieval. The query from the user is expressed in natural language. The objective of this chapter is to convert the natural language queries to SPARQL for working with the designed ontology. Rest of this chapter is organized as follows. Section 6.2 gives an overview of the types of query processing systems. Section 6.3 presents different approaches to query processing. Developing a query processing system for the semantic web is explained in section 6.4. How a natural language query is converted into SPARQL is described in section 6.5 and its subsections. Experimental setup and the evaluation of results is conducted in section 6.6. Finally, section 6.7 concludes the chapter.

6.2 Types of Query Processing Systems

There are two significant types of query processing systems, such as closed domain query processing systems and open domain query processing systems. The closed-domain query processing system deals with questions under a specific domain and can be seen as an easier task. In closed-domain query processing, only a limited type of questions are being asked. The open domain query processing system deals with questions about anything. On the other hand, these systems typically have much more data available, from which to extract the answers [176].

6.3 Approaches to Query Processing

In this section, different approaches for query processing are briefly described. They include linguistic approaches, statistical approaches, and pattern matching approaches for building query processing systems.

(a) Linguistic Approaches

The linguistic approach understands natural language text and uses common knowledge linguistic techniques such as tokenization, part of speech tagging, and parsing. These were implemented to user's queries for formulating it into a precise query that merely extracts the respective response from the structured datasets [177].

(b) Statistical Approaches

Obtainability of a considerable amount of data on the Internet increases the importance of statistical query processing approaches. A statistical learning method gives better results than other methodologies. Online text sources and statistical approaches are independent of structured query languages and can formulate queries in natural languages. The Statistical query processing systems use techniques such as Support Vector Machine (SVM) classifier, Bayesian classifier, and Maximum Entropy model, *etc*. [178].

(c) Pattern Matching Approaches

The pattern matching method deals with the expressive power of text patterns. It replaces the sophisticated processing involved in other computing approaches. Most of the query processing systems based on pattern matching practice the surface text pattern, even though some rely on templates for the response generator.

The following section describes the significant steps involved in the development of the query processing system.

6.4 Developing A Query Processing System for Semantic Web

Query processing is typically based on formalisms assuming a closed world in Databases, and uses the principle of negation as failure [179]. Since the semantic web needs to deal with incomplete knowledge, that kind of assumption is not adequate. The semantic web uses the background knowledge referred to by the ontology to cope with this incompleteness. The ontology specifies classes of objects, subclass relations between these classes, and typed relations between classes. Since ontologies are represented with RDF, SPARQL is the best choice to query the RDF data. Figure 6.1 shows the proposed architecture of the query processing system for the ontology-based semantic web.

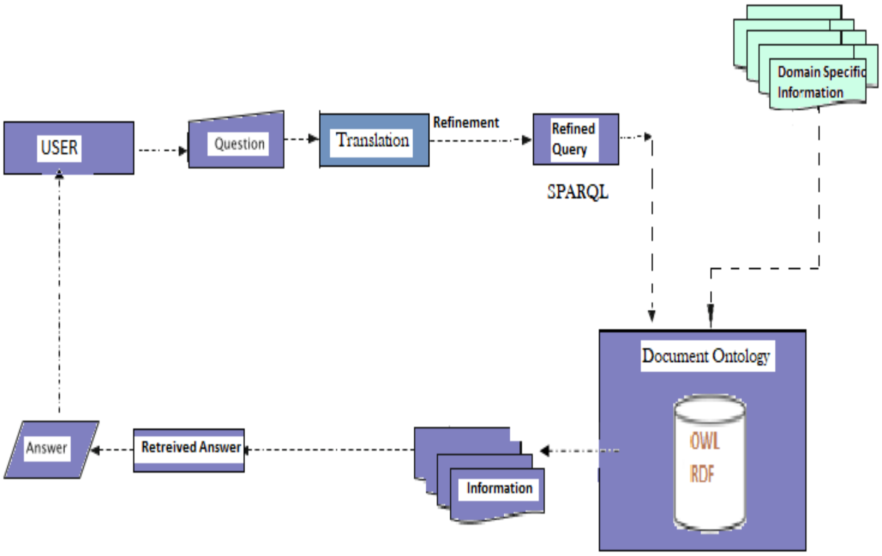


Figure 6.1: Query processing system for the Ontology-based Semantic Web

SPARQL is the RDF query language which is a recursive acronym for SPARQL Protocol and RDF Query Language. It can be used to query, retrieve, and manipulate semantic data stored in Resource Description Framework (RDF) format [180] [181]. The RDF Data Access Working Group (DAWG) of the World Wide Web Consortium made it as a standard and is recognized as one of the significant technologies of the semantic web [182].

RDF data can be considered as a table consisting of three columns – the subject column, the predicate column, and the object column. The subject in RDF is equivalent to an entity in a SQL database. In an SQL database, data elements or fields for a given object are retained in multiple columns, occasionally range across multiple tables, and identified by a key. In RDF, those fields are denoted as distinct predicates (object rows) sharing the same subject, often the same unique key, where the predicate is corresponding to the column name and the object is corresponding to the actual data [183].

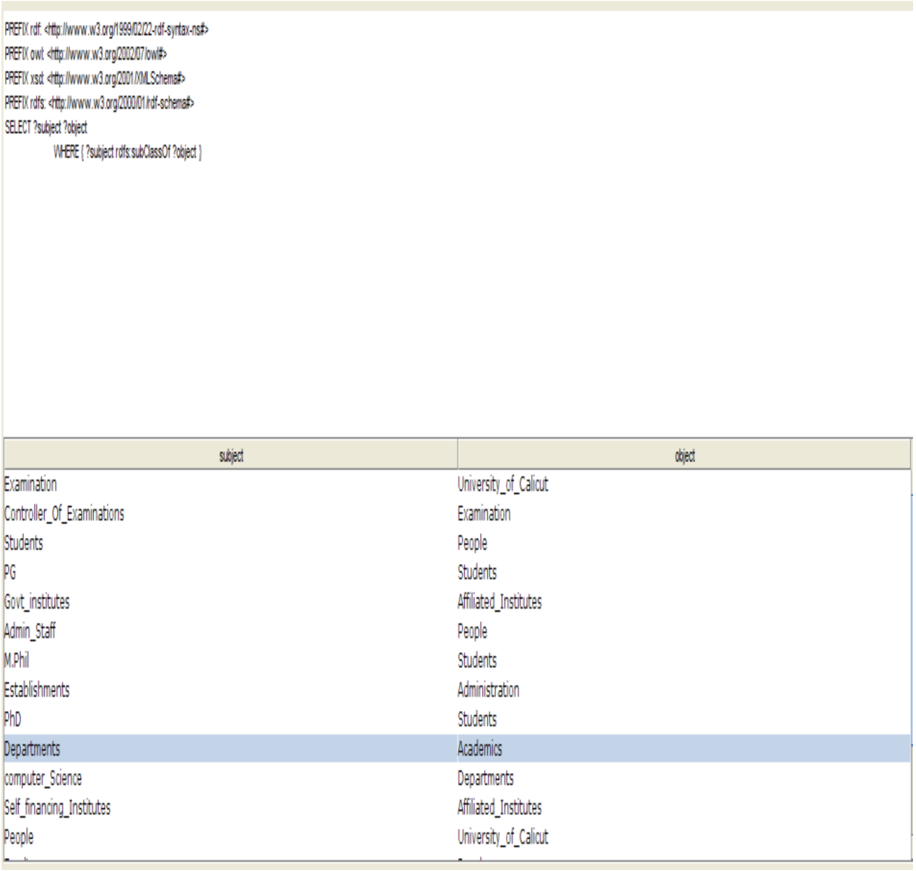
The proposed system uses SPARQL to query the University ontology. Since the interface accepts user input in the form of natural language questions, it should first be converted into the corresponding SPARQL query. The development of ontology for the University domain with the Protégé tool has already been described in Chapter 5. This tool provides an interface to query the ontology with SPARQL.

The query specified is

```
SELECT ?Subject ?Object
```

WHERE {?subject rdfs subClassof ?object}

To retrieve the subject and object components defined in the University ontology. Figure 6.2 shows the result when a SPARQL query is fired against the ontology.



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subject ?object
        WHERE { ?subject rdfs:subClassOf ?object }
```

subject	object
Examination	University_of_Calicut
Controller_Of_Examinations	Examination
Students	People
PG	Students
Govt_institutes	Affiliated_Institutes
Admin_Staff	People
M.Phil	Students
Establishments	Administration
PhD	Students
Departments	Academics
computer_Science	Departments
Self_financing_Institutes	Affiliated_Institutes
People	University_of_Calicut

Figure 6.2: Sample SPARQL query results

The data acquisition and pre-processing has been dealt with in Chapter 3. The query dataset used for this study named Short Queries for University Services in English (SQUSE) consists of 7500 queries framed in English. The following section presents the translation process of natural language query to SPARQL to be used with the ontology.

6.5 Conversion of Natural Language Query to SPARQL

While translating natural language questions into SPARQL or any other specific queries, the first step is to identify the category of the question. This will help to relate these categories to the identical concepts in the ontology to retrieve accurate results [184].

This procedure consists of different steps. In the first step, the input query type is identified. According to the query type, the natural language query can be categorized into four types, as listed below.

- Definition questions: Those begin with "What is/are" or "What does it mean."
- Boolean or Yes/No questions
- Factual questions: That gives a fact or precise information as an answer.
- List questions: Those for which the answer is a list of entities

- Complex Questions: that begins with "How" or "Why." For these types, a precise answer is almost impracticable.

In the second step, the probable answer, which is necessary to relate the question to the domain, is identified. Then the next step consists of extracting *Entities* from the given questions and their expected answers, which is identified in step 2. In the fourth step, *entity* types of answers such as class, data property, object property, annotation, axiom, and instance, etc. are to be identified in the ontology. The final step is the construction of the SPARQL query based on the processes completed in the previous four steps [185]. Table 6.1 gives examples for questions under various categories.

Table 6.1: Sample queries under various categories

Categories	Examples
Definition questions	What is an additional degree?
Yes/No questions	Can a student under regular mode continue through distance education during the course?
Factual questions	What is the fee for confidential mark lists?
List questions	Who all are the UG students of Nirmala College of Engineering, ChaIakkudy avail condonation?
Complex questions	If there is name correction necessitated, how can it be done?

The proposed method for translating the natural language query to SPARQL is given in figure 6.3.

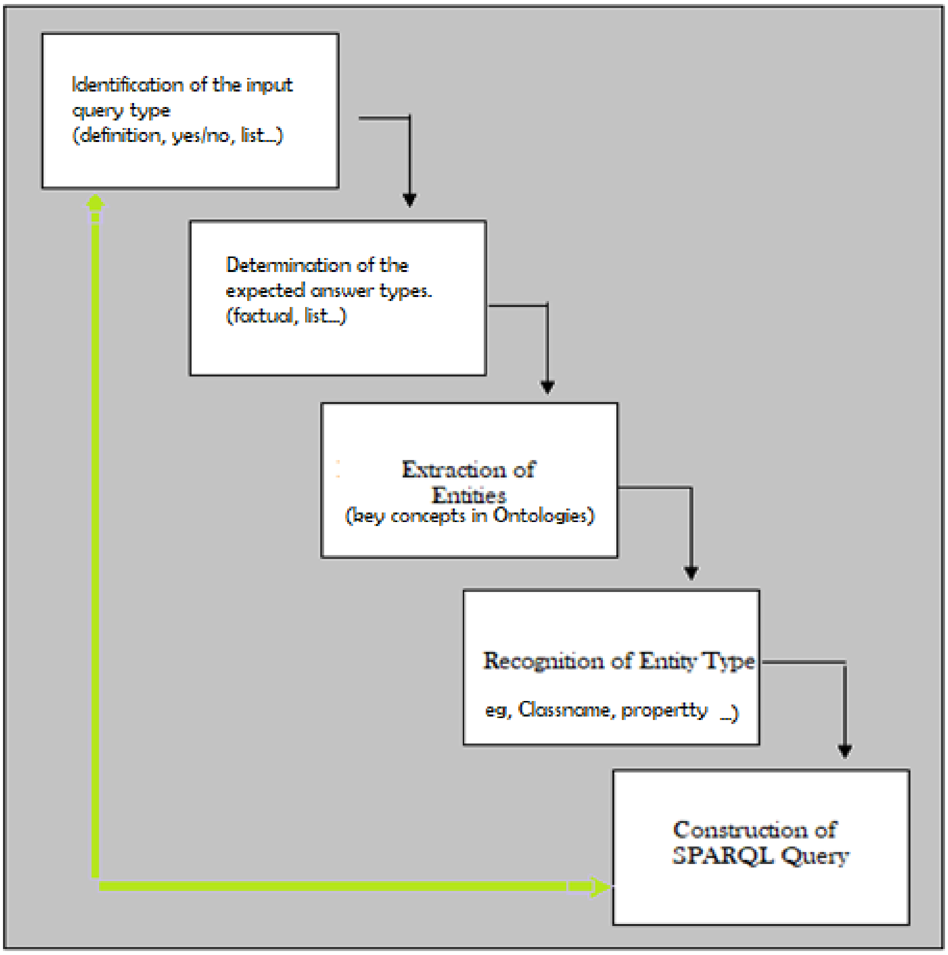

Identification of the input query type (definition, yes/no, list…)
Determination of the expected answer types. (factual, list…)
Extraction of Entities (key concepts in Ontologies)
Recognition of Entity Type
eg, Classname, propertty …)
Construction of SPARQL Query

Figure 6.3: Translation of natural language question to SPARQL

Based on the method given, the conversion of natural language queries into SPARQL queries are carried out. As a first attempt to detect ontology requirements, the query category is identified. It may be any domain of the university namely: administration, academics, examination, finance, and planning and development are identified. A subset of subdomains considered in this study is given in table 6.2.

Table 6.2: Domains and subdomains of University Ontology

Domain	Subdomains
Administration	Senate
	Syndicate
	Academic Council
	Board of Studies
	Students Council
Academics	Departments
	NSS
	Courses
	Affiliated Colleges
	Off-Campus centers
Examination	Registration
	Time table
	Syllabus
	Results
	Revaluation
Finance	Pension
	Payroll
	Salary
	Provident Fund
	NSS Expenditure
	Remuneration to contract staff
	Maintenance of furniture
	Pay fixation
Planning & Development	State plan grant
	RTI
	Digital India
	Global education meet
	Hostel Development

The following section describes the construction of a SPARQL query from a natural language query.

6.5.1 Construction of SPARQL Query

The category of questions will help in the construction of the corresponding SPARQL queries. For instance, in the case of definition queries, to provide as much information as possible, a set of SPARQL queries are to be combined. This combination is divided into five sets, such as

i. Super classes

ii. Sub classes

iii. A List of descriptive properties or data properties

iv. Relations or object properties

v. Annotations such as definition, comment, label, etc.

Whenever a definition query is fired, this combination may not be complete and may hinge on the scope of the anticipated answer.

In the case of Yes/No questions, it should provide a Boolean type response. Hence ASK form of the SPARQL query is preferred over the other forms, like SELECT, DESCRIBE, or CONSTRUCT. These are the four types of SPARQL queries.

Table 6.3 gives an overview of the different types of SPARQL queries and their corresponding answer types [186][187][188].

Table 6.3: SPARQL query types

SPARQL QUERY TYPE	TYPE OF PROCESSING	ANSWER TYPE
SELECT	Project out variables and expressions	Results in a table of values
CONSTRUCT	Constructs RDF triples or graphs	Results in RDF triples (S-O-P)
ASK	Asks for any matches	Results in either TRUE/FALSE
DESCRIBE	Describes the resources matched by nay variables	Results in RDF triples

The objective of the factual questions is one specific entity, which might be a class, an instance, or whatever it may be, whereas for the list questions, expect to obtain several entities as a single answer. A detailed answer is expected for Complex questions, and in most cases, ontology annotations can be considered as the best answers to this type of questions.

The likely answers can be determined for the questions fired may come from several information sources, such as official academic websites, official reports, and notifications from the university, experts' opinion, etc. Table 6.4 depicts sample queries and the corresponding answers under various domains / subdomains used in this study.

Table 6.4: Sample query and Possible answers from various domains or subdomains

Sl.No	Domains/ Subdomains	Examples	Corresponding Answers
1	Academics/ courses	What is credit?	Each course bears a specified number of credits. In general, the number of credits a course carries is determined by the number of class hours the course meets each week.
2	Academics/ Teacher/ Researcher	Must a university teacher be a researcher?	Nearly all faculty members are expected to engage in research.
3	Administration/ Governing board	What average size and duration have a governing board?	The average size of public boards is approximately ten people, and the average size among independent (private) institutions is 30. The length of board members' terms varies from three years to as long as 12 years.
4	Examination/ Time table	When will the examination time table be published?	Examination time tales will be published ten days before the commencement of exams.
5	Administration/ departments	Why are universities organized into departments?	The basic unit of academic organization in most institutions is the department (e.g., chemistry, political science). Every department belongs to an academic field.

As the next step, the extraction of *entities* from both queries and their expected answers is carried out. This extraction is centred on a mapping between relevant terms in question answer pairs and their equivalent *concepts* in the ontology [189]. Table 6.5 gives an extract of sample *entities*.

Table 6.5: Entities extracted from Questions and their Answers

Relevant Terms from questions	**Relevant Terms from answers**	**Corresponding concepts in the ontology**
Credit?	…course … number of credits.	Course, a credit number
teacher … researcher ?	engage in research	Teacher, researcher
…size ..duration… governing board?	…10 …30 people …varies from three years to as long as 12 years	Size, duration, governing board
…ranks…teacher?	Assistant Professor, Associate Professor, Full Professor, Professor Emeritus	Rank, Teacher, Assistant Professor, Associate Professor, Full Professor, Professor Emeritus
universities … organized into departments?	… basic unit … is the department… Every department belongs to an academic field.	Higher education organization, department

Extraction of entities from both questions and their expected answers consists of a physical extraction of appropriate terms which preferably should be equivalent to some ontology entities. Otherwise less chance for the SPARQL query to obtain a suitable answer encoded in the ontology [184][188]. This situation is also required to warn the ontology evaluator, which is essential to update the ontology by adding the missing entity.

To identify Entity Type and Entity Location, questions and answers must be represented in one of the allowed forms of ontology entities like: classes, data properties, object properties, axioms, instances and annotations. The details of proposed ontology and its development process has already been explained in chapter 5 in detail.

SPARQL query syntax is highly dependent on the entity type of the expected answer.

When the answer is an INSTANCE, the SPARQL query will then be:

