

M.Sc. Engg. (CSE) Thesis

Knowledge Graph Augmented Document Concept Hierarchy
Generation by Extracting Semantic Tree

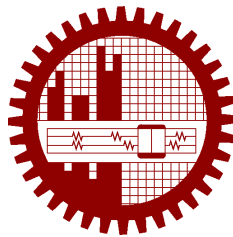
Submitted by

Sanjida Nasreen Tumpa

1015052073

Supervised by

Dr. Muhammad Masroor Ali



Submitted to

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh

in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Engineering

June 2019

Candidate's Declaration

I, do, hereby, certify that the work presented in this thesis, titled, "Knowledge Graph Augmented Document Concept Hierarchy Generation by Extracting Semantic Tree", is the outcome of the investigation and research carried out by me under the supervision of Dr. Muhammad Masroor Ali, Professor, Department of CSE, BUET.

I also declare that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

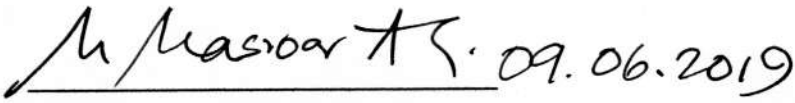
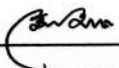
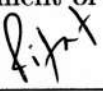

Sanjida Nasreen Tumpa.

Sanjida Nasreen Tumpa

1015052073

The thesis titled “Knowledge Graph Augmented Document Concept Hierarchy Generation by Extracting Semantic Tree”, submitted by Sanjida Nasreen Tumpa, Student ID 1015052073, Session October 2015, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents on June 9, 2019.

Board of Examiners

1.  09.06.2019
Dr. Muhammad Masroor Ali
Professor
Department of CSE, BUET, Dhaka
Chairman
(Supervisor)
2. 
Dr. Md. Mostofa Akbar
Professor and Head
Department of CSE, BUET, Dhaka
Member
(Ex-Officio)
3. M. M. Islam 09.06.19
Dr. Md. Monirul Islam
Professor
Department of CSE, BUET, Dhaka
Member
4. 
Dr. Rifat Shahriyar
Associate Professor
Department of CSE, BUET, Dhaka
Member
5. 
Dr. M. Kaykobad
Professor (in PRL)
Department of CSE
Bangladesh University of Engineering and Technology
Dhaka
Member
(External)

Acknowledgement

First of all, I would like to declare that all the appraisals belong to the Almighty ALLAH. I want to convey my heartfelt thanks to my supervisor Professor Dr. Muhammad Masroor Ali for introducing me to the fascinating and prospective field of Semantic web and Text mining. I have learned from him how to carry on research work, how to write, how to work hard, how to speak and present well. My heartiest gratitude goes to him for his patience steering to the right path, towards my goal. I again express my indebtedness, sincere gratitude, and profound respect to him for his constant supervision, suggestions, and wholehearted guidance throughout the progress of this work.

Besides my supervisor, I would like to thank my thesis committee: Professor Dr. Md. Mostofa Akbar, Professor Dr. Md. Monirul Islam, Associate Professor Dr. Rifat Shahriyar and Professor Dr. M. Kaykobad for their insightful comments and encouragement to widen my thesis from various perspectives.

I would acknowledge my heartfelt thanks to my parents, my husband and other family members for giving their best support throughout my years of study and through the arduous process of new findings during this research.

Finally, every honor and every victory on earth is due to ALLAH, who descended from Him, must be ascribed to Him. I deeply express my sincere thankfulness to the endless generosity of the Almighty ALLAH.

Dhaka
June 9, 2019

Sanjida Nasreen Tumpa
1015052073

Contents

Candidate’s Declaration	i
Board of Examiners	ii
Acknowledgement	iii
List of Figures	vi
List of Tables	vii
List of Algorithms	viii
Abstract	ix
1 Introduction	2
1.1 Problem Definition	2
1.2 Research Aim and Objective	3
1.3 Thesis Contribution	4
1.4 Thesis Organization	4
2 Related Work and Current Status	5
2.1 Document Content Representation	5
2.1.1 Standard Boolean Model	5
2.1.2 Vector Space Model	6
2.1.3 Graph Representation	6
2.2 Concept Hierarchy	9
2.3 Research Questions	9
3 Preliminaries	11
3.1 Basic Terminology	11
3.1.1 Semantic Web	11
3.1.2 Text Mining	13
3.1.3 Coreference Resolution	14
3.1.4 Some Knowledge Bases	14

3.1.5	Tree, Rooted Tree, Forest	17
4	Methodology	19
4.1	Document Content Tree Generation	19
4.1.1	Tokenization and Preparing Tagged Document	20
4.1.2	Applying Coreference Resolution	20
4.1.3	Labeling and Filtration of Extracted Information	21
4.1.4	Tree Construction	21
4.2	Concept Hierarchy Generation	25
4.2.1	Concept Extraction from Provided Document	25
4.2.2	Knowledge Extraction Using Knowledge Bases	26
4.2.3	Word Embedding and Similarity Checking	26
4.2.4	Concept Clustering and Hierarchy Generation	27
4.3	A Framework for Multilingual Ontology	27
4.3.1	Multilingual Ontology	28
4.3.2	Multilingual Ontology Mapping	29
4.3.3	Information Retrieval	31
4.4	Summary	32
5	Experimental Result and Analysis	33
5.1	Experimental Setup	33
5.2	Selection of Language	34
5.3	Experimental Data Set	34
5.4	Experiment Conduction	34
5.5	Result Analysis	35
5.5.1	Analysis on Document Content Tree	35
5.5.2	Analysis on Concept Hierarchy	36
5.6	Comparative Analysis	37
5.6.1	Knowledge Base Augmentation	37
5.6.2	Document Content Tree Generation	37
5.6.3	Semantic Rules	38
6	Conclusion	39
6.1	Contribution of the Work	39
6.2	Scope of Future Work	40
	References	41
A	Algorithms	47
A.1	Algorithm of Document Content Tree Generation	47

List of Figures

2.1	A semantic network generated from a single configuration of concept senses [1].	8
2.2	Graph constructed from the presented sample text, considering the window size is 2 [2].	8
3.1	Latest version of the Semantic Web Stack [3]	12
3.2	An example RDF model [4]	13
3.3	A Text mining framework [5]	14
3.4	An example of Coreference resolution	15
3.5	Logarithmic graph of the 20 largest language editions of Wikipedia [6] . . .	16
4.1	A snippet of some lines from a Bengali document	21
4.2	Content tree from the sample texts of Figure 4.1	23
4.3	Reduced content tree from the sample texts of Figure 4.1 after applying the semantic rules	24
4.4	A snippet of gathered information from knowledge bases related to “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]”.	26
4.5	Document content tree generated from the document on “সাকিব আল হাসান [English translation: Sakib Al-Hasan]” and “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]”.	28
4.6	Generation of combined ontology.	30
4.7	Linking dictionary to ontology	31

List of Tables

4.1	Processed tagged document for the first two lines of Figure 4.1	22
4.2	The number of instances for a set of selected classes within the canonicalized DBpedia data sets for each language.	29
5.1	Statistics of the corpus to evaluate the model and extracted tree nodes. . .	36

List of Algorithms

1	Algorithm for representing the document as tree.	48
---	--	----

Abstract

Semantic Web, as an extension of the traditional web, is concerned about the vast amount of unstructured data, and with its motive to make the entire knowledge content machine-readable, as well as machine-interpretable, all the processes of structuring the data are highly significant. Knowledge representation in trees has been a familiar mechanism for some time. However, such representations lack in existence when it comes to document content. This thesis properly presents a general mechanism that can generate a representation of the concepts of a document in the form of the knowledge tree. This rooted tree helps represent the contents of a document in an organized way as well as to find the core concepts of the document. We more considerably augment knowledge from various knowledge bases and analyze those data by mapping it with an existing ontology to obtain the taxonomy. We explain how this can be effective to create hierarchical concept recommendations of a document as well as to categorize documents easily. Finally, we introduce a framework for multilingual and able ontology to adopt new languages, also the addition of new data to the existing sources. The framework enriches the domain of the current ontology by integrating an infinite number of languages through mapping the dictionaries. Hence, the framework helps make the whole system and the central knowledge repository language independent. To conclude, we present the results obtained by the experimental implementation of the frameworks to demonstrate the accuracy of the tree and concept hierarchy to amply fulfill our ultimate goal.

Publications

- [1] M. T. M. Ankon, S. N. Tumpa and M. M. Ali, “A Multilingual Ontology Based Framework for Wikipedia Entry Augmentation”, In *19th International Conference on Computer and Information Technology (ICCIT)*, pp. 541-545, IEEE, 2016, held at North South University, Dhaka, Bangladesh.
- [2] S. N. Tumpa and M. M. Ali, “Document Concept Hierarchy Generation by Extracting Semantic Tree Using Knowledge Graph”, In *4th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE 2018)*, pp. 88-91, IEEE, 2018, held at Pattaya, Thailand.

Chapter 1

Introduction

The World Wide Web acts as a repository for an immeasurable amount of data. Each and every day, this amount keeps on increasing. All software and web systems that use such web content as data source have to access this increasing amount of information. However, such systems face many difficulties while traversing through this immense amount of unstructured information. Hence, the concept of bringing the entire information set under a central knowledge repository got significant. In this thesis, we intend to structure these numerous unstructured information with the help of some semantic rules. Also, the augmentation of knowledge graph will help us create the concept hierarchy incorporating additional knowledge for unstructured texts. This chapter introduces the problem which we want to deal in our thesis in Section 1.1. Then, it discusses our research aim, objectives, and contribution in Section 1.2 and Section 1.3. Finally, it gives an overview of the organization of the rest of the thesis in Section 1.4.

1.1 Problem Definition

The theory of bringing all the information on the web under a central knowledge repository has been an attractive concept to the researchers for a long time. The subject has been brought under the domain of Semantic Web [7], which itself is an extension of the traditional web. In this golden era of information technology, Semantic Web [8–10] plays a vital role by establishing well-defined expressions to extract meaningful pattern and information. Most of the information in the traditional web is unstructured and not suitable for machines to retrieve intelligently. The Semantic Web focuses on structuring the immeasurable amount of information to make this information machine readable. As a result, analysis of tonnes of documents which seemed nearly impossible previously can now be interpreted by machines.

The World Wide Web is quickly moving from the era of search engines to the era of discovery engines. Search engines help find information that exists directly in the repository, on the other hand, discovery engines discover information for the users. In this regard, the concept of the

Knowledge graph comes. A Knowledge graph is a knowledge base used by Google and its services to enhance the result of the search engine with information gathered from a variety of sources. To extract knowledge from documents, the concept of Text mining was introduced. Text mining [5] represents the science of retrieving high-quality information from unstructured documents and transforming them into a structured format for further analysis. The research led to the formatting of information in a universally recognized way, such that the central knowledge repository can identify the significance and provide users with ample suggestions. The theory, if implemented, can provide a much better result, particularly for search engines. The formatting even helped to bring newer contents under the knowledge of the central knowledge repository. Linking and retrieving structured information are additionally required to establish the concepts of the Semantic Web. The contents on the web still follow a huge number of formats. On the Semantic Web, the ontologies are used to describe and represent concepts and relations among the concepts within a specified domain. Ontology is very much important for knowledge sharing and understanding that domain. An additional requirement arose with the integration of multiple languages in the web content [7]. The issue put forward the idea, that, the knowledge repository should be able to recognize multiple languages as well. The initial approach consisted of developing individual ontologies for each individual language. This hypothesis conflicted with the idea of the central knowledge repository, its compactness, and diversity. Hence, the concept was updated, to develop a multilingual ontology, which can be adapted by all languages [11]. Thus, the concepts of knowledge graph and ontology possess their own significance in the field of data linking. If the content of the documents can be represented in a structured way, it will be possible to link core concepts with existing knowledge sources. In addition, the obtained information will help integrate the documents into the existing taxonomy as well as categorizing the documents.

1.2 Research Aim and Objective

The main objectives of our thesis are:

1. To propose a framework that represents the document content in a structured way.
2. To propose an algorithm along with some semantic rules to generate the content tree of a document.
3. To establish connections among knowledge from different sources to enhance the document knowledge and to create hierarchical concept recommendations.
4. To propose a framework for multilingual ontology.

1.3 Thesis Contribution

The main contributions of the thesis are as follows:

1. A framework along with an algorithm will be introduced to portray the document contents in a rooted tree structure that preserves semantic relations and extracts the core topics of the document.
2. A system that is expected to connect the knowledge bases to enhance the document content as well as clustering the knowledge to create hierarchical taxonomy of the document. This will help search the document from the numerous documents on the web.
3. A framework will be proposed to augment the domain of multilingual ontology.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 discusses the existing researches related to the problem definition. Through a detail literature review, it highlights the scope of works based on the limitations found in the state of the art. The research questions for the thesis are enumerated at the end of this chapter. Chapter 3 defines the terminologies and notations used in this thesis for a better understanding. Chapter 4 presents the entire concept that establishes the framework for multilingual ontology and the methodology that generates the concept hierarchy. Chapter 5 reports experimental results based on real-world data sets. Finally, Chapter 6 identifies the attainments of this thesis and, then concludes suggesting possible future extensions.

Chapter 2

Related Work and Current Status

In recent years, Text mining has been a major focus in the field of Natural Language Processing (NLP), notably through knowledge augmentation. Additionally, semantic representation of document contents and concept hierarchy generation to classification are also holding significant roles in the era of Semantic Web. Some substantial amount of research has been carried out on this domain. This chapter discusses the present state of document delineation specifically focused on the content representation and concept hierarchy generation. Section 2.1 discusses the methods so far have been used to represent document contents and Section 2.2 discusses the related works on concept hierarchy generation. Finally, we formulate the research questions to be addressed in our thesis in Section 2.3.

2.1 Document Content Representation

Document content representation is very significant for information retrieval. So many techniques have been introduced to effectively represent contents, namely Standard Boolean Model, Bag-of-Words Model, Graph Model, etc. The purpose is not only to provide a simple visual to the document but also to structure the documents for other uses. Many researchers worked on the document content representation for text mining or information retrieval [1, 2, 12–23, 23, 24, 24–28]. Some of them considered the document semantics, while some did not take into account the semantic relations between words.

2.1.1 Standard Boolean Model

The Boolean model is a model based on set theory and boolean algebra. In this model, documents are represented by the index terms assigned to them. All index terms bear equal importance as their weights are binary, either 0 or 1. The Boolean retrieval system is designed to retrieve all stored documents which contain the specific combination of keywords included in the query.

If two query terms are related to an AND connective, both terms are expected to be present to retrieve a particular document. If an OR connective is used, it requires at least one of the query terms to be present to retrieve a specific item. The Standard boolean model does not consider any semantic relation between the index terms [12, 13]. Some researchers have worked on the extended version of the Boolean model to overcome the drawbacks of the Standard boolean model. The Boolean model does not take into account the term weights in queries, and often the result set of a boolean query is either too small or too large. The idea of the extended model is to use partial correspondence and term weights as in the Vector space model. It combines the characteristics of the Vector space model with the properties of Boolean algebra and ranks the similarity between queries and documents. In this model, a material is considered relevant and returns as a result, only if it matches some of the requested terms [14–16].

2.1.2 Vector Space Model

The Bag-of-Words model is a popular method for object categorization, which disregards the semantic relation and order of words. However, it considers the multiplicity of any term in concerned documents [17]. In a vector representation of a document based on this model, each element indicates the number of occurrences of a term. To count the number of appearances, the Bag-of-Words model carries out exact word matching. Such matching can be considered as hard mapping of words to the terms. The Bag-of-Words approach is not a suitable technique to maintain term importance.

In [18], a new approach named Fuzzy Bag-of-Words (FBoW) has been proposed. In this model, a fuzzy mapping is adopted considering the semantic correlation between words which are expressed by cosine similarity measures between embedded words. Fuzzy Bag-of-Words encodes more semantics into the numerical representation as it uses semantic matching instead of exact word string matching.

Another document representation model uses a vector, namely Paragraph vectors. Paragraph vectors is an unsupervised method that learns continuous distributed vector representations for pieces of texts. The fundamental concept of this model is, the texts may vary in length, ranging from a phrase or sentence to a large document [19].

Some researchers worked on the extended versions of the above models to incorporate the semantic relations along with term representation [20, 21].

2.1.3 Graph Representation

Graph representation demonstrates the relationship and structural information conclusively. To incorporate text structure and context with document content representation, researchers have proposed some graphical representation based approaches. A document can be represented

as a graph using vertices and edges. Hence, vertices represent the feature terms and edges demonstrates the relation between concepts.

Graph, $G = \{\text{Vertices}, \text{Edges}\}$

There are generally five different types of vertices in graph representation [29]:

Vertex = $\{F, S, P, D, C\}$

where F is Featureterm, S is *Sentence*, P is *Paragraph*, D is *Document* and C is *Concept*

Edge = $\{\text{Syntax}, \text{Statistical}, \text{Semantic}\}$

The edge relation between two feature terms may differ in the following ways on the context of the graph:

- The togetherness of words in a sentence or paragraph or section or document.
- Common words in a sentence or paragraph or section or document.
- The co-occurrence of words on the fixed window of n words.
- The semantic relation among words, which means the words have a similar meaning, words spelled the same way but have a different meaning, opposite words.

Document representation using graph models provide the opportunity to perform various computations related to term weight and ranking, which is helpful in many applications in information retrieval.

In [1], authors have discussed an approach that represents document contents by a semantic network named “Document semantic core”. At first, every document is represented as a set of concepts. After this, two stages are necessary to generate the “Document semantic core”. Firstly, as each concept could have multiple senses, similarity measures are computed between all possible concept senses. Secondly, a global disambiguation technique is done. The chosen sense of each concept depends on its similarity measure score with all the remaining concept senses happening within the same record. Figure 2.1 represents a semantic network generated from a single configuration of concept senses. Here, $S_2^1, S_7^2, S_1^3, S_1^4, S_4^5, S_2^m$ represents a possible semantic network resulting from a combination of the second sense of the first concept C_1 , the seventh sense of C_2 , the second sense of C_m . The links between the concept senses (P_{ij}) in Figure 2.1 are computed using the similarity measures defined in [1].

[2] proposed an approach for calculating term weights in a text classification task. This proposal considered individual feature while constructing graph avoiding syntactic filters. Term co-occurrence has been used as a measure of dependency between word features. A node is added to the graph if a term is new in the document. Also, an undirected edge is added if they co-occur within a certain window size. Figure 2.2 demonstrates a co-occurrence graph constructed from the sample text shown in the figure. The firmness of this model is the global representation of the context and its ability to model the co-occurrence between the features.

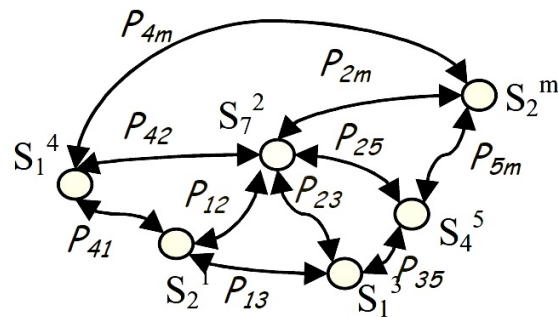


Figure 2.1: A semantic network generated from a single configuration of concept senses [1].

London-based sugar operator Kaines Ltd confirmed it sold two cargoes of white sugar to India out of an estimated overall sales total of four or five cargoes in which other brokers participated. The sugar, for April/May and April/June shipment, was sold at between 214 and 218 dtrs a tonne cif, it said.

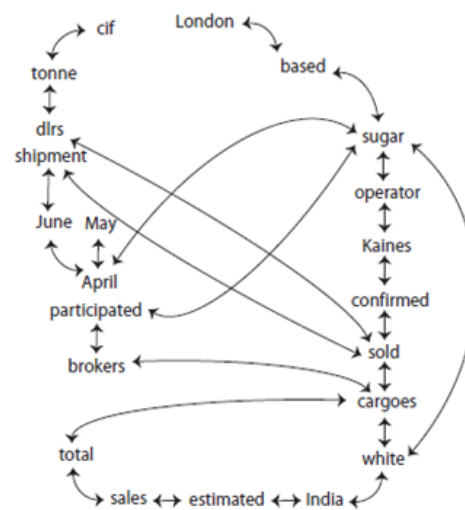


Figure 2.2: Graph constructed from the presented sample text, considering the window size is 2 [2].

In [22], nodes represent sentences and paragraphs, and there is an edge between the nodes if the associated sentences are neighbors or share at least one common keyword. The authors have constructed a small world topology for a document basing on actual document structure. The set of edges depends on the level of meaningfulness among the nodes. The graph considers not only local connections but also long distant relations inside a document.

Graph representation models are suitable for portraying the structural information of texts, but they do not consider the semantic relations between words in general. Some researchers have also addressed the representation of document semantics using network [23, 24]. Semantic similarity is stated using the Thesaurus graph and Concept graph. In the Thesaurus graph, terms are noted as vertex and meaning relationships such as synonym and antonym are noted as edges. In [23], the authors have analyzed statistically three semantic networks: word associations, WordNet, and Roget's Thesaurus. In [24], the author has discussed a system for constructing conceptual graph representation of text by using a combination of existing linguistic resources. WordNet and VerbNet are used to find the semantic roles in a sentence. These semantic roles

are used to process the raw text and map the disambiguate nouns to the WordNet concepts.

In [25], a graph-based text representation method was designed under word semantic space to obtain parts of speech, order, frequency, co-occurrence, and context of words in the document. Apart from this, a semantics-based graph structure was proposed to hold more structural information and mutual semantic relationship among words in [26].

In [27], a term graph model was proposed to represent the content and relationship among words. [28] presented another graph-based method for document representation incorporating co-occurrence networks and dependency networks.

2.2 Concept Hierarchy

Besides enlightening the representation of document contents, this section focuses on the hierarchical representation of concepts.

In [30], a hierarchical organization of concepts from a set of documents has been proposed without using any training data or standard clustering techniques. The authors preferred to use a co-occurrence technique to create the hierarchy of selected essential terms. [31] proposed an approach for concept hierarchy using formal concept analysis. This approach is based on the distributional hypothesis, and the hierarchy between words have been decided considering syntactic dependencies. A graph-based text mining technique, GDClust has been proposed in [32] based on co-occurrence of frequent senses to present text document as hierarchical document graphs. In [33], each document is represented as a concept tree using the concept associations obtained from a classifier. The concept vectors of a document are created using the highest-weighted top 20 concepts derived from the ACM's classification hierarchy.

2.3 Research Questions

After discussing the mentioned researches in the above sections, we can conclude that the contribution of the researchers in this field is quite significant. However, no researcher focused on the collection of more knowledge regarding the core concepts of a document from the knowledge bases to cluster the concepts and also to get the document hierarchy. Furthermore, from the above discussion, it can be said that the idea of representing document content using a tree is a barely touched topic. Trees are used in the taxonomic representations of concepts, but according to semantic relation and context, it is not in the field of document content representation. From the previous studies, we can formulate the following issues as our research questions to be addressed in this thesis.

Representing Document Content Using a Tree: A framework, along with an algorithm, will be helpful to portray the document contents in a structured way. A rooted tree structure

is needed to preserve semantic relations and contexts to extract the core topics of the document.

Knowledge Graph Augmentation: A system that is required to connect the knowledge bases to enhance the document content. Also, the system will cluster the knowledge to create hierarchical taxonomy of the document. This method will help to search a document from the various materials on the web.

Framework of the Multilingual Ontology: The part of hierarchy generation is mostly dependent on the extracted knowledge from the DBpedia ontology. To make a language-independent system, a robust multilingual ontology is an essential requirement. Therefore, the ontology should be able to adopt changes and addition of new knowledge without much hassle.

Chapter 3

Preliminaries

Our thesis develops a system that will generate a content tree from an unstructured document augmenting knowledge graphs and establishes the framework for a language-independent ontology. This chapter will be helpful to develop a basic understanding of the entire domain.

3.1 Basic Terminology

In this section, the basic terminologies on ontology, Natural Language Processing (NLP), Text mining, knowledge bases, etc. will be discussed. We will also elaborate on the concept of multilingual ontology here.

3.1.1 Semantic Web

Semantic Web [8, 10, 34], as an extension of traditional web, is concerned about structuring the vast amount of unstructured data to make the entire knowledge content machine-readable, as well as machine interpretable. In the Semantic web, information is given well-defined meaning to cooperate human-computer interaction on a large scale. According to the W3C [35], “The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries”.

In early 1960, the cognitive scientist Allan M. Collins, linguist M. Ross Quillian, and psychologist Elizabeth F. Loftus form the concept of the semantic network model to represent knowledge semantically. Later, the idea was applied in the context of the modern web by integrating machine-readable metadata about pages and the inter-relations among them with the human-readable web pages. This new concept allows quicker access to the internet by automated agents and performs more tasks on behalf of users. The inventor of World Wide Web and the director of the World Wide Web Consortium (W3C), Tim Berners-Lee named the extension of the traditional web as “Semantic Web” with the idea of a capable knowledge repository that will

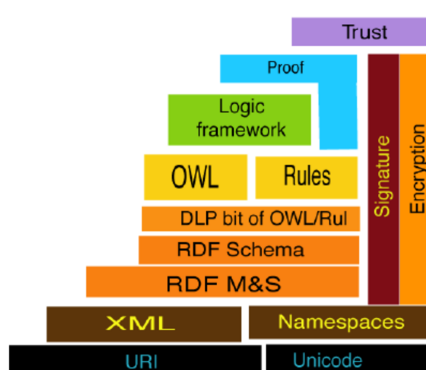


Figure 3.1: Latest version of the Semantic Web Stack [3]

analyze all the data available in the web to infer meaningful information by machines [36]. The idea has gradually evolved to reality, and as of 2013, more than 4 million web domains contained Semantic Web markup [37].