```
SELECT * WHERE

{?Fee rdf:type UNIV:marksheet . }.
```

If the answer is a CLASS, the SPARQL query will then be:

```
SELECT * WHERE

{ ?subclass rdfs:subClassOf UNIV:Student . }.
```

The location of the expected answer in the ontology plays another crucial role in constructing an efficient SPARQL query. The required information can directly be targeted with this parameter. For example, when the expected answer is located in an annotation which can be either definition, label or comment of a class, the SPARQL query will then be like

```
SELECT ?definition WHERE

{UNIV: additional_degree rdfs:isDefinedBy ?definition. }.
```

The translation process of competency questions is done by identifying entity types of each extracted entity from the question and answer pair. Locating it in the ontology is done using a combined action of the ontology editor search function, and the support of an ontology developer who knows the vocabulary and the syntax of the ontology. The experimental result obtained on this identification is presented in Table 6.6

Table 6.6: Entity types and Locations in the ontology

Entity Type	Entity Locations in the ontology
Class: Course Data Property: CourseCreditsNumber	CourseCreditNumber Domain Course
Classes: Teacher, Researcher	Teacher SubClassOf Researcher
Class: Governing Board Data Properties: Size, Duration	GoverningBoardSize Domain GoverningBoard GoverningBoardDuration Domain GoverningBoard
Class: Teacher Data Property: Rank, Assistant Professor, Associate Professor, Professor	TeacherRank Domain Teacher AssistantProfessor SubPropertyOf TeacherRank AssociateProfessor SubPropertyOf TeacherRank Professor SubPropertyOf TeacherRank
Classes: Higher Education Organization, Department	Department SubClassOf Faculty Faculty SubClassOf Role Role SubClassOf HigherEducationOrganization Department Definition

The process of constructing SPARQL query can be started once the identification of ideal answers, the recognition of equivalent entity type and localization in the ontology are completed. When a single SPARQL cannot provide all identified entities, it is possible to interpret a natural language question into several SPARQL queries. Another possibility is to make an UNION between all necessary sub queries.

Information retrieval system for University domain will receive queries from users which are specified in natural or spoken language. So it provides a user friendly atmosphere where the user need not worry about the learning of SPARQL for retrieving data from RDF/OWL based databases. The natural language query (NLQ) is converted into a SPARQL query using QUEPY framework [185][186][190].

The transformation from natural language to SPARQL is done using a distinct form of regular expressions:

person_name = Group(Plus(Pos("NNP")), "person_name")

regex = Lemma("who") + Lemma("be") + person_name + Question(Pos("."))

And formerly using a suitable means to precisely express semantic relations:

person = IsPerson() + HasKeyword(person_name)

definition = DefinitionOf(person)

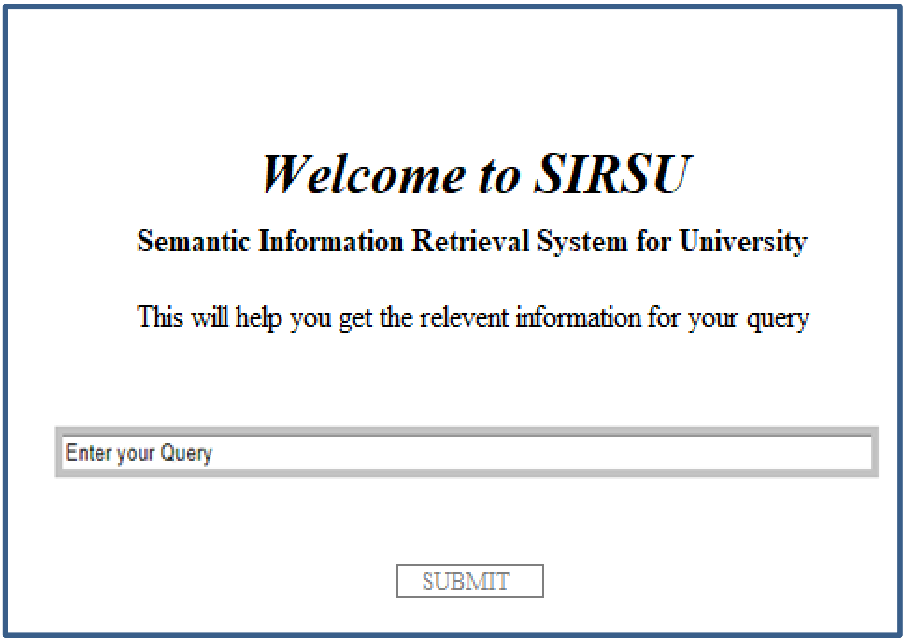
The rest of the transformation is controlled automatically by the framework to finally produce this SPARQL:

```
SELECT DISTINCT ?x1 WHERE

{ ?x0 rdf:typefoaf:Person.

?x0 rdfs:label "Babbage"@en.

 ?x0 rdfs:comment ?x1.