Figure 3.1 illustrates the architecture of the Semantic Web. The functions and relationships of the components can be summarized as follows [38, 39]:

Extensible Markup Language (XML) The Extensible Markup Language (XML) is a subset of Standard Generalized Markup Language (SGML) that defines a set of rules for encoding documents in human and machine readable formats [40]. The contents of the document have no association with semantics in XML. It only provides tags to the document contents to structure them.

XML Schema XML Schema is a language that provides and restricts more formally the structure and content of elements in XML documents.

Resource Description Framework (RDF) The Resource Description Framework (RDF) makes statements about resources in expressions using the form subject–predicate–object, known as triples. The subject denotes the resource, and the predicate denotes the resource’s attributes or aspects, expressing a relationship between the subject and the object. RDF is a basic standard of the Semantic Web. Figure 3.2 demonstrates an RDF model for the statement: “*Picasso’s home address is 31 Art Gallery, Madrid, Spain*”.

RDF Schema RDF Schema structures RDF documents to describe RDF based resource properties and classes, with semantics for generalized hierarchies of such properties and classes.

Web Ontology Language (OWL) Ontology is a formal way to describe class hierarchies and taxonomies. It also adds more vocabulary to the properties and classes. An ontology describing families may include axioms stating that a “hasMother” property only exists between two individuals when “hasParent” is also present and that class

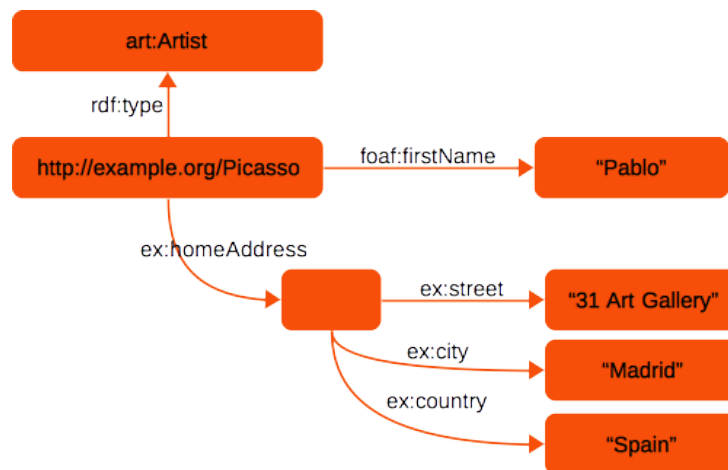


Figure 3.2: An example RDF model [4]

“HasTypeOBlood” individuals are never related to members of the “HasTypeABBlood” class via “hasParent”.

SPARQL SPARQL is an RDF query language for databases which can retrieve and manipulate data stored in RDF format.

Rule Interchange Format (RIF) RIF is an XML language for expressing web rules that a computer can execute.

3.1.2 Text Mining

The concept of text mining was introduced to extract knowledge from documents. Text mining [5] is the science of retrieving high-quality information from unstructured documents and transform them into a structured format for further analysis. Text mining is also known as Text data mining, which is closely related to Text analytics [41]. It can also be considered as an extension of data mining or knowledge discovery from (structured) databases [42].

Text mining scans documents written in natural languages to model those for predictive classification or to populate a database using the extracted information. The primary purpose is to convert the text into meaningful information or data for further processing and analysis. Information retrieval, pattern recognition, lexical analysis to study distributions of word frequency, tagging or annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analysis are also included in text mining. Text mining can be divided into two phases, Text refining, and Knowledge distillation.

Text refining Text refining transforms unstructured documents into an intermediate form which is semi-structured such as the conceptual graph representation or structured such as the

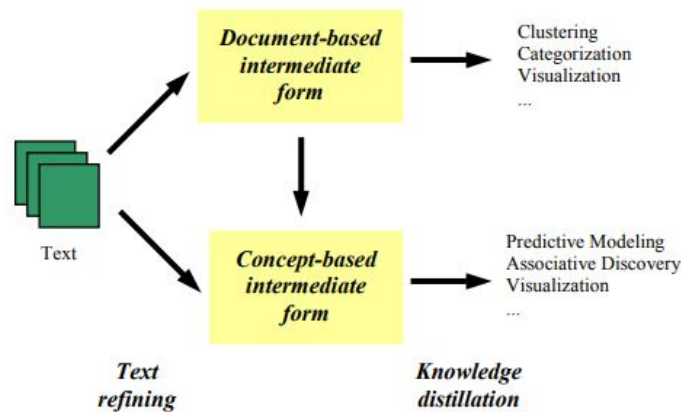


Figure 3.3: A Text mining framework [5]

relational data representation. The intermediate form can be document based as well as concept based.

Knowledge distillation Knowledge distillation from document-based intermediate form derives pattern and relationship across documents. Concept-based IF, on the other hand, achieves pattern and relationship across objects or concepts. By extracting object information relevant to a domain, a document-based intermediate form can be converted into a concept-based intermediate form. Knowledge distillation can be performed on a database to infer more knowledge.

Figure 3.3 demonstrates a generic framework involving two phases of Text mining.

3.1.3 Coreference Resolution

Coreference resolution is the process of determining the named, nominal, and pronominal entity that refer to the same entity in natural language [43, 44]. Figure 3.4 demonstrates an example of coreference resolution. Here, the pronoun “I”, “my” and “she” refer to the same entity, a girl stating the quoted statement. Additionally, “Mr. X” and “he” refer to another entity. Coreference resolution is an important subtask that involves understanding the natural language. It helps in summarizing documents, answering questions, extracting information, etc.

3.1.4 Some Knowledge Bases

In Natural Language Processing (NLP), a knowledge base is a centralized database for spreading information and data. Knowledge-based solutions provide a feasible alternative in the fields of NLP to the approaches based only on grammars and other linguistic information. Understanding the following knowledge bases will be required for our thesis.

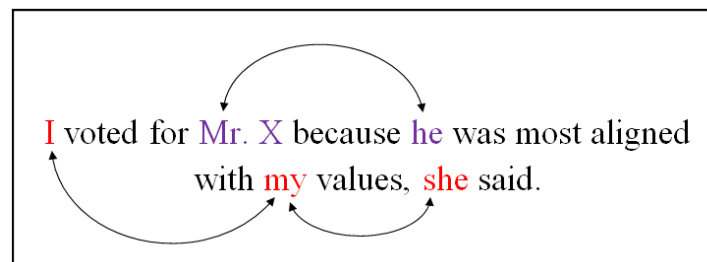


Figure 3.4: An example of Coreference resolution

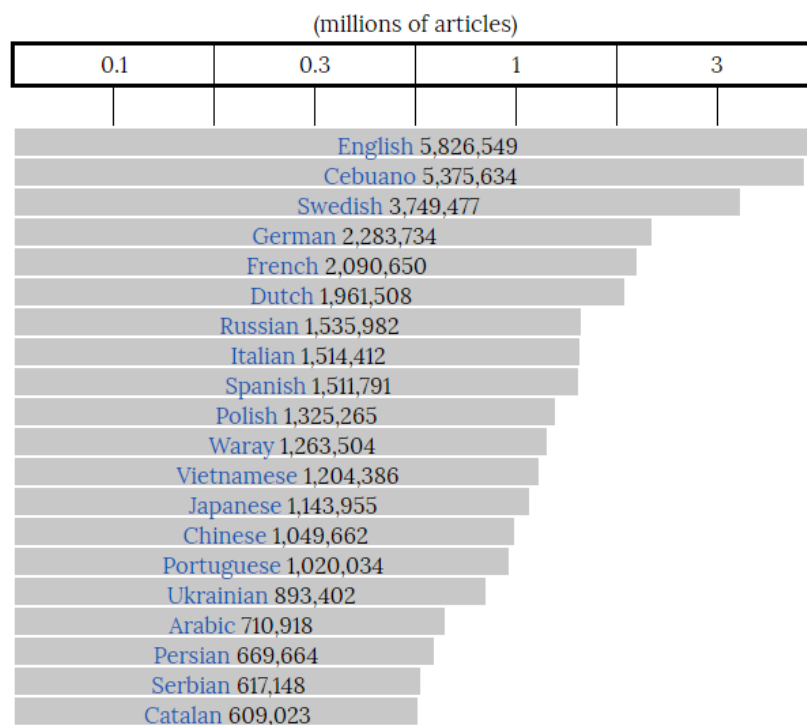
Wikipedia

Wikipedia is a web-based, multilingual, free encyclopedia based on a wiki, a model of openly editable and visible content. It is the largest and most popular established data source on the World Wide Web and is one of the most popular rank websites of Alexa's [45]. The Wikimedia Foundation, a non-profit organization that operates on money it receives from donors, owns and supports Wikipedia. With its inauguration in 2001 by Jimmy Wales and Larry Sanger, Wikipedia has integrated numerous languages and has evolved to accommodate more and more information. With 5,826,549 articles (as of March 19, 2019), the English Wikipedia is the richest of the more than 290 Wikipedia encyclopedias. The English Wikipedia has already been prepared in its machine-readable format with authentic mapping techniques and acts as a reference for countless research relevant to Semantic Web. Wikipedia currently has 301 language editions. The six most significant versions are the English, Cebuano, Swedish, German, French and Dutch Wikipedias. Figure 3.5 demonstrates the logarithmic graph of the 20 most extensive language editions of Wikipedia (as of March 19, 2019). This resource is one of the most reliable data sources for the researchers of this era [6].

WordNet

WordNet is a free online large English lexical database of nouns, verbs, adjectives, and adverbs grouped into sets of synonyms, each expressing a lexicalized concept. Conceptual-semantic and lexical relations interlink the synonym sets. Generally, vocabulary can be defined as a set W of pairs (f, s) , where a form f is a string over a finite alphabet and a sense s is an element from a given set of meanings. A string of ASCII characters represents a form, and a sense is represented by the set of synonyms with that meaning in WordNet. WordNet contains over 118,000 different word forms and over 90,000 different word senses, or more than 166,000 (f, s) pairs [46].

WordNet seems like a thesaurus, in which the words are grouped based on their meanings. However, there are some significant dissimilarities. Firstly, WordNet connects not only word forms but also specific senses of words. As a result, words that are found close to one another in the network are semantically disambiguated. Secondly, WordNet labels the semantic relationships between words, the groupings of words in a thesaurus; on the other hand, does not



The unit for the numbers in bars is articles.

Figure 3.5: Logarithmic graph of the 20 largest language editions of Wikipedia [6]

follow any specific pattern other than meaning similarity [47].

Google Knowledge Graph

The Google Knowledge Graph (GKG) is a knowledge base used by Google to enhance its search capability [48]. It collects all connected information of the searched entity and displays in infobox format to the users next to the search result. This knowledge graph is powered by another knowledge hub Freebase, which gathers structured information from various sources and connects them. The Knowledge graph intends to connect structured information. Google provides the Knowledge graph search API to find entities in the knowledge graph [49].

DBpedia

Though Wikipedia is the largest source of information, they are only suitable for human eyes. When it comes to a machine, Wikipedia itself does not hold much significance. A machine-readable version of Wikipedia was established through the project of DBpedia in 2007 [50]. DBpedia is a community effort to extract structured content from Wikipedia projects. DBpedia puts forward the knowledge base in Wikipedia in a machine-readable format and makes it accessible as Open Knowledge Graph (OKG) for everyone through the web. Similar to Google, this knowledge graph also stores knowledge in a standardized and machine-readable format, thus

allows users to semantically query relationships and properties of Wikipedia resources. DBpedia also provides links to other related data sets [51]. According to DBpedia website, the data set of DBpedia knowledge base describes 4.58 million things; among them, 4.22 million are classified in a compatible ontology [52].

YAGO

YAGO (Yet Another Great Ontology) is an open source semantic knowledge base which automatically extracts knowledge from Wikipedia, WordNet, GeoNames and other data sources. It was developed by Max Planck Institute for Computer Science in Saarbrücken [53]. According to its website, YAGO knows more than 10 million entities and contains more than 120 million facts about these entities [54].

3.1.5 Tree, Rooted Tree, Forest

The British mathematician Arthur Cayley invented the term *Tree* in 1857 [55]. Consider a graph, $G = \{N, E\}$ where N and E are the set of nodes and set of edges respectively. A tree is an undirected graph G that satisfies any of the following equivalent conditions [56]:

- G is connected and acyclic (contains no cycles).
- G is acyclic, and a simple cycle is formed if any edge is added to G .
- G is connected but would become disconnected if any single edge is removed from G .
- G is connected, and the 3-vertex complete graph K_3 is not a minor of G .
- Any two vertices in G can be connected by a unique simple path.

If G has n vertices, then the following conditions can be said equivalent to the above statements:

- G is connected and has $n - 1$ edges.
- G is connected, and every subgraph of G includes at least one vertex with zero or one incident edges.
- G has no simple cycles and has $n - 1$ edges.

A tree T is rooted if there exists a distinct vertex v designated the *Root* [57]. The edges of a directed rooted tree can be assigned either away from or towards the root. When a directed rooted tree has an orientation away from the root, it is called an arborescence branching, or out-tree. On the other hand, if it has an orientation towards the root, it is called an anti-arborescence or in-tree.

A *forest* is an undirected acyclic graph consisting of disjoint union of trees. We can count the number of trees within a forest by the difference between total vertices and total edges. If TV is the total number of vertices present in a forest and TE is the total number of edges of that forest, then $TV - TE$ is the number of trees present in a forest.

Chapter 4

Methodology

The proposed system and its subsequent implementation are divided into three major hypotheses, namely:

- Document content tree generation.
- Concept hierarchy creation.
- Framework for a multilingual ontology.

The first hypothesis designates the generation of the content tree from an unstructured text document. After that, the second hypothesis creates the concept hierarchy augmenting the knowledge bases. In the end, the third hypothesis introduces a concept of multilingual ontology for DBpedia data set augmentation. The mentioned hypotheses will be discussed elaborately in the following sections.

4.1 Document Content Tree Generation

Document content tree can be defined as a tree-based representation of any document, that demonstrates the dependencies among words in that document. A content tree is an acyclic directed graph, $G = \{N, E\}$ where N and E are the set of nodes and edges respectively.

In the content tree model proposed in the first hypothesis, every document is converted into a rooted tree structure based on the concepts present in it. There can be multiple trees or a forest if the document discusses various topics. A content tree has three types of elements: root, nodes, and edges. The general structure of a document content tree is as follows:

Root A tree contains information regarding the root node only. Root has no parent, as well. Root has been chosen from the main entities of the sentences in the provided document.

Nodes Nodes denote the concepts of the document. Nodes may consist of single or multiple concepts. With the help of the semantic rules, the concepts will be merged to accommodate them in a single node.

Edges A directed edge between two nodes resembles the relation between the nodes. Usually, verbs of the sentences are chosen as the edges as verbs represent the relations in the sentences.

The generation of the content tree varies with the language of the document. The tree generation module comprises the following sub-modules:

4.1.1 Tokenization and Preparing Tagged Document

The document is segmented based on some pre-defined separators. At this stage, we need to preserve some information together, like name, date, etc. For instance, if we use white space as a separator in “মাইকেল জোসেফ জ্যাকসন একজন সঙ্গীতশিল্পী ছিলেন। [English translation: Michael Joseph Jackson was a singer]”, the name “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]” will be tokenized into three tokens as “মাইকেল [English translation: Michael]”, “জোসেফ [English translation: Joseph]” and “জ্যাকসন [English translation: Jackson]” which is not anticipated. For this reason, external knowledge is incorporated to get the desired tokens. We also extract information related to a token. Some examples are the sentence position of that particular token in the provided document, token position in a sentence, and overall token position in the document. These are used to determine the coreference resolution in the next step. Furthermore, this information helps in the tree construction algorithm.

4.1.2 Applying Coreference Resolution

A combination of heuristic and supervised methods is used for coreference resolution [44]. It helps to find all expressions that refer to the same entity. We consider a specific window size for the referred entity. Therefore, our method recursively tracks the possible antecedent, and the referred entity replaces the pronoun. Let us consider the first two sentences of the sample document shown in Figure 4.1: “মাইকেল জোসেফ জ্যাকসন এর জন্ম ২৯ আগস্ট, ১৯৫৮ সালে। তার মৃত্যু ২৫ জুন, ২০০৯ সালে। [English translation: Michael Joseph Jackson was born on August 29, 1958. He died on June 25, 2009.]”. After applying the coreference resolution, the pronoun of the second sentence “তার” will be replaced by the referred noun “মাইকেল জোসেফ জ্যাকসন[English translation: Michael Joseph Jackson]”. Finally, the sentences will be - “মাইকেল জোসেফ জ্যাকসন এর জন্ম ২৯ আগস্ট, ১৯৫৮ সালে। মাইকেল জোসেফ জ্যাকসন এর মৃত্যু ২৫ জুন, ২০০৯ সালে। [English translation: Michael Joseph Jackson was born on August 29, 1958. Michael Joseph Jackson died on June 25, 2009.]”.

মাইকেল জোসেফ জ্যাকসন এর জন্ম ২৯ আগস্ট, ১৯৫৮ সালে। তার মৃত্যু ২৫ জুন, ২০০৯ সালে। তিনি একজন মার্কিন সঙ্গীতশিল্পী, নৃত্যশিল্পী, গান লেখক, অভিনেতা, সমাজসেবক এবং ব্যবসায়ী। তাকে পপ সঙ্গীতের রাজা হিসেবে আখ্যায়িত করা হয়।

Figure 4.1: A snippet of some lines from a Bengali document

4.1.3 Labeling and Filtration of Extracted Information

In corpus linguistics, part-of-speech tagging (PoS tagging) is also known as grammatical tagging that assigns parts of speech to each word, such as noun, verb, adjective, etc. An extensive dictionary is used to determine the parts of speech along with some additional knowledge regarding the extracted tokens. For example, the parts of speech of “বাংলাদেশ [English translation: Bangladesh]” is a *Noun* but more specifically, it is the name of a *Country*. If we label the first line of the sample text shown in Figure 4.1, the tagged sentence will be - “মাইকেল জোসেফ জ্যাকসন <Noun, Name, Person, Male> এর <Stop word> জন্ম <Verb> ২৯ <Number> আগস্ট <Noun, Month> ১৯৫৮ <Number> সালে <Year>” .

Furthermore, not all tokens hold significance for the content tree. Thus, we use Bengali dictionaries for less significant words, to filter the tokens. Here, the word “এর” can be removed from the above sentence.

Following these sub-modules, the text of Figure 4.1 will be converted into a processed tagged document. The tagged document for the initial two lines is demonstrated in Table 4.1. The first four columns hold information regarding the token position and the last three columns are related to the parts of speech and additional information of the tokens. The first column represents the name of the extracted tokens, the second column is for the line number of the tokens, the following column contains the information of the token position in the residing line, and the succeeding column is for the overall token position in the document. The fifth column is for the parts of speech of the tokens, the sixth one is for the additional tags, and the last one is optional as it holds the other details of tokens if there is any.

4.1.4 Tree Construction

We construct the set of nodes N and edges E after filtration. Every edge in E_{ij} is considered as a directed edge between two vertices N_i and N_j . We developed an algorithm to generate the content tree. The tree construction algorithm is shown in Algorithm 1. The input of the tree algorithm is the processed tagged tokens, and the output is the representation of the content tree.

Table 4.1: Processed tagged document for the first two lines of Figure 4.1

Token Name	Line No	Token Position in Line	Overall Token Position in the document	Pos Tag of the Token	Additional Tag	Other Detail
মাইকেল	1	1	1	Noun	Person	Name
জোসেফ	1	2	2	Noun	Person	Name
জ্যাকসন	1	3	3	Noun	Person	Name
জন্ম	1	5	5	Verb	Verb	
২৯	1	6	6	Noun	Number	
আগস্ট	1	7	7	Noun	Month	
১৯৫৮	1	8	8	Noun	Number	
সালে	1	9	9	Noun	Year	
মাইকেল	2	1	10	Noun	Person	Name
জোসেফ	2	2	11	Noun	Person	Name
জ্যাকসন	2	3	12	Noun	Person	Name
মৃত্যু	2	5	14	Verb	Verb	
২৫	2	6	15	Noun	Number	
জুন	2	7	16	Noun	Month	
২০০৯	2	8	17	Noun	Number	
সালে	2	9	18	Noun	Year	

Figure 4.2 demonstrates the content tree constructed from the texts shown in Figure 4.1 before applying the merging rules.

Semantic Rules

We merge some nodes to reduce the number of nodes as well as the height of the content tree that may result in the reduction of traversal time. Therefore, we propose some semantic rules to unify adjectives with a noun, adjectives with adjectives, etc. and we consider the following notations to establish the theoretical terms for the rules: *CurrNode* is the current node, *PrevNode* is the immediately preceding node of the current one in the tree, and *ResultNode* is the output node after applying a semantic rule. E_1 is the edge between the previous node of *PrevNode* and *PrevNode* itself. E_2 is the edge between *PrevNode* and *CurrNode*. The semantic rules are as follows:

- **Rule #1:**

If the parts of speech tag of the current node and previous node both are *Noun* and the additional tags are also same, we can merge them into a single node. The parts of speech of the resultant node will be the same as the current node. Let us consider “সঙ্গীত” as the

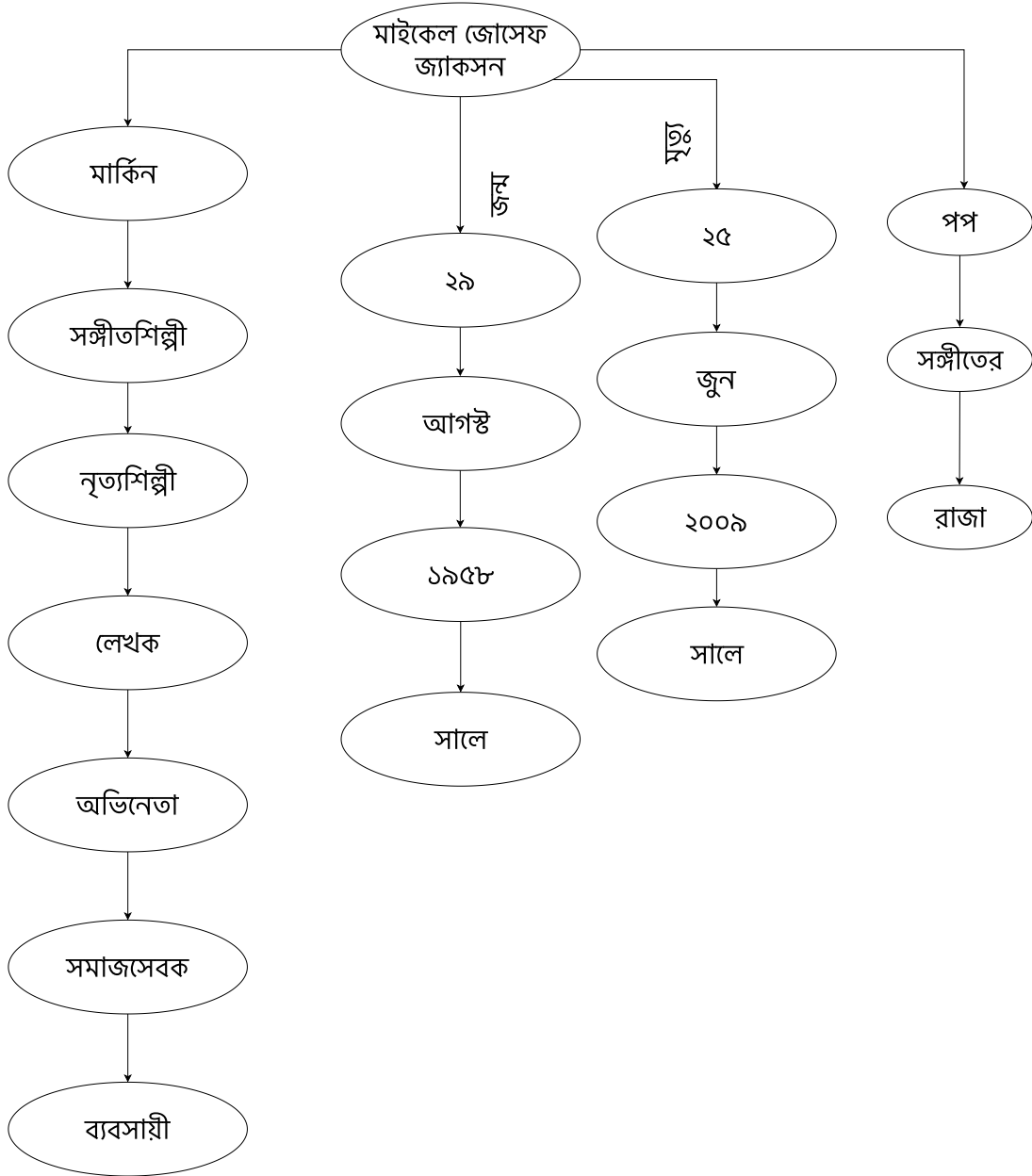


Figure 4.2: Content tree from the sample texts of Figure 4.1

current node and “পপ” as the previous node. Parts of speech of both tokens are *Noun*. We can merge these two tokens into a single resultant node - “পপ সঙ্গীত”.