}
```

The implementation of query conversion is then carried out based on the procedure described in this session. The experimental setup and the analysis of the results are detailed in the following section.

6.6 Experimental Setup and Result Evaluation

The natural language query given by the user needs to be converted into a SPARQL query for information retrieval from the ontology based semantic web. A GUI designed with Python Flask for this purpose is shown in figure 6.4. It allows the user to enter a query in natural language which is processed and converted into the corresponding SPARQL query which is fired into the ontology for information retrieval.



Welcome to SIRSU

Semantic Information Retrieval System for University

This will help you get the relevent information for your query

Figure 6.4: GUI designed to input user query to be converted into SPARQL

Experiments are conducted in five steps as follows. In the first step, the identification and disambiguation of named entities is performed. As a second step parsing and disambiguation of the natural language query is conducted. It is a method to determine which meaning of a word is activated *by the use of word* in a particular context [187][191]. Matching query terms with ontology *concepts* and *properties* are dealt in the third step. Generation of candidate triples is carried out in the fourth step. Then the integration of triples and generation of SPARQL queries are done as the final step.

The examples of natural language queries and the corresponding SPARQL equivalents are given in table 6.7.

Table 6.7: Natural language Queries and corresponding SPARQL equivalent

Sl. No	Natural Language Query	SPARQL Equivalent

1	What is a credit?	SELECT ?comment
		WHERE
		{
		UNIV:CourseCreditsNumber
		rdfs:comment ?comment
		}

2	Must a university teacher be a researcher?	ASK {UNIV:Teacher rdfs:subClassOf UNIV:Researcher. }

3	What is the average size and duration of the governing board?	SELECT ?university ?size
		WHERE { ?university
		rdf:type UNIV:
		HigherEducationOrganization;
		?y
		rdfs:subClassOf ?university ; ?y
		UNIV:GoverningBoardSize ?size
		}

		SELECT ?university ?duration WHERE { ?university rdf:type UNIV:HigherEducationOrganization ; ?y rdfs:subClassOf ?university ; ?y UNIV: GoverningBoardDuration?duration }
4	What are the probable academic ranks of a teacher?	SELECT ?a ?b ?c ?d WHERE {?a rdfs:subPropertyOf UNIV:TeacherRank. ?b rdfs:subPropertyOf ?a . ?c rdfs:subPropertyOf ?b . ?d rdfs:subPropertyOf ?c . }
5	Why are universities organized into departments?	SELECT * WHERE { UNIV:Department rdfs:subClassOf ?x ; OPTIONAL {

		?x rdfs:subClassOf ?y ; OPTIONAL { ?y rdfs:subClassOf UNIV:HigherEducationOrganization }}}

		SELECT ?definition
		WHERE
		{
		UNIV:Department
		rdfs:isDefinedBy ?definition .
		}

The correctness of translation from NL query to SPARQL is measured with the accuracy of the answers. A set of questions containing 100 natural language queries were fired to the ontology. The results are tabulated on the type of queries, and a confusion matrix is created as in Table 6.8. Based on its values, performance factors including precision, recall and F-measure are evaluated using the equations 6.1, 6.2 and 6.3 respectively.

$$\mathrm{Pr}\,ecision = TP\,/\,(TP + FP) \qquad (6.1)$$

$$\mathrm{Re}\,call = TP\,/\,(TP + FN) \qquad (6.2)$$

$$F\ \ measure = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Re}\,call}{(\mathrm{Pr}\,ecision + \mathrm{Re}\,call)} \qquad (6.3)$$

Table 6.8: Confusion matrix for 100 converted queries

ACTUAL	PREDICTED		
		Positive	Negative
	Positive	TP=72	FN=14
	Negative	FP=5	TN=9

The Computed values of precision, recall and F measure for the converted natural language queries are given in table 6.9

Table 6.9: Performance measures of the converted natural language queries

Performance Measure	Value
Precision	93.51%
Recall	83.72%
F -Measure	88.34%

A comparative study of the performance of the proposed system with similar systems reported in the literature used for converting natural language query to structured query is summarized in the table 6.10.

Table 6.10: Comparison of the proposed system with other existing systems for converting natural language to structured language

Reference	Year	Structured Language	Precision (%)	Recall (%)	F Measure (%)
Damiljanovic *et.al.,* [185]	2011	SPARQL	77	68	74
Garima Singh *et. al.,* [192]	2016	SQL	94.1	80	86.4
Tatu et.al., [193]	2016	SPARQL	78.20	65.82	85.29
Ferre S *et.al.,* [194]	2017	SPARQL		-	-
Coelho *et.al.,* [195]	2019	SPARQL	87	86	86
Proposed Method	2019	SPARQL	93.51	83.72	88.34

It is evident from the experimental results that the proposed method improves the performance of the query translation process. The analysis of the results shows that the decrease in the performance measure of precision is due to the missing of the critical domain specific required to answer the query.

6.7 Conclusion

The translation of Natural language questions into SPARQL queries is an essential step in the retrieval of information from the ontology. When it is a closed domain, we can identify the concepts more accurately when compared to an open domain problem. A well-defined approach of this transformation process is critical for the ontology assessment in particular and machine-readable question answering in a more general outlook.

This chapter presents an overview of query processing for an ontology-based information retrieval system. The natural language query, which is expressed in the spoken language, is converted into the corresponding representation in SPARQL to be fired onto the ontology developed in the University domain. To measure the performance, the metrics such as precision, recall and F measure are computed and compared with the previous works reported in the literature. The recall value improves compared to the previously reported work with the accuracy of more than 3% and F measure is improved by 2% in the proposed work. The precision obtained is declined with a value of 0.59. This performance degradation points to the unavailability of critical domain specific information which are essential for precisely answering the query.

Chapter 7

Implementation of a Fuzzy Ontology in University Domain for Query Answering in Semantic Web

7.1 Introduction

The semantic web is a vision that aims to solve the problem faced by the World Wide Web users by utilizing smart web agents that retrieve useful, meaningful and relevant results. In order to obtain precise and high quality information from the search engines, ontologies that form an important component of the semantic web are used for communication among the web agents [196]. The conventional ontologies do not acknowledge human perception in the desired way. Hence, there is a need to introduce fuzziness in the ontologies.

Domain knowledge can be incorporated using ontology in information retrieval tasks. The concepts of the documents are represented as classes and their associations in the corresponding ontology. To characterise vague information associated with user query in natural language, fuzzy logic can be integrated into ontology. Typically, fuzzy ontology is constructed using a predetermined concept hierarchy which is used to address the different classes of users with different expertise levels and information needs.

In semantic web, machines require the description of the web resources for manipulating these resources. Several languages including OWL (Web Ontology Language), OIL (Ontology Interchange Language), DAML (DARPA Agent Mark-up Language) etc. have been defined for the purpose of manipulating web resources. They can express data and metadata. With this intention, RDF Schema (Resource Description Framework Schema) is used in this study to specify the *concepts* and their properties and values. An RDF data model, rather called ontology, to represent the University Information System is designed. The details regarding the development of ontology and querying have already been discussed in chapter 5 and 6. But there may not always be a clear cut boundary between *concepts* of the domain due to the vague constructions of natural language.

The objective of the study described in this chapter is to represent the uncertain values confined in the query, by introducing a new data type referred to as fuzzy linguistic variables to the RDF data model. The semantic query expansion in SPARQL is constructed by order relation, equivalence relation and inclusion relation between fuzzy concepts defined in linguistic variable ontologies [197]. Results show that this study assists the semantic retrieval of information through fuzzy concepts for University domain. The rest of this chapter is organised as follows. The section 7.2 describes fuzzy linguistic variables and member functions. Section 7.3 describes the important properties of fuzzy ontology in relation with crisp ontology. Section 7.4 studies information retrieval from a system based on fuzzy ontology. Generation of fuzzy ontology by incorporating fuzzy information in the existing ontology is defined in the subsections. Querying the developed fuzzy ontology and its evaluation are performed in section 7.5. Finally section 7.6 concludes the chapter.

7.2 Fuzzy Linguistic Variables and Member Functions

The problem to deal with imprecise concepts has been addressed several decades ago, which caused the so-called fuzzy set and fuzzy logic theory and a huge number of real life applications are benefited with this Fuzzy Logic [198]. To achieve the knowledge share and reuse for fuzzy systems on the semantic web, it is necessary to represent the fuzzy linguistic variables with ontology. In this work a fuzzy extension of the ontology description language OWL DL is considered, and presents its syntax and semantics. The main feature of fuzzy DL is that it allows representing and reasoning about vague or ambiguous concepts.

Fuzzy sets have been introduced by Zadeh as a way to deal with vague concepts [199]. Formally, a fuzzy set A with respect to a universe X is characterized by a membership function

$$\mu A : X \rightarrow [0, 1] \qquad (7.1)$$

Assigning a membership degree using a membership function μ, $\mu A(x)$, to each element x in X. $\mu A(x)$ gives an estimation of the belonging of x to A. Typically, if $\mu A(x)=1$ then x definitely belongs to A, while $\mu A(x)=0.8$ means that x is "close" to be an element of A with a degree of membership 0.8. When switched to fuzzy logics, the notion of degree of membership $\mu A(x)$ of an element x ∈ X w.r.t. a fuzzy set A over X is regarded as the degree of truth in [0, 1] of the statement "x belongs to A".

Fuzzy linguistic variable is a 3-tuple (X, T, M), where:

- X is a name of fuzzy linguistic variable, e.g., "duration" or "salary".
- T is a set of terms which are the values of the fuzzy linguistic variable, e.g. T = {*short, normal, long*} or T = {*high, medium, low*}.
- M is a mapping rule which map every term of T to a fuzzy set.

The value of a fuzzy linguistic variable is the term or concept in natural language to be represented in ontology.

7.2.1 Membership Functions

Membership function characterises fuzziness. It is a generalization of the indicator function for classical set by the graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, defines functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets on final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different membership functions associated with each input and output response. The commonly used functions are trapezoidal function and triangular function. The trapezoidal function $trz(x:a,b,c,d)$ is defined as in equation 7.2.

$$trz(x:a,b,c,d)=\begin{cases}\dfrac{x-a}{b-a} & if\ a\le x\le b\\ 1 & if\ b<x\le c\\ \dfrac{d-x}{d-c} & if\ c<x\le d\\ 0 & if\ x<a\ or\ x>d\end{cases} \tag{7.2}$$

The triangular function $tri(x:a,b,c)$ is defined as in equation 7.3

$$tri(x:a,b,c)=\begin{cases}\dfrac{x-a}{b-a} & if\ a\le x<b\\ 1 & if\ x=b\\ \dfrac{x-c}{b-c} & if\ b<x\le c\\ 0 & if\ x<a\ \ or\ \ x>c\end{cases} \tag{7.3}$$

7.3 Comparison of Crisp Ontology and Fuzzy Ontology

Fuzzy logic emulates human thinking, cognition and inference and it is designed in a way such that it can be processed by the computer. Fuzzy logic is the theory of fuzzy sets, sets that express uncertainty. Fuzzy logic is based on the concept of membership degrees. Fuzzy set theory mimics human perception in its application of vagueness and impreciseness to make decisions. It was designed to mathematically represent uncertainty for dealing with the inbuilt vagueness in some domains. Fuzzy logic is based on the mathematical concepts for depicting knowledge based on degrees of membership. The classical logic consists of only two values i.e. true and false and it has constraints in dealing with problems related to the real world domain. Fuzzy logic uses a continuum of logical values between 0 and 1. It rests on the idea that things can be partly true and partly false at the same time [200][201]. In the present work, a crisp ontology belonging to the