IF $CurrNode \rightarrow Noun \wedge PrevNode \rightarrow Noun$ **THEN**

$ResultNode = \text{MERGE}(PrevNode, CurrNode)$,

$ResultEdge = \text{MERGE}(E_1, E_2)$,

$POS_OF(ResultNode) = POS_OF(CurrNode)$

• **Rule #2:**

If the parts of speech tag of the current node is *Noun* and the previous node is *Adjective*, then it can be assumed that the last node is defining the current node. We can merge them

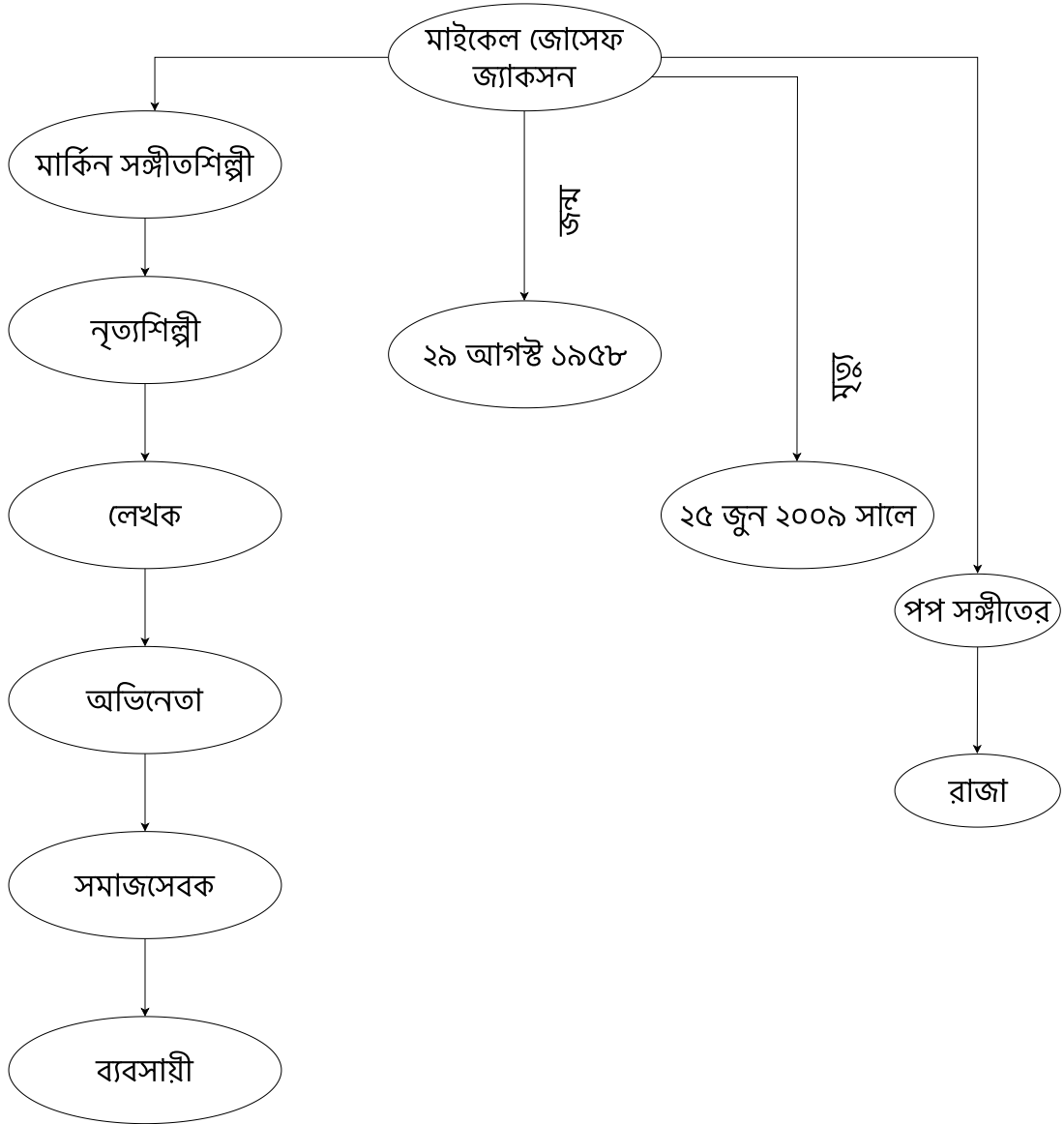


Figure 4.3: Reduced content tree from the sample texts of Figure 4.1 after applying the semantic rules

into a single node. The parts of speech of the resultant node will be the same as the current node. Let us consider “সঙ্গীতশিল্পী” as the current node and “মার্কিন” as the previous node. After merging these two tokens, the resultant node will be - “মার্কিন সঙ্গীতশিল্পী”.

IF $CurrNode \rightarrow Noun \wedge PrevNode \rightarrow Adjective$ **THEN**

$ResultNode = \text{MERGE}(PrevNode, CurrNode),$

$ResultEdge = \text{MERGE}(E_1, E_2),$

$POS_OF(ResultNode) = POS_OF(CurrNode)$

• **Rule #3:**

If the parts of speech tag of the current node is *Month* or *Year* and previous node is *Number*, then it can be assumed that the previous node and current node together resemble a *Date*.

We can merge them into a single node. The parts of speech of the resultant node will be *Date*. Let us consider “আগস্ট” as the current node and “২৯” as the previous node. After merging these two tokens, the resultant node will be - “২৯ আগস্ট”.

IF ($CurrNode \rightarrow Month \vee CurrNode \rightarrow Year$) $\wedge PrevNode \rightarrow Number$ **THEN**

$ResultNode = \text{MERGE} (PrevNode, CurrNode),$

$ResultEdge = \text{MERGE} (E_1, E_2),$

$POS_OF (ResultNode) = Date$

- **Rule #4:**

If the parts of speech tags of the current node and previous node both are *Adjective* and the additional tags are also same, we can merge them into a single node. The parts of speech of the resultant node will be the same as the current node. Let us consider “মার্কিন” as the current node and “জনপ্রিয়” as the previous node. Parts of speech of both tokens are *Adjective*. We can merge these two tokens into a single resultant node - “জনপ্রিয় মার্কিন”.

IF $CurrNode \rightarrow Adjective \wedge PrevNode \rightarrow Adjective$ **THEN**

$ResultNode = \text{MERGE} (PrevNode, CurrNode),$

$ResultEdge = \text{MERGE} (E_1, E_2),$

$POS_OF (ResultNode) = Adjective$

Figure 4.3 demonstrates the reduced content tree after applying the semantic rules.

4.2 Concept Hierarchy Generation

In Natural Language Processing (NLP), concepts can be expressed as senses of a document. A single concept may consist of a single word or can have multiple words. Concept hierarchy establishes the hierarchical structure in this scenario. Concept hierarchy, or taxonomy, is a mechanism to demonstrate the generalized hierarchical relationships among the concepts. It ensures efficient categorization of a document.

Concept hierarchy increases the efficiency of document retrieving on a large scale. To implement the concept hierarchy, we maintain a few steps which are discussed as follows:

4.2.1 Concept Extraction from Provided Document

The concept of document content tree mentioned in the previous section helps us to extract the major concepts of a document. Here, we consider the roots of the trees in the set of core concepts as the rooted tree holds information related to the root. Apart from the roots of document content trees, frequent occurrence of concepts may have significance on the document type. From the

Michael Jackson: American singer-songwriter	type : Abstraction100002137
Michael Jackson: Writer	type : WikicatAfrican-AmericanMaleDancers
Michael Jackson: Canadian actor	type : Biographer109855433
Michael Jackson: American basketball player	type : WikicatSongwriters
Michael Jackson: Radio DJ	type : Musician110340312
Michael Jackson: ["Person", "Thing"]	type : Poet110444194
Michael Jackson: ["Person", "Thing"]	type : Singer110599806
Michael Jackson: ["Thing", "Person"]	type : WikicatAmericanDanceMusicians
Michael Jackson: ["Person", "Thing"]	type : WikicatAmericanDancers
Michael Jackson: ["Person", "Thing"]	type : WikicatChildPopMusicians
occupation : record	type : WikicatDiscographies
occupation : Singer	WikicatAmericanPeopleOfJewishDescent
occupation : philanthropist	type : Wikicat20th-centuryAmericanSingers
occupation : actor	type : Screenwriter110564400
occupation : businessman	type : WikicatAmericanSingers
occupation : dancer	type : Catalog106487897
occupation : songwriter	type : MusicalArtist
type : Unfortunate109630641	type : WikicatPeopleAcquittedOfSexCrimes
type : Q215627	type : Person
tvne : Person100007846	tvne : WikicatSoulSingers

Figure 4.4: A snippet of gathered information from knowledge bases related to “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]”.

tree of Figure 4.3, we shall consider “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]” as the core concept.

4.2.2 Knowledge Extraction Using Knowledge Bases

After extracting core concepts from the document, existing knowledge bases, like Google Knowledge Graph [49], DBpedia [51], YAGO [53], WordNet [46] etc. are used to gather more information. There are some other enriched knowledge graphs, such as: Cyc and OpenCyc [58], Freebase [59], Wikidata [60], NELL [61], Yahoo’s Knowledge Graph, Microsoft’s Satori, Facebook’s Entities Graph etc.

Knowledge graphs provide knowledge as linked data. The yielded information is in raw format that requires to embed to be more meaningful. Figure 4.4 demonstrates a snippet of gathered information from knowledge bases related to “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]”.

4.2.3 Word Embedding and Similarity Checking

The process of word embedding demonstrates a class of approaches for representing words in a continuous vector space where semantically similar words are mapped to nearby points [62]. There are two popular methods of word embedding from the text, namely Word2Vec [63] and GloVe [64]. These algorithms embed words in a vector space such that words that share common contexts in the corpus are located close to one another in the space.

Cosine similarity is one of the popular similarity measures for text documents. The similarity is measured as the cosine of the angle between two vectors according to the orientation. The cosine of 0° is 1 for two vectors with the same orientation. The similarity is -1 if they are diametrically opposite [65]. If \vec{D}_A and \vec{D}_B are two vector documents, their cosine similarity is measured using the following formula,

$$\text{Cosine Similarity, } S_C = \frac{\vec{D}_A \cdot \vec{D}_B}{|\vec{D}_A| \times |\vec{D}_B|} \quad (4.1)$$

We incorporated Word2Vec method to embed the information in the vector space and to determine the cosine similarity among the information extracted from the knowledge bases.

4.2.4 Concept Clustering and Hierarchy Generation

The extracted information is clustered based on the similarity weight. There are pre-defined cluster tags for all clusters which are obtained from DBpedia Ontology. The taxonomic representation of the extracted information is generated following the class hierarchy of this particular ontology.

For example, the intermediate labels for “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]” are *Singer* and *Actor* in the taxonomy. From these labels, we shall extract the upper level of the hierarchy, which is a bit more generalized than the previous one. Here, the upper-level label is *Person*. In a nutshell it can be said, the texts of Figure 4.1 is related to a *Person*, more specifically associated to a *Singer* and an *Actor*. Section 4.2.4 illustrates the generated content tree from the document on “Sakib Al-Hasan” and “Michael Joseph Jackson” showing the concept hierarchy.

4.3 A Framework for Multilingual Ontology

We have gathered knowledge on the core concepts from various knowledge bases, including DBpedia. The quality of the generated hierarchy is highly related to the amount of effective knowledge accumulation. The data set of DBpedia and other knowledge bases are quite sufficient for some major languages. However, for the Bengali language, the data sets are inadequate. English is the most enriched language for Wikipedia to date [66]. Since the DBpedia is based on the data set of Wikipedia [67], it can be suggested that English is the richest data set for the ontology of DBpedia.