University domain has been fuzzified based on the keywords obtained from the queries given by the users. The need for fuzzification arises in order to introduce an improved mapping to the domain of the document which could provide better information. Table 7.1 gives a sample list of keywords noted from the query which are selected for fuzzification.

Table 7.1: Sample fuzzy linguistic variables to be represented in the ontology

No	User Query	Fuzzy concepts	Fuzzy linguistic variables
1	What are the short term courses provided by the University	Duration	Short Normal Long
2	Designation of the highly paid staff in the University	Salary	High Medium Low
3	Which course is good for a student completed Humanities in plus two level	Eligibility	Good Fair bad
4	Which is the research topic in chemistry having large number of research outcome	Research_ Publications	Large Average Less

Obtaining correct and relevant information at the right time to a user's query is quite a difficult task. This becomes even more complex, if the query terms have many meanings and occur in different varieties of domain. This study focuses on the fuzzy extension of the classic ontology built for the university domain.

A crisp ontology is a precise (i.e., binary) specification of a conceptualization. In other words, it is an enumeration of the accurate concepts and exact relationships that prevail for any information accumulation. In crisp ontology, the domain knowledge [202] is organized in terms of

concepts (O)

properties (P)

relations (R) and

axioms (A)

It is formally defined as a 4 – tuple

$O = (C, P, R, A)$

where:

- C is a set of concepts defined for the domain. A concept corresponds to a class.
- P is a set of properties of *concepts*
- R is a set of two fold semantic relations defined between the *concepts* in C.

- A is a set of axioms and it is a real fact or a reasoning rule.

7.3.1 Fuzzy Ontology

Fuzzy ontologies are described as an extension of crisp ontologies of a particular domain for resolving the uncertainty or inaccuracy problems. Impreciseness and inaccuracies are often encountered in the present systems [203][204]. Fuzzy ontology aims to encapsulate the vagueness in itself by adapting the uncertainties and bringing forth a view which is machine processable and interpretable. A fuzzy ontology can be represented as a 7-tuple

$$O_F = (C, P, C_F, P_F, R, R_F, A_S, A_{SF}, A)$$

where:

- C is a set of crisp concepts defined for the domain.
- P is a set of crisp concept properties.
- C_F is a set of fuzzy *concepts*
- P_F is a set of fuzzy *concept properties*
- R is a set of crisp binary semantic *relations* defined between concepts in C or fuzzy *concepts* in C_F .

- R_F is a set of fuzzy binary semantic relations defined between crisp *concepts* in C or fuzzy concepts in C_F

- A_S is a set of crisp binary associations defined between *concepts* in C or fuzzy *concepts* in C_F .

- A_{SF} is a set of fuzzy binary associations defined between crisp *concepts* in C or fuzzy concepts in C_F .

- A is a set of axioms. An axiom is a real fact or reasoning rule.

The fuzzy ontology defined for the first query " What are the short term courses provided by the University?" is shown in figure 7.1.

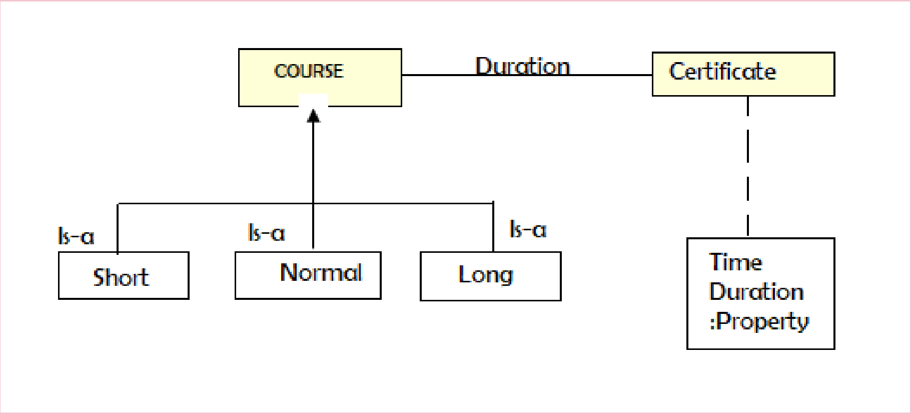


Figure 7.1: Example of a Fuzzy- ontology

A fuzzy concept represented as c_f is a concept which possesses, at least, one fuzzy *property* represented by p_f. A set of fuzzy *concepts* , C_F is defined as in equation 7.4.

$$C_F = \left\{ c_f / c_f = \left(N_{cf}, P, P_F \right) \right\} \tag{7.4}$$

Where:

- N_{cf} is a name of fuzzy *concept*, e.g., "short term course", or "long term course",
- P is a set of crisp *properties*, and
- P_F is a set of fuzzy properties, which is non empty.

A fuzzy *property* represented as p_f is a *property* which is represented in the form of a fuzzy linguistic variable. A set of fuzzy properties is represented as P_F and is defined as in equation 7.5

$$P_F = \{ p_f / p_f = (N_{pf}, T, M \} \tag{7.5}$$

Where:

- N_{pf} is a name of fuzzy *property*, such as duration

- T is a set of terms which are the values of the fuzzy *property*, e.g., T = {short, normal, long}.
- M is a mapping rule which map every term of T to a fuzzy set.

An example of fuzzy *concept* and fuzzy *property* is shown in figure 7.2.

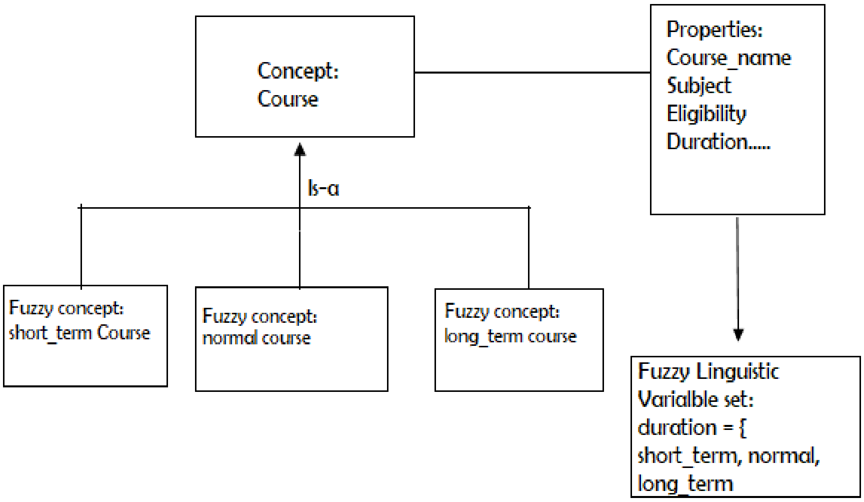


Figure 7.2: Example of a Fuzzy concepts and related properties

In figure 7.2, a *concept* named Course is represented which possesses three fuzzy *concepts* such as 'short term course', 'normal course' and 'long term course'. There are different *properties* for the *concept* named Course such as Course name, Subject, Duration etc. Here the fuzzy *concept* maps to the duration of the course and hence the property named Duration is associated with a fuzzy linguistic variable set with values short_term, normal and long.

The following section denotes the *concept* of fuzzy binary semantic relation and fuzzy binary association used in this study to specify the *relationship* and *association* of *concepts* and fuzzy linguistic variables.

7.3.2. Fuzzy Binary Semantic Relation and Association in Fuzzy Linguistic variable

A fuzzy binary semantic relation denoted by r_f is a relation which is represented in the form of a fuzzy linguistic variable. A set of fuzzy binary semantic relations R_F is defined as in equation 7.6.

$$R_F = \{r_f / r_f = (N_{rf}, T, M)\} \qquad (7.6)$$

Where:

- N_{rf} is the name of fuzzy relation e.g., “Equivalence”.
- T is a set of terms which are the values of the fuzzy relation, e.g., T= {weak equivalence, middle equivalence, strong equivalence}.

- M is a mapping rule which maps every term of T to a fuzzy set.

A fuzzy binary *association* represented by as_f is a relation which can be represented in the form of a fuzzy linguistic variable. A set of fuzzy binary *semantic relations* AS_F is defined with equation 7.7:

$$AS_F = \{as_f / as_f = (N_{asf}, T, M)\} \tag{7.7}$$

Where:

N_{asf} is the name of a fuzzy relation e.g., “duration”.

T is a set of terms which are the values of the fuzzy relation,

e.g., T = {short, normal, long}.

M is a mapping rule which map every term of T to a fuzzy set.

Figure 7.3 shows an example of fuzzy association generated from the relation duration

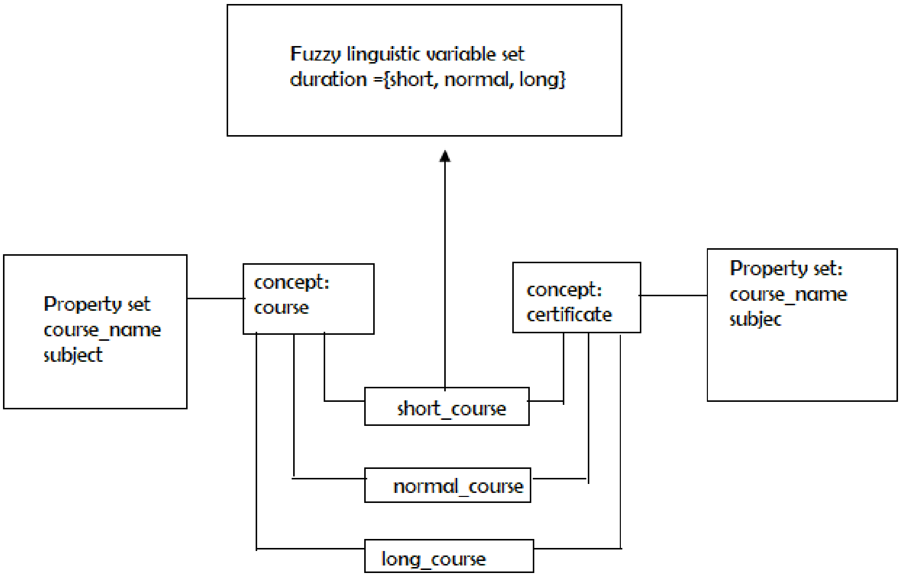


Figure 7.3: A sample Fuzzy association

7.4 Information Retrieval based on Fuzzy Ontology

Ontology tools are generally based on crisp logic and do not provide well-defined means for expressing fuzziness. The goal of fuzzy ontology is to incorporate uncertainties and imperfections. Fuzzy ontology has been introduced by Bobilloa *et.al.*, to represent knowledge in all domains in which the concepts to be represented have imprecise definitions [205]. Fuzzy *concepts* and *roles* are considered as fuzzy sets.

Information retrieval system based on Fuzzy-Ontology typically measures the relevance of documents to users' query based on the meaning of dominant words in each document.

Surface form representation of query terms which match query terms with document terms does not sufficiently retrieve documents that are relevant to the user's query. For instance, words like 'charge' can occur in the context of fee and the person in charge; close (of door) and close meaning near; etc. are used frequently in natural language queries. Words in these forms normally cause the retrieval of irrelevant documents as response to a user's query, if surface form is applied.

Fuzzy-ontology allows an easy determination of the precise meaning of a word as it relates to a document collection. The proposed system helps to directly integrate fuzzy logic in ontology in order to obtain an extension of the ontology that is more suitable for solving uncertainty reasoning problems. It is a first step towards the realization of a theoretical model of a complete framework based on ontologies for the University domain that are able to deal with the nuances of natural languages.

Since this study aims to be as general as possible, both the possibilities are considered: define a precise value or a linguistic one. In the former case, the expert, while creating the ontology, defines a function

$$f\colon ((\mathit{Concepts} \cup \mathit{Instance}) \times \mathit{Properties}) \rightarrow \mathit{Property\ Value} \times [0, 1] \tag{7.8}$$

With the meaning that *f(o, p)* is the value that a concept or an instance *o* assumes for property *p* with associated degree.

The following section describes the proposed methods and its implementation for generating fuzzy ontology to retrieve the required information based on a user query.

7.4.1 Generating a Fuzzy Ontology for Information Retrieval

A fuzzy ontology is an ontology extended with fuzzy values which are assigned through the two functions given below.

$$g : (Concepts \cup Instances) \times (Properties \cup Prop\ val) \rightarrow [0, 1] \quad (7.9)$$

and

$$h : Concepts \cup Instances \rightarrow [0, 1] \quad (7.10)$$

Hanene GHORBEL et al., proposed the Fuzzy Protégé as an extension of ontology editor Protégé by defining new meta classes to define the parameterized membership functions [202]. This framework uses parameterized membership functions to allow automatic attribution of membership degree and possible to perform inconsistency checking. They defined a new meta class named “Fuzzy_Class" in the Protégé meta classes hierarchy as shown in the figure 7.4.

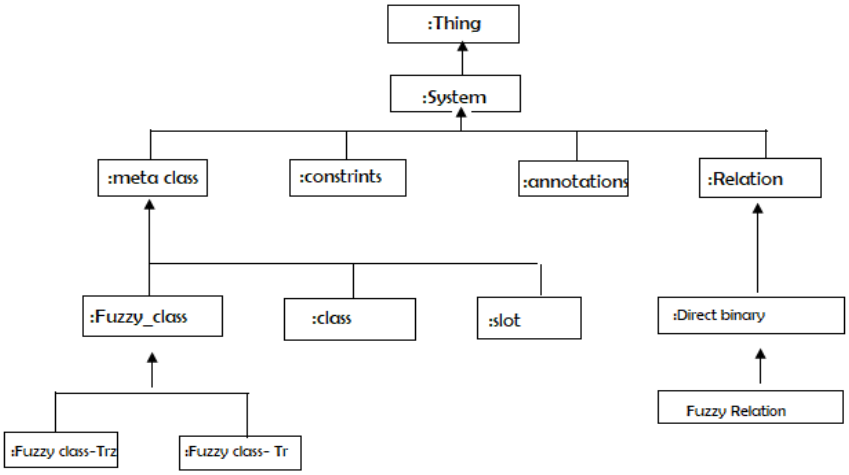


Figure 7.4: Fuzzy-Protégé Hierarchy

The construction of fuzzy ontologies passes through the same steps of constructing crisp ontologies. In a fuzzy ontology, each index term or object is related to every term (or object) in the ontology, with a degree of membership assigned to the relationship and based on fuzzy logic [206].

7.4.2 Extending Queries

When performing a query on a document, it is a usual practice to extend the set of concepts already present in the query with other concepts which can be derived from ontology. Typically, given a concept, also its parents and children can be added to the query and then searched in the document. A possible use of fuzzy ontology is to extend queries with, besides children and parents, instances of concepts which satisfy the query to a certain degree. For example if the given query is “Charge for revaluation of answer scripts”, it can be interpreted as the fee for revaluation or it can mention the person in the University who is responsible or in charge for dealing with the revaluation process of answer scripts.

Hence the impreciseness in the query is to be dealt with some contextual information to map to the fee section or the staff section accordingly. The following section describes the proposed method for fuzzifying the contextual information.

7.4.3 Fuzzifying Contextual Information

The aim of this component is to perform a contextualized fuzzification of the already obtained ontology. It consists of assigning membership values for each relation with the pivotal concept in order to favour the most relevant concepts.

7.4.4 Fuzzy Ontology Generation

To represent the fuzzy entities in the University ontology created as part of this study, the Fuzzy OWL2.1.1 plugin in Protégé tool is used. The plug in eases the fuzzy representation by allowing the specification of the type of fuzzy logic used, definition of fuzzy data types, fuzzy modified concepts, weighted concepts, weighted sum concepts, fuzzy nominals, fuzzy modifiers, fuzzy modified roles and data types, and fuzzy axioms [206]. In Fuzzy OWL 2, three main alphabets of symbols are assumed: *concepts* (fuzzy sets of individuals), roles, and individuals. These are represented in ontology as classes, relations, and individuals, respectively. For example, the University Ontology O_{UNIV} = (C,A^C,R,X) is a fuzzy ontology where its components are considered as follows:

C = {“Document”, “University_Section”}

A^C(“Document”) = {”Name”, “Title”, “Keywords”, “Abstract”,
“Body”, “Official”, “ Date”}

A^C(“University_Section”) = {“Name”, “Section”, “Keyword”}

R_N = {belong-to(“Document”, “University_Section”), consist-of(“University_Section”, “Document”)}

R_T = { superclassof(”University_Section,”
“Univeristy_Section”), subclassof(“University_Section”,
“University_Section”)}

X = {Implies(Antecedent(consist-of(I-variable(x1)
I-variable(x2)))
Consequent(belong-to(I-variable(x2) I-variable(x1))))
Implies(Antecedent(belong-to(I-variable(x1) I-variable(x2)))
Consequent(consist-of(I-variable(x2) I-variable(x1))))

```
Implies(Antecedent(superclass(I-variable(x1) I-variable(x2)))
          Consequent(subclass(I-variable(x2) I-variable(x1))))

Implies(Antecedent(subclass(I-variable(x1) I-variable(x2)))

          Consequent(superclass(I-variable(x2) I-variable(x1))))}
```

Figure 7.5 shows an overview of the Fuzzy Ontology Generation Process as proposed by Q.T Tho et.al,. which is implemented as part of this study for experimental purpose [204].

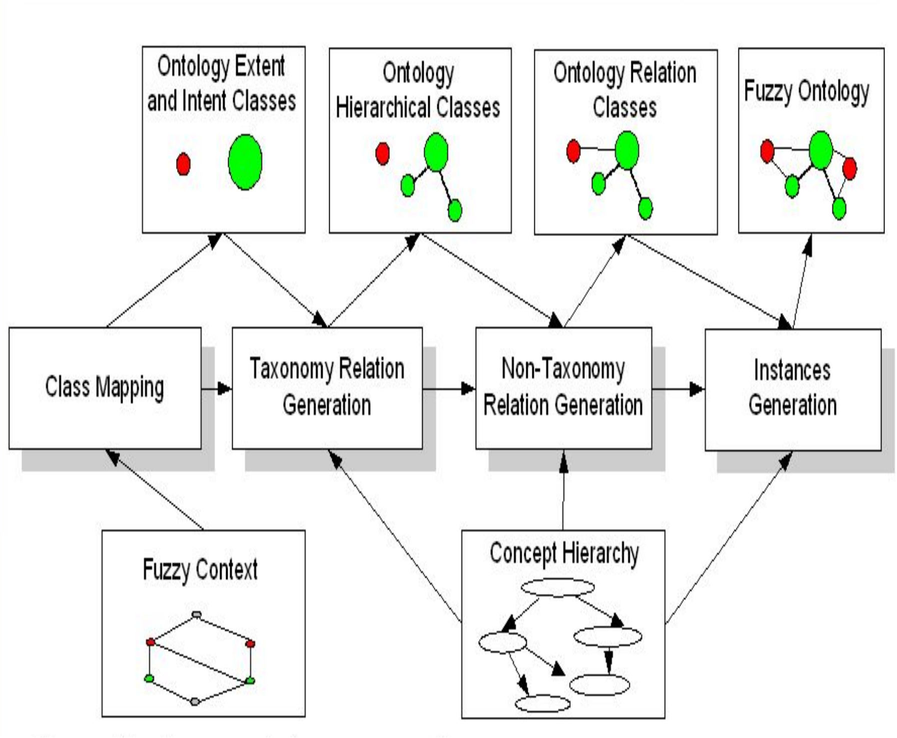

Ontology Extent and Intent Classes
Ontology Hierarchical Classes
Ontology Relation Classes
Fuzzy Ontology
Class Mapping
Taxonomy Relation Generation
Non-Taxonomy Relation Generation
Instances Generation
Fuzzy Context
Concept Hierarchy

Figure 7.5: Fuzzy Ontology Generation Process

The various steps involved in the generation of fuzzy ontology are class mapping, generation of taxonomy relation and non-taxonomy relation and finally the generation of instances. The following sub section describes each of these steps in detail.

a. ***Class Mapping***:

Class Mapping furnishes *C={E,I}* in which *E* and *I* are classes corresponding to extent and intent of the fuzzy context. For example, the extent class mapped from the extent of the fuzzy context given in Figure 7.5 can be labelled manually as a document. Appropriate names can be used to represent keyword attributes and use them to label the intent class names as well. For example, the class *'Research Area'* can be used to label the initial intent class.

b. ***Taxonomy Relation Generation***

The Taxonomy Relation Generation is represented as

$$R_T = \{Superclass(I, I), Subclass(I, I)\} \qquad (7.11)$$

Thus, the hierarchical relations between instances of intent classes are defined. Also, two rules are added to X accordingly:

$$Superclass(X, Y): -Subclass(Y, X) \qquad (7.12)$$

$$Subclass(X, Y): -Superclass(Y, X) \qquad (7.13)$$

For example, the class *Examination* can be expanded into a hierarchy of classes, in which each class represents a research area, corresponding to the *concept* hierarchy

c. ***Non-taxonomy Relation Generation***

Non-taxonomy Relation Generation is represented as

$$R_N = \{R_{IE}(I, E), R_{EI}(E, I) \qquad (7.14)$$

R_{EI} is the relation between the extent class and intent class. R_{IE} is the reversed relation of R_{EI}. However, it is necessary to label the nontaxonomy relation. For example, the relation between class Document and class Research Area can be labelled as belong-to, which implies that a document can belong to one or more research areas. Also, two rules are added to X accordingly:

$$R_{EI}(X, Y): -R_{IE}(Y, X) \qquad (7.15)$$
$$R_{IE}(X, Y): -R_{EI}(Y, X) \qquad (7.16)$$

d. ***Instances Generation***

Instances Generation generates the instances set $I = \{I_I, I_E,$ where I_I and I_E are instances of the intent and extent class. Then, it furnishes membership values for the instances' *attributes* and relationships. For example, each instance of the class *Document* corresponding to an actual document will be associated with the appropriate research areas. It is performed using the following rules:

Rule 1: Intent Instance Generation.

$\forall C \in H,$ add a corresponding instance, denoted as $I_I(C) to\ I_I$. The attributes values of $I_I(C)$ are defined as

$$A^C(\ I_I(C)) = \Phi(C) \qquad (7.17)$$

Rule 2: Extent Instance Generation.

$\forall O \in G,$ add an corresponding instance, denoted as $I_E(O) to\ I_I$. The attributes values $I_E(O)$ correspond to attributes values of O defined in *K*.

Rule 3: Membership Value Generation for Instance Taxonomy Relations.

If $C_1 < C_2, \ \forall C_1, C_2 \in H,$ the membership values of the taxonomy relations

$Subclass(I_I(C_1), I_I(C_2)$ and $Superclass(I_I(C_2), I_I(C_1)$ are defined as

$$\varphi(superclass(I_I(C_2), I_I(C_1)) = \varphi(subclass(I_I(C_1), I_I(C_1)\ ))$$
$$= subsethood(\Phi(C_1), \Phi(C_2) \qquad (7.18)$$

Rule 4: Membership value Generation for Instance Nontaxonomy Relations.

$\forall O \in G, C \in H, O \in P_o(C)$

the membership values of the taxonomy relations ,

$R_{IE}(I_I(C), I_E(O)) and\ R_{EI}(I_E(O), I_I(H))$ are defined as

$$\varphi(R_{IE}(I_I(C), I_E(O))) = \varphi(R_{IE}(I_I(C), I_E(O))) = \mu_o(C) \qquad (7.19)$$

7.5 Experimental Setup and Performance Evaluation

The experiments are carried out for hybrid ontology using fuzzy logic and ontology representation to retrieve the relevant information, and ranking the documents. The dataset has been built using preprocessed documents and queries in the University domain which are described in Chapter 3

7.5.1 Fuzzy Ontology Querying

Fuzzy Protégé contains a module for fuzzy ontology querying. When a user asks a query about the instances of a given fuzzy concept, the system has to return those instances that have a membership degree to the corresponding concept greater or equal to a given threshold. In order to do that, this work proposed to realize a pre-treatment on the instances of fuzzy concepts before query processing.

The goal of this study is to verify the performance of the fuzzy ontology and to compare it with the crisp ontology in the University domain. An evaluation parameter namely *scalability* is defined for this purpose [196]. The *scalability* is understood as the capability of the ontology to perform with a rule set and a reasoner to achieve domain recognition, in reasonable execution time for the developed data set.

Table 7.2 demonstrates the retrieval accuracy and execution time obtained for the proposed strategy. The five sub domains of the University such as academics, administration, examination, finance and planning & development are considered for generating queries to conduct the experiments with respect to the given document dataset. The number of queries and data set considered are the same as that of evaluating crisp ontology for better comparison. A total of 50 queries are fired against the same domain of documents and the precision and recall are noted.

Table 7.2: Retrieval accuracy and execution time obtained for the fuzzy ontology

Domain	Retrieval accuracy (%)	Execution time (seconds)
Academics	87.6	18
Administration	88	14
Examination	90.26	21
Finance	93.54	25
Planning & Development	85.46	16

The same set of 50 queries and corresponding document set are used for testing both the fuzzy and crisp ontologies developed for University domain, which make the comparison easy. Average precision and recall for the five sub domains of the University are given in table 7.3.

Table 7.3: Precision and recall obtained for crisp and fuzzy ontology generated the for University domain

Category Name	Crisp Ontology			Fuzzy Ontology		
	Precision (%)	Recall (%)	Time (ms)	Precision (%)	Recall (%)	Time (ms)
Academics	95.25	93.0	200	97.65	94.12	203
Administration	98.1	97.84	194	98.28	98.64	196
Examination	94.38	93.67	195	97.1	95.87	199
Finance	93.47	92.4	180	96.84	95.0	186
Planning and Development	99.24	99	150	99.49	99.13	154

From the table 7.3, it is evident that the precision and recall values are improved for fuzzy ontology when compared with crisp ontology. The results show that more fuzzy terms are present in the sub domains including Finance, Examination and Academics hence there is increment in precision and recall for fuzzy ontology when compared with crisp ontology. A graphical representation of the average precision and recall values obtained for crisp and fuzzy ontology are shown in figure 7.6 and 7.7 respectively. From the experiment results it is evident that the proposed Fuzzy Ontology method provides high exactness when compared with crisp ontology used for querying in the University domain

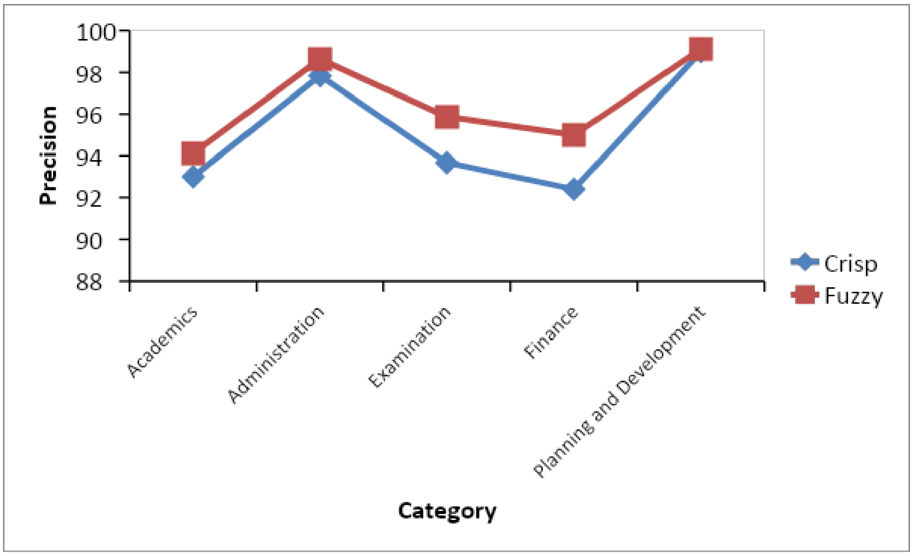


Figure 7.6: Precision for Crisp and Fuzzy Ontology querying obtained for the University domain

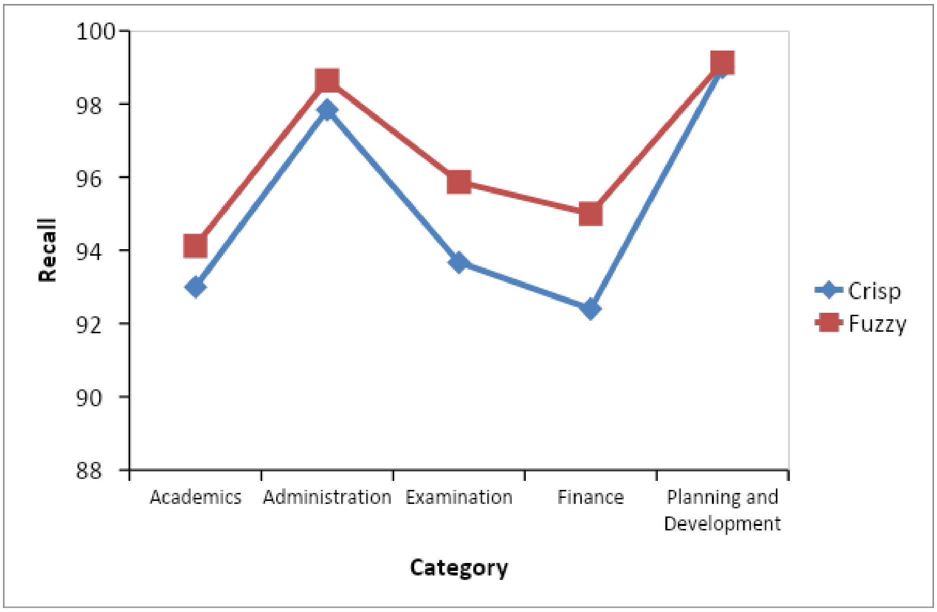


Figure 7.7: Recall for Crisp and Fuzzy Ontology querying obtained for the University domain

7.6 Conclusion

In this chapter, a fuzzy ontology framework is proposed for the query answering. Fuzzy ontology generation is performed by implementing Fuzzy Concept Analysis, Fuzzy Conceptual Clustering, Fuzzy Ontology Generation, and Semantic Representation Conversion. The fuzzy ontology is generated incrementally by furnishing new instances. The proposed framework would be useful to construct ontology from uncertainty data as it can represent uncertainty information and construct a concept hierarchy from the uncertainty information. The benefits of fuzzy ontology to crisp ontology are evaluated with scalability.

The experiments are conducted to verify the expressiveness and practical implications of fuzzy ontologies with crisp approaches. The precision and recall values of crisp and fuzzy systems are compared with the same data set for a comparative study. Though the execution time taken for a fuzzy ontology based system is more with an average of 2.12%, precision is increased with 0.18% to 3.61% and recall is enhanced with 0.13% to 2.81%. Hence the proposed Fuzzy ontology based query processing system can be effectively used for the reliable implementation of domain specific information retrieval systems on semantic web.

Chapter 8

Conclusions and Future Directions

8.1 Conclusions

The modern era has perceived a radical revolution in the growth of web-based information. Lots of researchers from various parts of the globe are engaged in research related to efficient information extraction and retrieval. One of the most exciting issues in today's world of digital information is to retrieve more relevant information from a pool based on a user query, integrating domain knowledge. This type of research becomes more challenging when the user requirements are given in the form of natural language queries or statements and the database consists of unstructured documents.

This thesis primarily addresses the problem of retrieving information from a specific domain by incorporating the semantic information provided by the user at the time of specifying a query in a natural way as simple as a natural language sentence. Though different information retrieval systems exist, there are only a few which incorporate semantic knowledge through ontology and Fuzzy logic. Hence this work is unique in its approaches and the domain of interest which comprises all the important branches of a typical state University in India.

The critical challenge faced in this work was to experiment with the keyword-based retrieval to concept-based retrieval of information by utilizing Ontologies. It was also crucial to identify the concepts in information present in the document dataset and the natural language queries given by users.

The domain considered is a University domain and the enquiries on various services offered by the University. The queries related to a University system in general are taken into consideration. The experiments are conducted based on the domain specific Query set and corresponding Document set. Both Query and Document set is subdivided into five separate categories such as Academics, Administration, Examination, Finance, and Planning & Development. The information is provided on user enquiries based on the documents available in the University repository. An overview of the query data acquisition and database creation is also presented. The different query and document pre-processing techniques used in the study are also detailed with experimental evidence. A query data set consisting of 7500 queries and a document data set consisting of 2800 University related documents are acquired for this study.

Efforts are also made to measure the similarity of documents and queries to improve the results of information retrieval. When a natural language query is fired for retrieving the information, the best similar document has to be retrieved. Since machines cannot work with the text as it is, texts are represented as feature vectors which are a set of independent units, which acts as a building block of the feature space. Usually, the text is represented by the assigned values such as binary, Term Frequency (TF), or Term Frequency-Inverse Document Frequency (TF-IDF) features. In this case, both the documents and queries are represented as vectors, and their similarity is measured to find the best match for a given query. Cosine and Jaccard coefficients are computed for this purpose. From the experimental results it is observed that the accuracy obtained for the similarity measures are 91.6 for Cosine and 88.4 for Jaccard respectively.

To incorporate semantic information or contextual information, word embedding is used. The GloVe is used to exploit the overall co-occurrence statistics of the words and thus topic modelling. It is a method to find the optimum number of iterations to the data set to identify the topic. From the experimental results the optimum value of iteration is obtained as 50 for the data set considered for this study. Then to deal with the vagueness a hybridized model of fuzzy with cosine and Jaccard is proposed. To analyse the proposed method, average precision and recall values are computed on randomly selected queries and on top 10, 25 and 50 documents based on Cosine, Jaccard and the proposed hybridized method named Fuzzy-Jaccard-Cosine Similarity measure (FJCSM). The result analysis of the experiment shows that the proposed method improves the Precision by more than 25% and Recall by more than 40%.

Initially, the natural language query which is expressed in the spoken language (English) is converted into a corresponding representation in SPARQL (SPARQL Protocol And Query Language) which has to be fired onto the Ontology developed in the University domain.