We put forth here the concept to establish a framework to augment the data sets of the knowledge bases such as DBpedia. In Table 4.2, it can be clearly said that the number of instances for a set of selected classes within the canonicalized DBpedia data sets of Bengali language is less

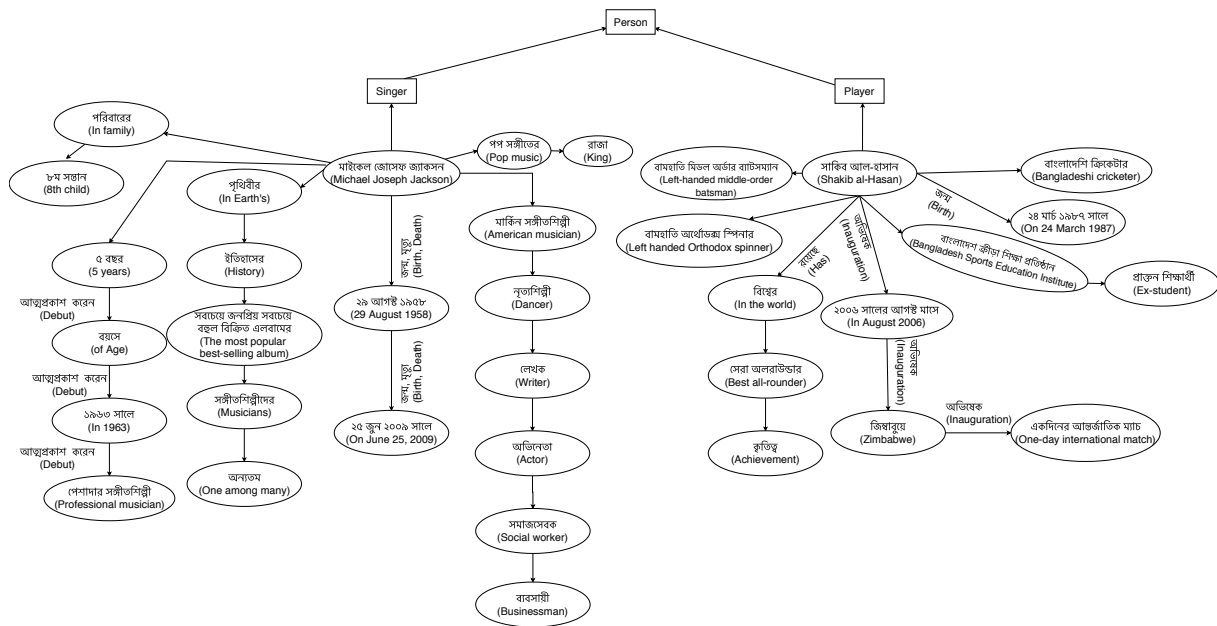


Figure 4.5: Document content tree generated from the document on “সাকিব আল হাসান [English translation: Sakib Al-Hasan]” and “মাইকেল জোসেফ জ্যাকসন [English translation: Michael Joseph Jackson]”.

compared to English [68]. According to these statistics, Bengali is one of the languages yet to be brought under the domain of a single ontology. We propose here the main concepts required for the augmentation.

The initial step to implement a single ontology is To augment the DBpedia data sets, which can use dictionaries for each language as sources and link up all the data on its own. Hence, a multilingual ontology is the first step. Researchers have developed the basic framework for the multilingual ontology [69]. The next step will be, to enrich the data set of the ontology. In such a case, a mapping technique will be implemented, which will link up all words, from different languages, with the same meaning. The ontology will then create a universal identifier for each particular concept. The final step will be to retrieve the information for use in any web content. An addition to this step is rendering of web content, which allows the user a more straightforward way to link up the content with the ontology.

The previously mentioned steps are discussed with details in the following subsections.

4.3.1 Multilingual Ontology

The concept of multilingual ontology follows that there should be one, highly enriched ontology. The ontology will have the capability to utilize an infinite number of dictionaries. Thus, these dictionaries will enable the ontology to cope up with multiple languages [11]. At the moment, we are assuming that there will be one dictionary for each language. The multilingual ontology acts as a formal and explicit specification of a concept, which can be shared by all existing

Table 4.2: The number of instances for a set of selected classes within the canonicalized DBpedia data sets for each language.

Language	en	bn
Person	763643	425
Artist	61073	0
Actor	2431	0
MusicalArtist	34246	0
Athlete	185126	0
Politician	23096	0
Work	333269	125
Book	26198	40
Film	71715	62
MusicalWork	159070	0
Album	112248	0
Single	41774	0
Software	27947	23
TelevisionShow	23480	0

and upcoming dictionaries [70]. It consists of a set of distinct concepts, having inter-relations, denoted by a set of relations [11].

Existence of such an ontology eases the opportunity to integrate newer languages. Thus, the integration of a language-based ontology will augment the data set of DBpedia and other sources.

4.3.2 Multilingual Ontology Mapping

The mapping of multilingual ontology acts as the most significant part of the augmentation of DBpedia. We need to consider the existence of a multilingual ontology. A process that will require frequent changes to the multilingual ontology will never be an efficient way. Thus, the requirement for a useful mapping technique is of utmost significance.

For our research, we are proposing a technique, based on [7], that will take an established dictionary as a source, given that the dictionary is already integrated with the multilingual ontology, and map a new dictionary with it. This process will extend the universal resource identifiers that are linking the ontology with the existing vocabulary, to the new dictionary, linking up its vocabulary.

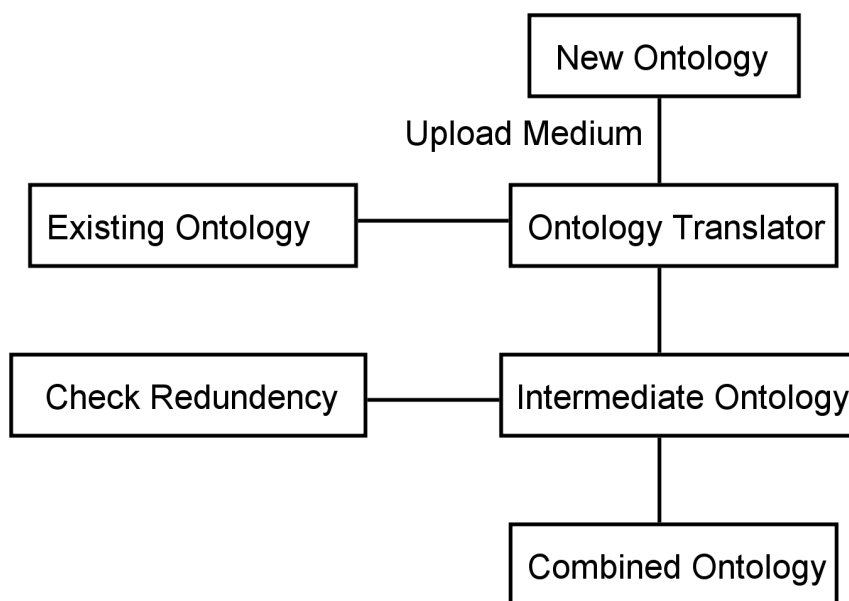


Figure 4.6: Generation of combined ontology.

The first requirement to augment the entries is the ontology of the new language. Considering it as *new ontology*, we upload it to a *medium*, which is a web interface. This medium acts as a window to provide the new ontology to the existing system. Upon receiving the *new ontology*, a translation has to be conducted, to generate an ontology similar to the *existing ontology*. From this stage, the checking for redundancy will be conducted. Any concept already existing in the *existing ontology* will be discarded, and new concepts will be added. The final, *combined ontology* will thus, be a combination of the *existing ontology* updated by the addition of some new concepts [11]. Its generation is sequentially demonstrated in Figure 4.6.

At this stage, we establish the mapping between the two dictionaries. For this, we will require translation from the *new dictionary* to the *existing dictionary*. Let us consider that we have the Bengali dictionary as the *new dictionary* and English dictionary as the *existing dictionary*. After translation, we will map the direct translations with the universal resource identifier that connected the corresponding English vocabulary. This will ensure that no redundancy exists. For the portion that will not come to much effect for direct translation, the new concepts added to the *existing ontology* will cover them up. The sequence is shown in Figure 4.7.

To map the vocabularies in the dictionary, which will not provide any specific result for direct translation or may not be suitable for linking with the new concepts, we will require some measure to find out the closest match. For such cases, the mapping proposed in [7] can be used, which will provide a *confidence degree* [7]. The best match, judged by the *confidence degree* can be used to link it up with the universal resource identifier.

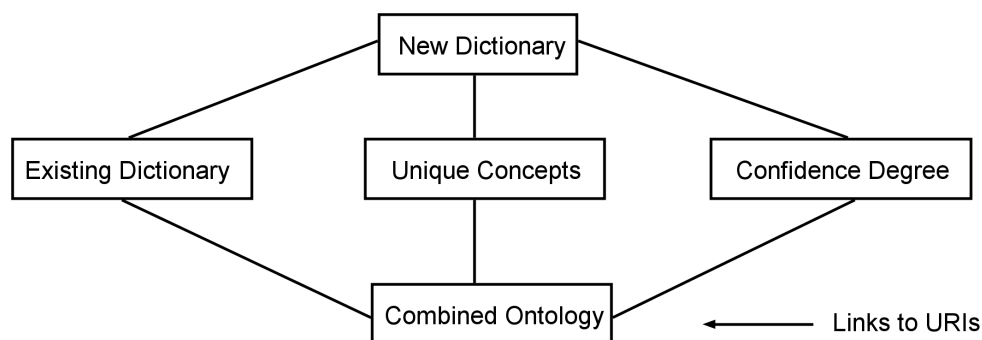


Figure 4.7: Linking dictionary to ontology

4.3.3 Information Retrieval

We can retrieve information from the mapped ontology by constructing SPARQL queries directly to the SPARQL ENDPOINT¹ or by making textual queries which will be converted into SPARQL queries later. Just as the multilingual ontology comprises of a set of mapped ontologies from various languages, retrieval of information also requires to be language independent. The mapping technique, proposed in the previous subsection, for augmenting multilingual ontology, can also be adapted in this case.

To make the retrieval system language independent, the textual queries require to be translated into an existing language from which the conversion to SPARQL query will be more straightforward. Considering a similar example as used in the previous section, the English language can be used as the existing one, and the Bengali language will be used as the local one to demonstrate the implementation. Whenever a textual query is made in Bengali, a tag needs to be attached to state the language. Consider the tag for Bengali to be <BN>. Such tags will help the system to identify the language.

At first, the Bengali textual query will be translated into English textual query using a regular language translator. Then the English textual query will be tokenized into entities. Open Calais², Twine³, Zemanta⁴ etc. are some of the products, which are built on Named-Entity Recognition (NER). These provide APIs which can be used to extract entities. These entities will serve as the unit parts of the query. Then the SPARQL query will be formed using these entities.

An example can be illustrated for better understanding of the concept. Let, a query is made in Bengali - “১৯৩০ সালের পরে জন্ম এমন বাংলাদেশী অভিনেতাদের নাম? [English translation: List of Bangladeshi actors born after 1930]”. The expected result will be the list of URIs of the Bangladeshi actors who were born after 1930. To process this query, we first translate it into English. The translated query will be “After 1930, the name of the actors who were born in

¹<http://dbpedia.org/sparql>

²<http://www.opencalais.com/>

³[https://en.wikipedia.org/wiki/Twine_\(website\)](https://en.wikipedia.org/wiki/Twine_(website))