Ontology development is considered as the most fundamental step to be performed as part of the study and it has been illustrated with a prototype developed in the University domain. This proposed information retrieval system architecture is meant for overcoming the limitations of keyword-based searching. The proposed system extracts relevant information instead of giving a list of all the relevant documents containing related information. As the University Domain is subdivided into five sub domains, the Precision and Recall values are computed for a single domain in which query was fired against documents belonging to the same domain and also for the mixed domain in which query was fired against documents belonging to multiple domains. But mixed domain shows a decrease in performance due to the lack of some critical information. Hence Fuzzy Logic is incorporated with ontology to deal with the vagueness and impreciseness.

An attempt is also made to design and implement a promising framework for fuzzy ontology, in which concepts are represented with the help of fuzzy logic and reasoning. The experiments are carried out by implementing a hybridized ontology using fuzzy logic and RDF representation to retrieve the relevant information and ranking the documents. The proposed framework would be highly beneficial to construct ontology from uncertainty data as it can represent uncertainty information and construct a concept hierarchy from the uncertainty information automatically.

The benefits of Fuzzy ontology to Crisp ontology are evaluated with two parameters. The *scalability* is understood as the capability of the ontology to perform with a rule set, and a *reasoner* to achieve document retrieval. In reasonable execution time for querying and retrieving from the developed data set is considered. The experiments are conducted to verify the expressiveness and practical implications of fuzzy ontologies with crisp approaches. It is noticed that the execution time does not vary significantly for fuzzy queries compared to crisp queries.

8.2 Contributions

The considerable contributions to the field of semantic information retrieval as well as ontology development that has been reported as part of this research work are detailed below.

The query datasets containing the possible questions collected from the university enquiry section, authorities and staff, students and other stakeholders are framed in natural language and digitized into text files. This dataset is named Short Queries for University Services in English (SQUSE) and made available publically for research purposes.

The fuzzy *weights* are determined on the basis of the individual similarity scores of different similarity measures including Cosine and Jaccard similarity measures. To achieve improved results, the development of a hybridized similarity measure is proposed for finding appropriate weights to be used for each similarity measure. The Fuzzy logic is employed in the proposed approach to develop hybridized similarity measures. The mathematical formulation for a proposed Fuzzy-Jaccard-Cosine Similarity measure (FJCSM) is expressed in this study. It is evident from the experimental results that the proposed method increases average precision and an average recall of the IR system for dealing with document retrieval.

The most crucial step involved in this study is the development of Ontology for the University domain. The system proposed can overcome the limitations of keyword-based searching. Instead of providing a list of all the documents having related information, it can excerpt the pertinent information. The proposed system retrieves answers to queries related to a specific domain of interest. So this work can be considered as a model for the development and deployment of large and complex ontology.

To deal with the impreciseness in the user queries and to improve the retrieval of relevant information, a fuzzy ontology is developed. A comparison of the fuzzy ontology with crisp ontology for the University domain is carried out in terms of precision and recall. For the fuzzy system, precision is increased with 0.18% to 3.61% and recall is increased with 0.13% to 2.81% when compared with crisp systems even though the execution time is more for fuzzy ontology based systems with an average of 2.12%.

8.3 Performance Comparison with Existing Systems

The recent research findings reported for query document similarity and ontology based information retrieval along with the proposed methods derived out of this study are given in table 8.1 and 8.2 respectively for comparison.

The system proposed for document and query similarity is tested with the CISI dataset for information retrieval which is a publicly available dataset from the University of Glasgow's Information retrieval group.

Table 8.1: Comparison of the proposed method for document and query similarity with recent works reported

Sl.No	**Authors**	**Method**	**Precision**	**Recall**
1	Robertson SE *et.al.,* [207]	Okapi-BM25	0.0918	0.1178
2	Pathak-Gordon-Fan *et al.* [154]	GA-based method	0.0911	0.1145
3	Y.Gupta *et.al.,* [152]	FBHSM1	0.1036	0.1407
4	Y.Gupta *et.al.,* [152]	FBHSM2	0.1393	0.1772
5	Proposed System	FJCSM	0.1456	0.1782

There are different ontologies developed related to the University domain. But they have only considered the sub domains such as examination, courses *etc.* In this study, the entire University system is considered under the five sub domains with all their related concerns with the help of concepts, their object and data properties. The Table 8.2 gives a summary of the comparative study of the proposed system with existing ones related to University domain. The methods used for documents and query similarity of related works are also compared. The performance measure obtained from the experimental results is evident to prove that the proposed method is promising for the University domain.

Table 8.2: Comparison of the proposed ontology based Information Retrieval system with existing ones

Sl No	Reference	Year	Tool used	Use of Fuzzy Logic	Domain Considered						Evaluation Query	
					Academic	Administration	Examination	Finance	Planning & Development	Internship Assignment	DL Query	SPARQL
1	Ling Zeng *et.al.,*[208]	2009	TM4L		√							
2	A Ameen *et al.,*[209]	2012	Protégé		√	√	√			√		
3	Lijun Tang *et al.,* [210]	2015	Protégé		√	√	√				√	√
4	Abir M 'Baya *et al.,* [63]	2016	Protégé		√	√				√		√
5	Proposed System	2019	Protégé	√	√	√	√	√	√		√	√

8.4 Future Direction

A hybridization of Ontology and Vector Space Model is proposed in the present study to improve the retrieval of relevant information. Fuzzy logic is incorporated to deal with the confusing concepts present in the document or query and the performance of the proposed system is compared with existing results. In this context, other soft computing techniques such as rough sets and their hybrid models can also be considered for improving the retrieval of relevant information as a part of future study.

Also, to address the semantic or contextual information and to create a semantic web, Ontology is the crucial factor. But ontology design is only a semi-automatic process since the concepts to be represented as classes in the hierarchy need human intervention. The Fuzzy terms also need to be specified by an expert having profound knowledge of the domain being considered. Only an expert in the domain can identify critical factors to be expressed and the properties to be mapped. Hence a fully automatic method for the generation of ontology could be explored with the creation of a corpus for the domain of interest.

References

[1] T. B. L. et.al, "The semantic web," Scientific American, May 2001.

[2] T. W. K. J. C. Xingbin Chen, "Ontology-Based Representations of User Activity and Flexible Space Information: Towards an Automated Space-Use Analysis in Buildings," Advances in Civil Engineering, vol. 2019, 2018.

[3] O.-B. R. o. U. A. and. [Online]. Available: https://www.hindawi.com/journals/ace/2019/3690419/.

[4] Anwar A. Alhenshiri, "“Web Information Retrieval and Search Engines Techniques”," ,Al- Satil journa, pp. 55-92., 2010.

[5] S. R. a. K. S. Jones., " “Relevance weighting of search terms"," Journal of the American Society for Information Science, vol. 27, p. 129–146, 1976.

[6] "A survey of concept based information retrieval Tools," [Online]. Available: https://www.mii.It/ADBIS/local2/haav.pdf.

[7] T. H. J. a. L. O. Berners-Lee, "The Semantic Web," 2001.

[8] R. Millman, "What Is the Semantic Web?," IT Pro , Dennis Publishing Ltd, 2017.

[9] "Semantic Web - Wikipedia.," [Online]. Available: https://en.wikipedia.org/wiki/Semantic_Web.

[10] S. A. H. Mohammad Aman Ullah, "Ontology-Based Information Retrieval System for University: Methods and Reasoning: Proceedings of IEMIS," vol. 3, 2018.

[11] P. N. e. a. RIVINDU PERERA∗, "Semantic Web Today: From Oil Rigs to Panama Papers," rivinduPerera.com/publications, 2017.

[12] T. R. Gruber, "A Translation Approach to Portable Ontology specifications," Knowledge Acquisition,, vol. 5, no. 2, pp. 199-220, 1993.

[13] "What are Ontologies and What are the Benefits," [Online]. Available: https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/.

[14] H. D, "Overview of the TREC-9 Web Track," in Text REtrieval Conference (TREC), 2000.

[15] T. B. Lee, "W3C Semantic Web Activity," World Wide Web Consortium (W3C), 2011.

[16] S. R. a. K. S. Jones., ""Relevance weighting of search terms," Journal of the American Society for Information Science, vol. 27, p. 129–146., 1976.

[17] A. G. M. J.-R. E. V.-G. G. &. I. Soylu, " Experiencing OptiqueVQS: a multiparadigm and ontology-based visual query system for end users.," Universal Access in the Information Society, vol. 15, no. 1, p. 129, 2016.

[18] B. H. A. &. T. Muller, " Life Science Ontologies in Literature Retrieval: A Comparison of Linked Data Sets for Use in Semantic Search on a Heterogeneous Corpus," EKAW (Satellite Events), no. November, pp. 158-161, 2016.

[19] O. l. S. (OLS), "https://www.ebi.ac.uk/ols/ontologies/eupath".

[20] M. &. A. J. Bansal, " A Novel OBIRS System for Ontology Based Information Retrieval System," 2016.

[21] J. M.-S. J. N. G. R. F. S. A. S. Peter Bourgonje, "Towards a

platform for curation technologies: enriching text collections with a semantic-web layer," in The Semantic Web. ESWC 2016. Lecture Notes in Computer Science, vol 9989. Springer, Cham, 2016.

[22] S. &. A. M. Vigneshwari, "Social information retrieval based on semantic annotation and hashing upon the multiple ontologies," Indian Journal of Science and Technology, vol. 8, no. 2, pp. 103-107, 2015.

[23] P. University, " "About WordNet." WordNet..," Princeton University, 2010. .

[24] A. S. I. B. A. B. A.-M. C. Halaschek, "Semantic Web Technology Evaluation Ontology (SWETO): A test bed for evaluating tools and benchmarking applications," in Developers Day: Semantic Web Track, Intl WWW Conference, New York, 2004.

[25] N. W. C. L. M. R. K. &. L. W. Cao, " Privacy-preserving multi-keyword ranked search over encrypted cloud data," IEEE Transactions on parallel and distributed systems, vol. 25, no. 1, pp. 222-233, 2014.

[26] M. A. V.-G. R. G. F. &. S.-Z. J. J. Rodiguez-Garcia, "Ontologybased annotation and retrieval of services in the cloud," Knowledge-Based Systems, vol. 56, pp. 15-25, 2014.

[27] V. &. S. M. (. .. ,. 5. 6. Jain, "Ontology based information retrieval in semantic web: A survey," International Journal of Information Technology and Computer Science (IJITCS), vol. 5, no. 10, p. 62, 2013.

[28] R. G. R. S. R. &. C. A. Chauhan, "Domain ontology based semantic search for efficient information retrieval through automatic query expansion," Intelligent Systems and Signal Processing (ISSP), pp. 397-402, 2013.

[29] D. H. B. D. A. J. Kinjal Sheth, "Ontology Based Semantic Web Information Retrieval Enhancing Search

Significance," IJFRCSE, vol. 3, no. 8, p. 139 – 148 , 2017.

[30] G. &. J. V. Singh, " Information retrieval (IR) through semantic web (SW): An overview," arXiv preprint arXiv:1403.7162, 2014.

[31] Y. Yang, "A bag-of-objects retrieval model for web image search," in Proceedings of the 20th ACM international conference on Multimedia, 2014.

[32] L. S, "The Semantic Web And Its Languages," IEEE Intelligent Systems, 2013.

[33] E. Drymonas, "Unsupervised Ontology Acquisition from Plain," Springer, Vols. NLDB 2010, LNCS 6177, no. - Verlag Berlin Heidelberg 2, p. 277–287, 2010.

[34] "https://wiki.dbpedia.org," Wiki, 2009. [Online]. Available: https://wiki.dbpedia.org/about/dbpedia-community.

[35] M. Hepp, "GoodRelations: An Ontology for Describing Products and Services Offers on the Web," International Conference on Knowledge Engineering and Knowledge Management, Vols. LNCS, volume 5268, pp. 329-346, 2008.

[36] V. Lopez, "Aqualog: An Ontology-Driven Question Answering System for Organizational Semantic Intranets," Journal of Web Semantics, First look, pp. 1-34, 2007.

[37] L. S. a. B. Motik, "Ontology evolution within ontology editors," in Conference on the Evaluation of Ontology-based Tools, 2002.

[38] "http://protege.stanford.edu".

[39] "http://www.w3.org/2001/11/IsaViz/Overview.html".

[40] "http://apollo.open.ac.uk/index.html".

[41] B. P. E. S. B. C. G. J. A. H. Aditya Kalyanpur, "Swoop: A Web Ontology Editing Browser," J. Web Sem., vol. 4, no. 2, pp. 144-153, 2006r.

[42] "Web Ontology Language (OWL)," http://www.w3.org/2004/OWL, 2004.

[43] A. M. T. S. Karim, "Towards the Use of Ontologies for Improving User Interaction for People with Special Needs," Computers Helping People with Special Needs, Springer Berlin / Heidelberg, vol. 4061/2006, p. 77–84, 2006.

[44] R. ,. C. Bansal, "Semantic Web Tool:For Efficient retrieval of Links and Required Information," ,IJITEE, vol. 3, no. 4, 2013.

[45] M. N. M. S. S. Naveen, " Developing University Ontology using protégé OWL tool: process and reasoning," Int. J. Sci. Eng. Res, vol. 2, no. 9, 2011.

[46] J. a. B. A. Cullen, "The knowledge acquisition bottleneck: time for reassessment," Expert Systems, vol. 5, p. 216–225, 1988.

[47] A. a. S. Maedche, " Ontology learning for the semantic web," IEEE Intell. Syst, p. 16, 2001.