⁴<http://www.zemanta.com/>

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX db-ont: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?person ?name ?birth
WHERE {
  ?person db-ont:birthPlace <http://dbpedia.org/resource/Bangladesh> .
  ?person db-ont:occupation <http://dbpedia.org/resource/Actor> .
  ?person foaf:name ?name .
  ?person db-ont:birthDate ?birth
  FILTER (?birth > "1930-01-01"^^xsd:date) .
}

```

Listing 4.1: Generated SPARQL query.

Bangladesh?”. The extracted entities are *birthPlace: Bangladesh, occupation: Actor, Name, birthDate > 1930* etc. So, a SPARQL query will be generated from these entities. Listing 4.1 shows the generated SPARQL query.

Since the English and Bengali vocabularies are already integrated to the multilingual ontology, the universal identifier returned by the translated query will thus, be the same for the object the Bengali query was initially formed. If we want only the results written in Bengali language, then, we need to add a language filter in the SPARQL query.

These SPARQL queries can be applied to fetch knowledge from DBpedia and other knowledge sources. The standard format for consideration about this knowledge base is the triple format of subject-property-object. If the concept of multilingual ontology can be applied for data set augmentation, we can directly fetch knowledge form knowledge bases using any language.

4.4 Summary

- An algorithm is designed and implemented to construct the content tree preserving semantic relations.
- A set of semantic rules is applied to optimize the content tree to reduce the traversal time.
- The core concepts of the document is used to gather more knowledge from the knowledge graphs. Also, a semantic query language for databases is used to fetch all information from existing ontology. At this point, the integration of a multilingual ontology results in better information accumulation.
- The extracted knowledge is clustered to get the hierarchical taxonomy of the document.
- The multilingual ontology framework is designed to be used as a single ontology. It uses multiple knowledge sources and dictionaries of various languages and map them to enhance it's knowledge [71].

Chapter 5

Experimental Result and Analysis

In this chapter, we will discuss the experimental analysis process and evaluate the performance of our proposed model through result analysis using our data set. Initially, in Section 5.1, we will discuss the experimental set up of our implementation followed by Section 5.2 explaining the language selection of our experiment. Then, we'll give an overview of the data-set in Section 5.3 that we used to evaluate the proposed method. Finally, in Section 5.5, we'll discuss and analyze the result of our proposed system elaborately.

5.1 Experimental Setup

Our thesis requires to communicate between multiple systems and platforms over the web. Hence, we have used JAVA¹ as the development language. After pre-processing the input document, the extracted tokens are tagged using synthetic POS tagger along with all information. We used the SQLite database to keep the tagged tokens. To establish the database connection with JAVA, an application programming interface (API) named JAVA Database Connectivity (JDBC) is used which defines how a client may access a database. It provides methods to query and update data in a database. To extract RDF triples from the DBpedia, we have used Apache Jena framework of JAVA, which is free and open source. The framework provides different APIs to interact together to process RDF data for building semantic web and Linked Data applications. The SPARQL query language also helped us to query data from DBpedia. We have also interacted with Google Knowledge Graph for augmenting knowledge graphs. Hence, we have used Google Knowledge Graph Search API to find entities in the Google Knowledge Graph. The API uses standard schema.org² types and is compliant with the JSON-LD specification. The translator API provided by Google Translate³ has been used as the translator for the proposed system.

¹<https://docs.oracle.com/javase/8/docs/>

²<https://schema.org/>

³<https://cloud.google.com/translate/docs/>

5.2 Selection of Language

The Java interface can take input of the unstructured documents of any language. After receiving the information, we'll tokenize and prepare the tagged document following the process described in the Section 4.1.1. The preparation of the tagged document is language dependent as we need to use specific parts of speech tagger of that particular language. The rest of the process is the same for all languages. To extract additional knowledge from the existing knowledge bases, we need to translate the tokens into English. Google Knowledge Graph contains a considerable number of entities in English than any other languages [72]. Besides, Table 4.2 shows the number of instances for a set of selected classes within the canonicalized DBpedia data sets for English and Bengali. The number of instances in English is higher than any other languages in DBpedia also. Hence, we have translated the tokens into English to serve the purpose. All further steps have used this translated version. We have chosen the Bengali language as the other language than English during implementation because being the sixth most spoken native language in the world by population, Bengali deserves much more attention in NLP while in reality there are very few NLP tools that support Bengali.

5.3 Experimental Data Set

We have used a modified version of Wikipedia pages as input. We have eliminated complex sentences from the input documents. We have conducted our experiment on Bengali documents. However, the proposed approach will work similarly for English documents also. The whole process has been divided into a few steps. Among all the steps, the language dependency involves in the POS tagging stage only. Once we have tagged the tokens, the proposed method will work for all languages.

5.4 Experiment Conduction

At first, the system takes input from the corpus for line segmentation and tokenization following the process stated in Section 4.1.1. From Corpus 1, we have got 133 tokens. The sentence position of these tokens in the provided document, token position in the sentences, and overall token position in the corpus are noted to determine the coreference resolution discussed in Section 4.1.2. After this step, all pronouns are replaced by the referred entity. The parts of speech tagging and labeling tokens with some additional information are performed on the extracted tokens with the help of an external dictionary. We have trained our system with 350 stop words, 12 month names, 64 districts of Bangladesh, 250 country names, 10000 numbers, etc. for the additional knowledge related to various nouns, pronouns and stop words. For parts of speech tagging, we have used a synthetic POS tagger. Then, the less significant tokens are filtered,

which results in 98 tokens in total. With the help of these steps, the processed tagged document is generated from the unstructured input text. The processed tagged document for the initial two lines of Figure 4.1 is demonstrated in Table 4.1.

The nodes and edges are chosen from these tagged tokens to construct the content tree. We have also applied some grammatical rules for merging some nodes. The semantic rules are mentioned in Section 4.1.4. The generated tree of Corpus 1 has 71 nodes and a single root. The highest referred entity is “মাইকেল জোসেফ জ্যাকসন” (English translation: Michael Joseph Jackson). Therefore, this entity has been selected as the root of the tree.

To generate the concept hierarchy for a corpus, we have chosen the root of the content tree as the core concept of that document. Here, we have extracted “মাইকেল জোসেফ জ্যাকসন” (English translation: Michael Joseph Jackson) as the core concept of Corpus 1. Then, we have gathered additional information related to “মাইকেল জোসেফ জ্যাকসন” (English translation: Michael Joseph Jackson) from Google knowledge graph and DBpedia using APIs. From all the collected information, we have filtered information which is useful for generating the corpus category. Then, we embedded the filtered information in the vector space using Word2vec algorithm to get the cosine similarity among the data extracted from the knowledge bases. This measure is used to cluster the data as well as to select cluster tags to generate the levels of the concept hierarchy. The DBpedia ontology has been used to get the upper level of hierarchy from the previous level, which is more generalized than the previous level. Thus, the knowledge graph augmented document concept hierarchy is generated.

5.5 Result Analysis

In this section, we will evaluate the proposed method through result analysis based on the obtained result. Firstly, we want to assess the accuracy of the content tree and the extracted concepts. Then, we want to focus on the generated concept hierarchy.

5.5.1 Analysis on Document Content Tree

Document content tree generation is the prior step for concept extraction. The very first step, tokenization works precisely after getting the input document. The segmentation of lines and tokens are 100 percent accurate. However, during tokenization, we could not remove the inflection properly from all tokens. For example: “মাইকেল জোসেফ জ্যাকসন এর জন্ম ২৯ আগস্ট ১৯৫৮ সালে।” (English translation: Michael Joseph Jackson born on August 29, 1958.) Here, the segregated tokens are ['মাইকেল', 'জোসেফ', 'জ্যাকসন', 'জন্ম', '২৯', 'আগস্ট', '১৯৫৮', 'সালে']. In the last token “সালে” the “ে” is not removed. Our system works well for any document that contains simple sentences. We can extract the core concepts successfully for these documents. Though complex and compound sentences are not in our thesis scope, the accuracy declines

Table 5.1: Statistics of the corpus to evaluate the model and extracted tree nodes.

Corpus	Number of Simple Sentence	Number of Perfectly Extracted Nodes	Number of Unsuccessful Extracted Nodes	Number of Generated Content Tree
Corpus 1	20	71	1	1
Corpus 2	15	58	3	1
Corpus 3	12	56	2	1
Corpus 4	15	57	0	1
Total	62	242	6	4

while working on such sentences. We have faced some issues which we are unable to address at present. Firstly, if a sentence contains two or more verbs, we could not identify which verb belongs to which clause. For instance, the extracted tokens of the sentence “মাইকেল জোসেফ জ্যাকসন এর জন্ম ২৯ আগস্ট, ১৯৫৮ এবং মৃত্যু ২৫ জুন, ২০০৯ সালে।” (English translation: Michael Joseph Jackson was born on August 29, 1958, and died on June 25, 2009.) will be [‘মাইকেল’, ‘জোসেফ’, ‘জ্যাকসন’, ‘জন্ম’, ‘২৯’, ‘আগস্ট’, ‘১৯৫৮’, ‘মৃত্যু’, ‘২৫’, ‘জুন’, ‘সালে’]. From these tokens, we’ll separate the verbs ‘জন্ম’ and ‘মৃত্যু’ together for the edge label of the whole branch constructed from this sentence. Secondly, if a complex or compound sentence contains more than one noun of the same category, our co-reference module unable to detect the subject of the next sentence containing a pronoun, consider a sentence “কবির রহিমের সাথে স্কুলে যায়। তার গ্রামের বাড়ি রসিদপুর।” (English translation: Kabir goes school with Rahim. His village home is Rashidpur). Here, our proposed system is unable to detect the precedent of the pronoun “তার” correctly. We do not have any good co-reference resolution module in the Bengali language like the one [73] exists for English. Besides, our proposed system lacks to differentiate between another noun, which is the same as the sentence subject name. For example: “মাইকেল জোসেফ জ্যাকসন প্রতি সন্ধ্যায় জোসেফ ক্যাফেতে গান করেন।” (English translation: Michael Joseph Jackson sings in Joseph cafe every evening). During the co-reference resolution, any part of the subject name will be replaced by the full name of the subject which is required to connect all branches with the root. Hence, the proposed method will replace all “জোসেফ” with “মাইকেল জোসেফ জ্যাকসন”. A strong co-reference resolution module for the Bengali language will help us to overcome this issue. Table 5.1 shows the statistics of the corpus and extracted tree nodes. We could extract almost all nodes, which are significant for the document content tree except a very few mentioned previously.

5.5.2 Analysis on Concept Hierarchy

During the generation of the content tree, our proposed method figures out the most significant concept of the document. Usually, the majority of the sentences of that document are related

to that concept. To serve this purpose, we have chosen the root of the content tree generated from that document. Besides the root, frequent concepts sometimes play a significant role in the categorization of a particular document. At present, we are not considering those. However, such consideration will have a positive impact on the hierarchical structure. As we have tested our method on Wikipedia documents, we have been able to extract the core concepts from the input documents successfully. Later, we have augmented knowledge bases to obtain more knowledge related to the core concepts of the document. We have faced difficulties while extracting data using the core concepts of Bengali documents as there is very less entry in the knowledge bases in Bengali. We had to put an extra effort to translate the concepts in English using Google translator to get data from the knowledge bases. The knowledge bases have the most substantial amount of data in English. However, we could barely get any data from Google knowledge graph while working on a document related to the Bangladeshi player “Sakib al-Hasan”. A similar thing happened for many other documents also. After extracting all data from the knowledge bases, the rest work was done smoothly with the help of clustering algorithms.

5.6 Comparative Analysis

5.6.1 Knowledge Base Augmentation

Knowledge elicitation is a major way to gather information. In our thesis, we have augmented existing knowledge bases to enhance the knowledge related to the document. Existing ontologies play significant roles to serve this purpose. In the era of Semantic web, knowledge integration from the available sources is one of the primary goal of us. We have analysed our proposed method with the knowledge base augmentation and also, without it. When we connect the knowledge sources, we get the privilege to improve the quality of concept hierarchy as we have more information in hand than the solo document content. The class hierarchy of the existing ontologies help to create the hierarchical taxonomy for the documents. Besides, the additional information can be used to enrich the document content also. Without augmenting the knowledge bases, it is not possible to create the concept hierarchy which helps to classify documents. We'll only be able to infer the document subject.

5.6.2 Document Content Tree Generation

In this thesis, we have generated rooted tree based structure for the documents. If we start traversing from the root, we'll be able to find information extracted from the document, once we reach to the appropriate node. If we generated a graph instead of tree, it will be difficult to portray the document content and the subject. Also, further work can be done on the augmentation among sub-trees holding related information. For a graph, this integration will be complex. In

addition, it is not possible to generate the hierarchical structure from a graph.

5.6.3 Semantic Rules