[48] C. S. V. S. S. e. J. G. d. L. G. R. Maddi, "Ontology Extraction from Text Documents by Singular Value Decomposition," in em ADMI 2001, 2001.

[49] D. S. e. A. Moreno, "Creating ontologies from Web documents," Recent Advances in Artificial Intelligence Research and Development, vol. 113, pp. 11-18, 2004..

[50] D. M. e. M. G. B. Fortuna, "Semi-automatic Construction of Topic Ontology," Lecture Notes in Computer Science, Springer, vol. 4289, pp. 121-131, 2005.

[51] N. Gantayat, "Automated Construction of Domain

Ontologies from Lecture Notes," Bombay, 2011.

[52] L. G. e. K. Ahmad, "Automatic Ontology Extraction from Unstructured Texts," Move to Meaningful Internet Systems CoopIS, DOA, and ODBASE, 2005.

[53] Cimiano,P. and Staab,S. (2005) Learning concept hierarchies from text with a guided agglomerative clustering algorithm, In: Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany

[54] IJntema,W., Sangers,J., Hogenboom,F. et al. (2012) A lexicosemantic pattern language for learning ontology instances from text. Web Semant., 15, 37–50.

[55] Bloehdorn,S. and Hotho,A. (2009) Ontologies for machine learning. In: Handbook on Ontologies. Springer, Berlin, Heidelberg, 637–661.

[56] Zavitsanos,E., Paliouras,G. and Vouros,G. (2008) A distributional approach to evaluating ontology learning methods using a gold standard. In: Third Ontology Learning and Population Workshop, ECAI, Patras, Greece.

[57] Zavitsanos,E., Petridis,S., Paliouras,G. et al. (2008) Determining automatically the size of learned ontologies. In: ECAI, IOS Press, Patras, Greece, 178, 775–776.

[58] Domingue, J. & Scott, P.: KMI Planet: Putting the Knowledge Back into Media. In M. Eisenstadt, and T. Vincent, (editors), The Knowledge Web: Learning and Collaborating on the Net, Kogan Press, (1998) 173-184

[59] D. H. K. Venkataraman, "Knowledge representation of university examination system ontology for semantic web," in 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017.

[60] L. Z. T. D. X. Zeng, "Study on construction of university course ontology: content, method and process," in International Conference 2009 Computational Intelligence and Software Engineering (CiSE), 2009.

[61] J. L. Y. J. J. Y. Y. Zhai, "Ontology-based information retrieval for university scientific research management," in 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), 2008.

[62] e. a. Tanuska, "The proposal of Ontology as a part of University Data warehouse," in 2010 IEEE 2nd International Conferences on Education Technology and Computer (ICETE), 2010.

[63] J. L. N. M. Y. O. A. M'Baya, " Ontology based system to guide internship assignment process," Signal-Image Technology & Internet-Based Systems (SITIS), p. 589–596 , 2016.

[64] K. K. R. R. B. Ameen, "Construction of university ontology," Information and Communication Technologies (WICT), 2012 World Congress, p. 39–44 , 2012.

[65] M. N. M. S. S. Naveen, " Developing University Ontology using protégé OWL tool: process and reasoning.," Int. J. Sci. Eng. Res., vol. 2, no. 9, 2011.

[66] L. E., "Enhanced text retrieval using natural language processing," Bulletin of the American Society for Information Science, vol. 24, pp. 14-16, 1998.

[67] S. W. Haas, "Natural language processing: toward large-scale robust systems," Annual Review of Information Science and Technology (ARIST, vol. 31, pp. 83-119, 1996.

[68] I. &. M. M. Mani, "Advances in automatic text summarization," Cambridge, MA: MIT Press, 1999.

[69] S. A.F, "Using NLP or NLP Resources for Information

Retrieval Tasks," Natural Language Information Retrieval, Kluwer Academic Publishers, pp. 99-111., 1999.

[70] A. J. Warner, "Natural language processing," Annual Review of Information Science and Technology (ARIST), vol. 22, pp. 79-108, 1987.

[71] R. &. K. R. Grishman, "Analysing language in restricted domains: sublanguage descriptions and processing.," London: Lawrence Erlbaum Associates, 1986.

[72] J. Leveling, "On the effect of stop word removal for SMS-Based FAQ retrieval," Natural Language Processing and Information Systems. Springer Berlin Heidelberg, pp. 128-139, 2012.

[73] D. Hogan, J. Leveling, H. Wang, P. Ferguson, and C. Gurrin. DCU@FIRE 2011: SMS-based FAQ retrieval. In FIRE 2011, 3rd Workshop of the Forum for Information Retrieval Evaluation, 2-4 December, IIT Bombay, pages 34–42, 2011

[74] E. Charniak, "Natural language learning," ACM Computing Surveys, vol. 27, pp. 3317-3319, 1995.

[75] N. A. L. L. Fuchun Peng, "Context Sensitive Stemming for Web Search," in SIGIR 2007 Proceedings, 2007.

[76] "Vector space model - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Vector_space_model.

[77] A. L. a. K. Y. Omid Shahmirzadi, "Text Similarity in Vector Space Models: A Comparative Study," September 2018. [Online]. Available: http://arxiv.org/abs/1810.00664v1.

[78] J. Ramos, "Using tf-idf to determine word relevance in document queries," 1999.

[79] K. C. G. C. a. J. D. Tomas Mikolov, " Efficient estimation of word representations in vector space," CoRR,

abs/1301.3781, 2013.

[80] Yang Junhui and Huang Chan, 2012, "Keywords Weights Improvement and Application of Information Extraction", Proc. of the 2011 2nd International Conference on CACS, Springer-Verlag Berlin Heidelberg, pp.95-100.

[81] Juan Ramos , Using TF-IDF to Determine Word Relevance in Document Queries ,2002, Lecture notes *Department* of *Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855.*

[82] Jinguo Sang1 , Shanchen Pang2*, Yang Zha3 and Fan Yang4, Design and analysis of a general vector space model for data classification in Internet of Things ,EURASIP Journal on Wireless Communications and Networking (2019) 2019:263

[83] SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrievalJuly 2008 Pages 435–442

[84] Q. V. L. a. T. Mikolov., "Distributed representations of sentences and documents," CoRR, abs/1405.4053, 2014.

[85] Jockers ML, Mimno D. Signifcant themes in 19th-century literature. Poetics. 2013;41(6):750–69. https://doi.org/10.1016/j.poetic.2013.08.005.

[86] Grimmer J. A Bayesian hierarchical topic model for political texts: measuring expressed agendas in senate press releases. Polit Anal. 2010;18(1):1–35. https://doi.org/10.1093/pan/mpp034

[87] Quinn KM, Monroe BL, Colaresi M, Crespin MH, Radev DR. How to analyze political attention. Am J Polit Sci. 2010;54(1):209–28. https://doi.org/10.1111/j.1540-5907.2009.00427.x

[88] Baum D. Recognising speakers from the topics they talk

about. Speech Commun. 2012;54(10):1132–42. https://doi. org/10.1016/j.specom.2012.06.003.

[89] DiMaggio P, Nag M, Blei D. Exploiting afnities between topic modeling and the sociological perspective on culture: application to newspaper coverage of U.S. government arts funding. Poetics. 2013;41(6):570–606. https://doi. org/10.1016/j.poetic.2013.08.004.

[90] Jockers ML, Mimno D. Signifcant themes in 19th-century literature. Poetics. 2013;41(6):750–69. https://doi. org/10.1016/j.poetic.2013.08.005.

[91] Ghosh D, Guha R. What are we "tweeting" about obesity? Mapping tweets with topic modeling and geographic information system. Cartogr Geogr Inform Sci. 2013;40(2):90–102.
https://doi.org/10.1080/15230406.2013.776210

[92] Evans MS. A computational approach to qualitative analysis in large textual datasets. PLoS ONE. 2014;9(2):1–11. https://doi.org/10.1371/journal.pone.0087908

[93] Guo L, Vargo CJ, Pan Z, Ding W, Ishwar P. Big social data analytics in journalism and mass communication. J Mass Commun Quart. 2016;93(2):332–59. https://doi.org/10.1177/1077699016639231

[94] Elgesem D, Feinerer I, Steskal L. Bloggers' responses to the Snowden afair: combining automated and manual methods in the analysis of news blogging. Computer Supported Cooperative Work: CSCW. Int J. 2016;25(2–3):167– 91. https://doi.org/10.1007/s10606-016-9251-z

[95] Maier D, Waldherr A, Miltner P, Wiedemann G, Niekler A, Keinert A, Adam S. Applying LDA topic modeling in communication research: toward a valid and reliable methodology. Commun Methods Meas. 2018;12(2–3):93–118. https://doi.org/10.1080/19312458.2018.1430754.

[96] J. L. a. T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," CoRR, abs/1607.05368, 2016.

[97] A. H. C. a. H. H. B. G. Marwa Naili, "Comparative study of word embedding methods in topic segmentation.," Procedia Computer Science, vol. 112, no. C, p. 340–349, 2017.

[98] Y. L. a. T. M. Sanjeev Arora, "A simple but tough-to-beat baseline for sentence embeddings," ICLR, 2017.

[99] P. G. a. M. J. Matteo Pagliardin, "Unsupervised learning of sentence embeddings using compositional n-gram features," CoRR, abs/1703.02507, 2017.

[100] C. O. a. Q. V. L. Andrew M. Dai, "Document embedding with paragraph vectors," CoRR, abs/1507.07998, 2015.

[101] J. E. Alvarez, ". A review of word embedding and document similarity algorithms applied to academic text," in bachelor thesis, university of freiburg, 2017.

[102] G. M. a. F. W. Pathak P, "Effective information retrieval using genetic algorithms based matching functions adaption," in Proceedings of 33rd Hawaii international conference on science (HICS), 2000.

103] A. S. A. K. S. Yogesh Gupta, "Fuzzy logic-based approach to develop hybrid similarity measure for efficient information retrieval," Journal of Information Science , vol. 40, no. 6, p. 846–857 , 2014.

[104] Y. Y. E. &. S. I. Y. Zhu, " A natural language interface to a graph-based bibliographic information retrieval system," Data & Knowledge Engineering, 2017.

[105] S. V. Mahboob Alam Khalid, " Passage Retrieval for Question Answering using Sliding Windows," in Proceedings of the 2 nd workshop on Information Retrieval for Question Answering (IR4QA),

Manchaster,UK, 2008.

[106] L.-W. K. K.-L. C. a. H.-I. C. I-chien Liu, "NTUBROWS System for NTCIR-7 IR for Question Answering," in Proceedings of NTCIR-7 Workshop meeting, Tokyo, Japan, 2008.

[107] G. Kothari, "SMS based interface for FAQ retrieval," in Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7, Singapore, August 2009.

[108] M. J. M. R. R. R. S. a. M. K. Anwar D. Shaikh, "Improving Accuracy of SMS Based FAQ Retrieval System," Multilingual Information Access in South Asian Languages. Springer Berlin Heidelberg, pp. 142-156, 2013.

[109] R. &. M. D. Mihalcea, "Automatic acquisition of sense tagged corpora," in Proceedings of the Twelfth International Florida AI Research Society Conference, Orlando, FL. Menlo Park, CA, 1999.

[110] J. Leveling, "Monolingual and Cross lingual SMS-based FAQ Retrieva," "DCU@ FIRE 2012, pp. 37-39, 2012.

[111] K.-S. C. Myoung-Cheol Kim, "A comparison of collocation-based similarity measures in query expansion," Information Processing and Management , vol. 35, pp. 19-30, 1999.

[112] D. e. a. Hogan, "DCU@ FIRE 2011: SMS -based FAQ retrieva," in 3rd Workshop of the Forum for Information Retrieval Evaluation, FIRE, 2011.

[113] Y.-F. K. Y.-H. K. e. M.-H. W. C.S. Lee, "Automated Ontology Construction for Unstructured Text Documents," Data & Knowledge Engineering, 2007.

[114] M. P. a. C.-F. Author links open overlay panelNataliaDíaz Rodrígueza, "A fuzzy ontology for semantic modelling

and recognition of human behaviour," Kowledgebased Sysetms, vol. 66, pp. 46-60, 2014.

[115] M. W. C. Lee, "A fuzzy expert system for diabetes decision support application," IEEE Trans. Syst. Man Cybern. Part B Cybern., , vol. 41, no. 1, pp. 139-153, 2011.

[116] M. P. G. D. M. R. F. T. e. a. Ali Harb, "Web Opinion Mining: How to extract opinions from blogs?.," in CSTST'08: International Conference on Soft Computing as Transdisciplinary Science and Technology, 2008.

[117] J. B. C. L. M. W. G. Park, "Ontology-based fuzzy-CBR support system for ship's collision avoidance," in Proceedings of the 6th International Conference on Machine Learning and Cybernetics, 2007, pp. 1845–1850..

[118] M. W. K. K. D. A. P. Alexopoulos, "IKARUS-Onto: a methodology to develop fuzzy ontologies from crisp ones," Knowl. Inf. Syst., vol. 32, no. 3, pp. 667-695, 2012.

[119] S.-S. M. Elmogy, "A fuzzy ontology modeling for case base knowledge in diabetes mellitus domain," Engineering Science and Technology,Elsevier, vol. 20, no. 3, pp. 1025-1040, 2017.

[120] R. S. D. L. Y. C. T.-H. a. H. J.-X. Lau, "Toward a fuzzy domain ontology extraction method for adaptive e-learning," Knowledge and Data Engineering, IEEE Transactions, vol. 21, no. 6, p. 800–813, 209.

[121] H. B. A. a. B. R. Ghorbel, "Fuzzy ontologies building method: Fuzzy ontomethodology," Fuzzy Information Processing Society (NAFIPS), Annual Meeting of the North American, p. 1–8., 2010.

[122] T. a. K. A. Kaur, "Extension of a crisp ontology to fuzzy ontology," International Journal Of Computational Engineering Research, vol. 2, 2012.

[123] H. a. B. G. H. Baazaoui-Zghal, "A fuzzy-ontology-driven

method for a personalized query reformulation," in FUZZ-IEEE 2014, IEEE International Conference on Fuzzy Systems., 2014.

[124] B. H. C. a. J. M. Chien, "Ontology-based information retrieval using fuzzy concept documentation," Cybernetics and Systems, vol. 41, no. 1, p. 4–1, 2010.

[125] F. M. Z. M. a. Y. L. Zhang, "Construction of fuzzy ontologies from fuzzy xml models," Knowl.-Based Syst, vol. 42, p. 20–39, 2013.

[126] H. B. N. N. T. a. N. P. K. Truong, "Fuzzy ontology building and integration for fuzzy inference systems in weather forecast domain," ACIIDS, vol. 1, p. 517–527, 2011.

[127] J. K. Gill, "Semantic Search Engine with Ontology and Machine learning," XENONSTACK, 2018.

[128] https://dcs.uoc.ac.in/UDDSE/#

[129] S. Alasadi, "Review of Data Preprocessing Techniques in Data Mining," Journal of Engineering and Applied Sciences , vol. 12, no. 6, pp. 4102-4107, 2017.

[130] "Preprocessing Techniques," 2015. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/preprocessing-technique.

[131] R. V. Loon, "Tokenizing Words and Sentences with NLTK," May 2015. [Online].

[132] "Removing-stop-words-nltk-python," [Online]. Available: https://www.geeksforgeeks.org › removing-stop-words-nltk-python.

[133] H. B. Jon Ezeiza Alvarez, "A review of word embedding and document similarity algorithms applied to academic text," University of Freiburg, 2017.

[134] J. &. M. J.H, "Speech and language processing: an

introduction to natural language processing, computational linguistics and speech recognition," Upper Saddle River, NJ: Prentice Hal, 2000.

[135] J. Leveling, "On the effect of stop word removal for SMS-Based FAQ retrieval," Natural Language Processing and Information Systems. Springer Berlin Heidelberg, pp. 128-139, 2012.

[136] R. KODAKARI, "WordNet2," [Online]. Available: https://snapcraft.io/wordnet2.

[137] "Categorizing and Tagging Words," [Online]. Available: https://www.nltk.org/book/ch05.html.

[138] https://dcs.uoc.ac.in/SQUSE/#

[139] A. Geitgey, "Natural Language Processing is Fun!," 2018.

[140] "Conversely in a database devoted to health or medicine," [Online]. Available: https://www.coursehero.com/file/p7rk8f0/Conversely-in-a-database-devoted-to-health-or-medicine-antibiotic-would/.

[141] R. Koenig, "NLP for Beginners: Cleaning & Preprocessing Text Data," [Online]. Available: https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data-ae8e306bef0f.

[142] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. New York: Cambridge University Press

[143] D. MUNTEANU, "Vector Space Model For Document Representation In Information Retrieval," The Annals Of "Dunarea De Jos" University Of Galati Fascicle III , 2007.

[144] [Online]. Available: http://www.infotoday.com/searcher/may01/liddy.htm.

[145] J. X. T. Q. W. X. a. H. L. T. Y. Liu, "LETOR: Benchmark

Dataset for Research on Learning to Rank for Information Retrieval on Re," in In Proceedings of the Learning to Rank workshop in the 30th annual International ACM SIGIR Conference (SIGIR'07), 2007.

[146] "Tf-IDF," [Online]. Available: https://en.wikipedia.org/wiki/Tf%E2%80%93idf# Inverse_document_frequency.

[147] Z. Harris, " Distributional structure," in Word, 10(23), 1954, p. 146–162.

[148] "Continuous Bag of Words (CBOW)," Words as Vectors, April 2015.

[149] K. C. C. D. Tomas Mikolov, "Efficient Estimation of Word Representations in," arXiv:1301.3781v3 , 2013.

[150] Dieu-ThuLe, "Queryclassificationusingtopicmodelsandsupportvectormachine".

[151] "https://docs.anaconda.com/," [Online]. Available: https://docs.anaconda.com/.

[152] A. S. A. S. Yogesh Gupta, "Development of hybrid similarity measure using fuzzy logic for performance improvement of nformation retrieval system," in International conference on computing for sustainable global development, India, 2014.

[153] A. S. A. S. Yogesh Gupta, "Fuzzy logic-based approach to develop hybrid similarity measure for efficient information retrieval," Journal of Information Science, vol. 40, no. 6, p. 846–857, 2014.

[154] G. M. a. F. W. Pathak P, " Effective information retrieval using genetic algorithms based matching functions adaption," in Proceedings of 33rd Hawaii international conference on science (HICS), 2000.

[155] J. K. Lotfi A. Zadeh, Fuzzy logic for the management of uncertainty, New York, NY, USA : John Wiley & Sons, Inc. , 1992.

[156] "Mamdani Fuzzy Model," [Online]. Available: http://researchhubs.com/post/engineering/fuzzy-system/mamdani-fuzzy-model.html.

[157] "fuzzy inference systems," [Online]. Available: https://mathworks.com.

[158] A. R. Akram Roshdi, "Information Retrieval: search process, techniques and strategies," International Journal of Computer Networks and Communications Security, vol. 3, no. 9, p. 373–377, 2015.

[159] F. a. K. U. Giunchiglia, "Concept Search," in The Semantic Web: Research and Applications, Berlin, Heidelberg, Springer Berlin Heidelberg, 2009, pp. 429--444.

[160] K. et.al, "Semantic Web Based Efficient Search," International Journal of Engineering and Technology (IJET), vol. Vol 5 , no. No 6 Dec, pp. 4914-4928, 2014.

[161] J. D. P. Y. Y. Yan, "Ontology-based intelligent information retrieval system," J, Software, vol. 26, no. 7, pp. 1675-1687, 2015.

[162] T. R. Gruber., " A Translation Approach to Portable Ontologies. Knowledge Acquisition," vol. 5, no. 2, p. 199–220, 1993.

[163] "https://www.w3.org," [Online]. Available: https://www.w3.org/OWL/.

[164] Reshma P K ,Lajish. V. L., "Ontology Based Semantic Information Retrieval Model for the Universty Domain," IJAER, vol. 13, no. 15, 2018.

[165] L. C. a. Jain, "Innovations in Knowledge Processing and Decision Making in Agent-Based Systems," in Knowledge

Processing and Decision Making in Agent-Based Systems, Springer-Verlag Berlin Heidelberg, 2009, pp. 1-12.

[166] "https://www.w3.org," 2015. [Online]. Available: https://www.w3.org/standards/semanticweb/.

[167] R. P. A. P. R.Swaminarayan, "Execution of SPARQL Query using Apache Jena Fuseki Server in AISHE," International Journal of Advance Engineering and Research Development, vol. Volume 4, no. Issue 9, pp. 387-395, 2017.

[168] J. L. Y. J. J. Y. Y. Zhai, " Ontology-based information retrieval for university scientific research management," 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), p. 1–4, 2008.

[169] "https://wordnet.princeton.edu," [Online]. Available: https://wordnet.princeton.edu/.

[170] R. B. C. F. D. G. a. K. M. e A. Miller, "Introduction to WordNet: An On-line Lexical Database Georg".

[171] M. S. a. N. P. Shashirekha H.L., "Ontology based similarity measure for Text Documents," Proceedings of International Conference on Signal & Image Processing, ICSIP 2009, vol. August , pp. 602-606, 2009.

[172] L. K. W. a. P. D. Stanley Loh, "Concept-based Knowledge discovery in Texts Extracted from the Web," In Proceedings of ACM SIGKDD Explorations, 2000.

[173] S. I. M. Rada F. Mihalcea, " Word Semantics for Information Retrieval: Moving One Step Closer to the Semantic Web".

[174] D. M. e. a. Vijayarajan V, " A generic framework for ontology-based information retrieval and image retrieval in web data," Human-centric Computing and Information Sciences, vol. 6, no. 1, p. 18, 2016.

[175] A. Sandhan, "Sandhan-Indian language search engine," April 2013. [Online]. Available: www.cse.iitb.ac.in/arjun/Sandhan -Intro.pptx.

[176] J. M. J.H., Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition, Upper Saddle River, NJ: Prentice Hall., 2000.

[177] P. P. Dhaval Patel, " A Review Paper on Different Approaches for Query Optimization using Schema Object base View," International Journal of Computer Applications, vol. Volume 114, no. No. 4, March 2015.

[178] J. L. ,. Yujun Yang, " The research of the fast SVM classifier method," in Wavelet Active Media Technology and Information Processing (ICCWAMTIP),12th International Computer Conference, 2015.

[179] F. v. H. a. F. G. Heiner Stuckenschmidt, "Query Processing in Ontology-Based Peer-to-peer Systems. Technical Report," AI Department VrijeUniversiteit, Amsterdam, November 2002.

[180] J. Rapoza, "SPARQL Will Make the Web Shine," eWeek, January 2007.

[181] T. Segaran, C. Evans and J. Taylor, "SPARQL," in Programming the Semantic Web, 1005 Gravenstein Highway North, Sebastopol, CA 95472, O'Reilly Media, Inc, p. 84.

[182] D. Beckett, "What does SPARQL stand for?," semantic-web@w3.org, .October 2011.

[183] A. R. G. Leila Zemmouchi-Ghomari, "Translating natural language competency questions into SPARQL queries: A case study,WEB," in The First International Conference on Building and Exploring Web Based Environments, 2013 .

[184] A. B. A. a. P. Zweigenbaum, "Medical Question

Answering: Translating Medical Questions into SPARQL Queries," in Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, Miami, Florida, USA, 2012.

[185] D. Damljanovic, "FREyA: an interactive way of querying linked data using natural language," Proceedings of the 8th international conference on The Semantic Web, no. May, p. 125–138, 2011.

[186] M. B. Marta Tatu, "Automatic Extraction of Actionable Knowledge," in IEEE Tenth International Conference on Semantic Computing, 2016.

[187] M. N. M. S. S. Naveen, "Developing University Ontology using protégé OWL tool: process and reasoning," Int. J. Sci. Eng. Res, vol. 2, no. 9, 2011.

[188] R. C. R. Bansal, " An approach for semantic informationretrieval from ontology in computer science domain," Int. J. Eng.Adv. Technol. (IJEAT), vol. 4 , no. 2, 2014.

[189] http://herontology.esi.dz/content/downloads, "herontology," 2012.

[190] Quepy, "https://www.python.org," [Online]. Available: https://pypi.python.org/pypi/quepy/.

[191] S. C. R Bansal, "Design and development of semanticweb-based system for computer sciencedomain-specific information retrieval," Perspectives in Science , vol. 8, p. 330—333, 2016.

[192] A. S. Garima Singh, "An algorithm to transform natural language into SQL queries for relational databases," Selforganizology, vol. 3, no. 3, pp. 100-116, 2016.

[193] M Tatu, S Werner, “Semantic question answering on big data”, SBD '16: Proceedings of the International Workshop on Semantic Big DataJune 2016 Article No.:

10 Pages 1–6https://doi.org/10.1145/2928294.2928302

[194] S. Ferré, " Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language.," Semantic Web, vol. 8, no. 3, p. 405–418, 2017.

[195] Mateus de Carvalho Coelho, Julio Cesar dos Reis," Learning to build SPARQL queries from natural language questions", https://lod-cloud.net (2019)

[196] C. D. Calegari S., "Towards a Fuzzy Ontology Definition and a Fuzzy Extension of an Ontology Editor," in Enterprise Information Systems. ICEIS 2006. Lecture Notes in Business Information Processing, vol 3. , Berlin, Heidelberg, Springer,, 2008, pp. 147-158.

[195] C. D. Calegari S., "Fuzzy Ontology, Fuzzy Description Logics and Fuzzy-OWL," in Applications of Fuzzy Sets Theory. WILF 2007. Lecture Notes in Computer Science, vol 4578., Berlin, Heidelberg, Springer,, 2007, pp. 118-126.

[196] R. Fullér, "What is fuzzy logic and fuzzy ontology?," in KnowMobile National Workshop, Helsinki, October 30, 2008.

[197] Y. Y. E. &. S. I. Y. Zhu, " A natural language interface to a graph-based bibliographic information retrieval system," Data & Knowledge Engineering, 2017.

[198] A. K. Tanumeet Kaur 1, "EXTENSION OF A CRISP ONTOLOGY TO FUZZY ONTOLOGY," International Journal Of Computational Engineering Research (ijceronline.com), vol. Vol. 2 , no. Issue. 6, pp. 201-207, 2012.

[199] L. A. Zadeh, "Fuzzy Sets," in Advances in Fuzzy Systems — Applications and Theory:, vol. Volume 6, 1996, pp. 19-30.

[200] A. B. B. Hanene GHORBEL Agrebi, "Fuzzy Ontologies Building Method: Fuzzy OntoMethodology," in

Conference: NAFIPS'10At: , Toronto, Canada, 2010.

[201] U. S. Fernando Bobilloa, "The fuzzy ontology reasoner fuzzyDL," Knowledge-Based Systems, vol. 95, pp. 12-34, 2016.

[202] A. B. R. B. Hanêne GHORBEL, "Fuzzy Protégé for Fuzzy Ontology Models," in IPC'09, 2009.

[203] D. Parry, " Evaluation of a fuzzy ontology-based medical information system," IJHISI, vol. 1, no. 1, p. 40–51, 2006.

[204] U. S. F. Bobillo, ",Fuzzy ontology representation using OWL 2,," Int. J.Approx. Reason, vol. 52, no. 7, p. 1073–1094., 2011.

[205] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic Fuzzy ontology generation for semantic Web," IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 6, pp. 842– 856, 2006.

[206] N. D. Rodriguez, "A fuzzy ontology for semantic modelling and recognition of human behaviour," ElseivierKnowledge - Based systems, vol. 66, pp. 46-6, 2014.

[207] W. S. a. B. M. Robertson SE, "Okapi-BM25 at TREC–7: Automatic ad hoc, filtering, VLC and filtering tracks.," in Proceedings of the seventh text retrieval conference (TREC-7), pp. 253–264., 1999.

[208] L. Z. T. D. X. Zeng, "Study on construction of university course ontology: content, method and process," International Conference 2009 Computational Intelligence and Software Engineering (CiSE), p. 1–4 , 2009.

[209] K. K. R. R. B. Ameen, "Construction of university ontology," in Information and Communication Technologies (WICT), 2012 World Congress, 2012.

[210] Lijun Tang1 and Xu Chen, Ontology-Based Semantic Retrieval for Education Management Systems, Journal of

Computing and Information Technology - CIT 23, 3, pp 255–267, 2015

List of Publications of the Author

1. **Reshma P.K**, Suharshala R, Lajish V L, "A Novel Document and Query Similarity Indexing using VSM for Unstructured Documents", *2020 6th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS),* Coimbatore, India, 2020, pp. 676-681, DOI: 10.1109/ICACCS48705.2020.9074255.

2. **Reshma P.K.**, Suharshala R, Lajish V L **,"Document and Query Similarity Indexing using VSM for Unstructured Documents and Natural Language Queries",** JAC : A Journal Of Composition Theory, Vol 12, Issue 12, Dec 2019 pp 1074-1080 ISSN : 0731-6755 **DOI:19.18001.AJCT .2019.V12I12.19.11615**

3. **Reshma.P.K.**, "Ontology Based Information Retrieval System – The State of Art", Frontiers of Research in Mathematical and Computing Sciences- UGC-HRDC, University of Calicut, Vol.2.Nov –Dec 2018. ISBN: 9-789387-398825

4. **Reshma.P.K** & Lajish V.L., "Ontology Based Semantic Information Retrieval Model For University Domain", International Journal of Applied Engineering Research, Vol 13, No. 15 2018, pp 12142-12145. ISSN: 0973-4562

5. **Reshma.P.K** & Lajish V.L., "An Application Model of Semantic Web Based Social Web Mining", International Journal of Engineering Research and Technology Conference Proceeding NSDMCC- 2015,Vol 4, Issue 06 pp 5-8 ISSN: 2278-0181 DOI: 10.17577/IJERTCON V3IS30004

6. **Reshma.P.K** , Lajish V.L & Madhu V T, "Semantic Web Framework for Improved e-Governance", International Journal of Engineering Research and Technology Conference Proceeding RTPPTDM- 2015, Vol 3, Issue 30, pp 92-95 ISSN : 2278-0181 DOI: 10.17577/IJERTCO NV4IS06017

7. **Reshma.P.K** & Lajish V.L , "Web Mining for Multimedia Data – A Soft Computing Framework", International Journal of Scientific & Engineering Research ,Vol 5, Issue 9, Sep. 2014, pp 27-32. ISSN: 2229-5518