Semantics provides relation among the bag of words to give the proper meaning. In our thesis, we have prepared some semantic rules which improve the quality of the nodes. These semantic rules are generated with the help of Bengali grammar as we have chosen to experiment on Bengali documents. Without these semantic rules, the generated rooted tree will contain more nodes, the tree height will increase and the quality of some nodes will be less.

Chapter 6

Conclusion

This chapter will conclude the thesis with a focus on the attainments in Section 6.1. Then, Section 6.2 will highlight the scope of future works.

6.1 Contribution of the Work

In this thesis, we have introduced a framework for generating knowledge graph augmented concept hierarchy. Also, we have also focused on the framework for a multilingual ontology. A brief description of the notable features of our proposed methods is as follows:

1. The framework portrays document content in a rooted tree structure that preserves semantic relations and extracts the core topics of the document. Therefore, we are considering the root as one of the main concepts of a document as the content has a significant amount of discussion regarding the root. We have also considered grammatical rules and relations between the words.
2. The system augments the knowledge graphs to enhance knowledge. The existing knowledge bases possess information related to the main concepts of the documents. Integration of additional information helps to create hierarchical taxonomy of the document.
3. The framework enlarges the domain of multilingual ontology, which integrates multiple languages to enrich its knowledge. Thus, the system will map the documents of any language into the existing knowledge bases without any hassle.
4. The multilingual ontology is capable of being mapped with dictionaries or data sources of an infinite number of languages.

6.2 Scope of Future Work

From the challenges and the explanation of result analysis, we can highlight some of the scope of the future works to step forward from this thesis. The future works are listed as follows:

1. A strong co-reference resolution module for the Bengali language can be developed to refer more than one expressions in a sentence to the same referent. The selection of the root of the content tree highly depends on the concepts which are referred significantly in the whole document. At present, there is no co-reference module available for the Bengali language. Therefore, to lessen the difficulty of referencing as well as to increase the accuracy of the root selection, further research can be conducted in this area.
2. While working with the unstructured text, this thesis focused on single subject oriented documents. To deal with higher challenges, source of unstructured text can be made general. Therefore, concept hierarchy can be generated from all kind of documents as well as web pages.
3. There is plenty of room for improvement in the generation of the content tree from the documents. The more grammatical rules are considered, the better the content tree will be. Sentence structures like complex, compound, and compound-complex can be taken under consideration. In addition, passive sentences, exclamatory sentences, document context, tense, idioms, and phrases also possess an excellent impact on the content tree.

Document hierarchy based on the content tree will contribute predominantly to the taxonomy of documents. The tree representation will also help us to merge any document with the same concepts to increase connectivity of knowledge efficiently. In addition, this will open up a vast field of the research area to represent the documents more structurally, making it more searching efficient and ultimately achieving the vision of Semantic web.

References

- [1] M. Baziz, M. Boughanem, and N. Aussenac-Gilles, “Conceptual indexing based on document content representation,” in *Context: nature, impact, and role*, pp. 171–186, Springer, 2005.
- [2] S. Hassan, R. Mihalcea, and C. Banea, “Random walk term weighting for improved text classification,” *International Journal of Semantic Computing*, vol. 1, no. 04, pp. 421–439, 2007.
- [3] I. Horrocks, B. Parsia, P. Patel-Schneider, and J. Hendler, “Semantic web architecture: Stack or two towers?,” in *International Workshop on Principles and Practice of Semantic Web Reasoning*, pp. 37–41, Springer, 2005.
- [4] E. RDF4J, “How to use rdf and the rdf4j framework.” <http://docs.rdf4j.org/rdf-tutorial/>. (Accessed on 05/17/2019).
- [5] A.-H. Tan, “Text mining: The state of the art and the challenges,” in *Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, vol. 8, pp. 65–70, sn, 1999.
- [6] W. contributors, “Wikipedia - wikiwand.” <http://www.wikiwand.com/en/Wikipedia>. (Accessed on 03/21/2019).
- [7] C. T. dos Santos, P. Quaresma, and R. Vieira, “A framework for multilingual ontology mapping,” in *Proceedings of the International Conference on Language Resources and Evaluation, LREC*, pp. 1034–1037, ACM, 2008.
- [8] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.
- [9] P. Hitzler, M. Krotzsch, and S. Rudolph, *Foundations of semantic web technologies*. Chapman and Hall/CRC, 2009.
- [10] L. Yu, *A developer’s guide to the semantic Web*. Springer Science & Business Media, 2011.

- [11] J. Guyot, S. Radhouani, and G. Falquet, "Ontology-based multilingual information retrieval," in *CLEF Workshop, Working Notes Multilingual Track*, pp. 21–23, 2005.
- [12] A. H. Lashkari, F. Mahdavi, and V. Ghomi, "A boolean model in information retrieval for search engines," in *Information Management and Engineering, 2009. ICIME'09. International Conference on*, pp. 385–389, IEEE, 2009.
- [13] W. Waller and D. H. Kraft, "A mathematical model of a weighted boolean retrieval system," *Information Processing & Management*, vol. 15, no. 5, pp. 235–245, 1979.
- [14] G. Salton, E. A. Fox, and H. Wu, "Extended boolean information retrieval," *Communications of the ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.
- [15] J. H. Lee, "Properties of extended boolean models in information retrieval," in *SIGIR'94*, pp. 182–190, Springer, 1994.
- [16] G. Bordogna, P. Carrara, and G. Pasi, "Fuzzy approaches to extend boolean information retrieval," in *Fuzziness in database management systems*, pp. 231–274, Springer, 1995.
- [17] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1–4, pp. 43–52, 2010.
- [18] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 794–804, 2018.
- [19] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," *CoRR*, vol. abs/1507.07998, 2015.
- [20] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser, "Latent semantic analysis," in *Proceedings of the 16th international joint conference on Artificial intelligence*, pp. 1–14, Citeseer, 2004.
- [21] E. A. Fox, *Extending the boolean and vector space models of information retrieval with p-norm queries and multiple concept types*. PhD thesis, Cornell University, Ithaca, NY, USA, 1983.
- [22] H. Balinsky, A. Balinsky, and S. Simske, "Document sentences as a small world," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2583–2588, IEEE, 2011.
- [23] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth," *Cognitive science*, vol. 29, no. 1, pp. 41–78, 2005.

- [24] S. Hensman, “Construction of conceptual graph representation of texts,” in *Proceedings of the Student Research Workshop at HLT-NAACL 2004*, pp. 49–54, Association for Computational Linguistics, 2004.
- [25] F. Zhou, F. Zhang, and B. Yang, “Graph-based text representation model and its realization,” in *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*, pp. 1–8, IEEE, 2010.
- [26] J. Wu, Z. Xuan, and D. Pan, “Enhancing text representation for classification tasks with semantic graph structures,” *International Journal of Innovative Computing, Information and Control (ICIC)*, vol. 7, no. 5, 2011.
- [27] W. Wang, D. B. Do, and X. Lin, “Term graph model for text classification,” in *International Conference on Advanced Data Mining and Applications*, pp. 19–30, Springer, 2005.
- [28] K. Valle and P. Ozturk, “Graph-based representations for text classification,” in *India-Norway Workshop on Web Concepts and Technologies, Trondheim, Norway*, pp. 2363–2366, 2011.
- [29] S. S. Sonawane and P. A. Kulkarni, “Graph based representation and analysis of text document: A survey of techniques,” *International Journal of Computer Applications*, vol. 96, no. 19, 2014.
- [30] M. Sanderson and B. Croft, “Deriving concept hierarchies from text,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 206–213, ACM, 1999.
- [31] P. Cimiano, A. Hotho, and S. Staab, “Clustering concept hierarchies from text,” in *Proceedings of the Conference on Lexical Resources and Evaluation (LREC)*, European Language Resources Association (ELRA), 2004.
- [32] M. S. Hossain and R. A. Angryk, “Gdclust: A graph-based document clustering technique,” in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pp. 417–422, IEEE, 2007.
- [33] P. Lakkaraju, S. Gauch, and M. Speretta, “Document similarity based on concept tree distance,” in *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pp. 127–132, ACM, 2008.
- [34] “Semantic web.” https://www.semanticweb.org/wiki/Main_Page.html. (Accessed on 03/16/2019).
- [35] “Semantic web-w3c.” <https://www.w3.org/standards/semanticweb/>. (Accessed on 03/16/2019).

- [36] T. Berners-Lee and M. Fischetti, “Weaving the web. harpersanfrancisco. chapter 12,” tech. rep., ISBN 978-0-06-251587-2, 1999.
- [37] R. V. Guha, “Light at the end of the tunnel,” in *Talk at the 12th International Semantic Web Conference (ISWC), Sydney*, vol. 10, 2013.
- [38] D. L. McGuinness, F. Van Harmelen, *et al.*, “Owl web ontology language overview,” *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.
- [39] “Semantic web - wikipedia.” https://en.wikipedia.org/wiki/Semantic_Web#cite_note-W3C-SWA-2. (Accessed on 03/17/2019).
- [40] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible markup language (xml) 1.0,” 2000.
- [41] M. A. Hearst, “Text data mining: Issues, techniques, and the relationship to information access,” in *Presentation notes for UW/MS workshop on data mining*, vol. 1, p. 997, 1997.
- [42] E. Simoudis, “Reality check for data mining,” *IEEE Intelligent Systems*, no. 5, pp. 26–33, 1996.
- [43] D. Zelenko, C. Aone, and J. Tibbetts, “Coreference resolution for information extraction,” in *Proceedings of the Conference on Reference Resolution and Its Applications*, 2004.
- [44] W. M. Soon, H. T. Ng, and D. C. Y. Lim, “A machine learning approach to coreference resolution of noun phrases,” *Computational linguistics*, vol. 27, no. 4, pp. 521–544, 2001.
- [45] “Wikipedia.org traffic, demographics and competitors - alexa.” <https://www.alexa.com/siteinfo/wikipedia.org>. (Accessed on 03/21/2019).
- [46] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [47] “Wordnet | a lexical database for english.” <https://wordnet.princeton.edu/>. (Accessed on 03/21/2019).
- [48] “Knowledge graph - wikipedia.” https://en.wikipedia.org/wiki/Knowledge_Graph. (Accessed on 05/17/2019).
- [49] “Google knowledge graph api.” <https://developers.google.com/knowledge-graph/>. (Accessed on 05/17/2019).
- [50] “Dbpedia-wikipedia.” <https://en.wikipedia.org/wiki/DBpedia>. (Accessed on 03/21/2019).

- [51] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “Dbpedia: A nucleus for a web of open data,” in *The semantic web*, pp. 722–735, Springer, 2007.
- [52] “Explore dbpedia.” <https://wiki.dbpedia.org>. (Accessed on 03/21/2019).
- [53] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: A large ontology from wikipedia and wordnet,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 3, pp. 203–217, 2008.
- [54] “Yago: A high-quality knowledge base.” <https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>. (Accessed on 03/21/2019).
- [55] A. Cayley, “Xxviii. on the theory of the analytical forms called trees,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 13, no. 85, pp. 172–176, 1857.
- [56] “Tree (graph theory)-wikipedia.” [https://en.wikipedia.org/wiki/Tree_\(graph_theory\)#cite_note-10](https://en.wikipedia.org/wiki/Tree_(graph_theory)#cite_note-10). (Accessed on 03/20/2019).
- [57] T. Beyer and S. M. Hedetniemi, “Constant time generation of rooted trees,” *SIAM Journal on Computing*, vol. 9, no. 4, pp. 706–712, 1980.
- [58] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, “An introduction to the syntax and content of cyc.,” in *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pp. 44–49, 2006.
- [59] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, AcM, 2008.
- [60] D. Vrandečić and M. Krötzsch, “Wikidata: a free collaborative knowledgebase,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [61] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, “Toward an architecture for never-ending language learning,” in *AAAI*, vol. 5, pp. 1306–1313, Atlanta, 2010.
- [62] “Vector representations of words.” <https://www.tensorflow.org/tutorials/representation/word2vec>, 2018. (Accessed on 03/20/2019).

- [63] Y. Goldberg and O. Levy, “word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method,” *arXiv preprint arXiv:1402.3722*, 2014.
- [64] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- [65] A. Huang, “Similarity measures for text document clustering,” in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pp. 49–56, 2008.
- [66] “Wikipedia.” <https://www.wikipedia.org/>. (Accessed on 03/16/2019).
- [67] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, *et al.*, “Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia,” *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
- [68] “Data set statistics.” <http://wiki.dbpedia.org/services-resources/datasets/dataset-2015-04/dataset-2015-04-statistics>. (Accessed 03/06/2017).
- [69] A. Tawfik, F. Giunchiglia, and V. Maltese, “A collaborative platform for multilingual ontology development,” *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, vol. 8, no. 12, pp. 3795–3804, 2014.
- [70] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [71] M. T. M. Ankon, S. N. Tumpa, and M. M. Ali, “A multilingual ontology based framework for wikipedia entry augmentation,” in *Proceeding of the 19th International Conference on Computer and Information Technology*, pp. 541–545, 2016.
- [72] J. Hernández-González, E. R. Hruschka Jr, and T. M. Mitchell, “Merging knowledge bases in different languages,” in *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pp. 21–29, 2017.
- [73] “The stanford natural language processing group.” <https://nlp.stanford.edu/projects/coref.shtml>. (Accessed on 04/10/2019).

Appendix A

Algorithms

A.1 Algorithm of Document Content Tree Generation

In Algorithm 1, we show how to generate the document content tree.

Algorithm 1 Algorithm for representing the document as tree.

```

Initialize  $N \rightarrow \{n \text{ belongs to } N : n \text{ is a node in } G\}$ 
and  $E \rightarrow \{e \text{ belongs to } E : e \text{ is an edge in } G\}$ 
 $i = 0$ 
 $currentNode = 0$ 
 $token = null$ 
while ! $EOF$  do
  if  $token.POS \neq null$  then
    if  $token.POS = Adverb$  then
       $continue$ 
    else if  $token.POS = verb$  then
       $edge = token$ 
    else if  $token.POS = Noun \vee token.POS = Adjective$  then
       $parent = token$ 
    end if
  end if
   $token = file.getRow(i)$ 
  if  $token.POS = Verb$  then
     $continue$ 
  end if
  if  $currentNode < file.lineNo$  then
     $currentNode = file.lineNo$ 
     $parent = null$ 
  end if
   $flag = true$ 
  for each  $n$  belongs to  $N$  do
    if  $token.word = n$  then
       $token = n$ 
       $flag = false$ 
       $break$ 
    end if
  end for
  if  $flag = true$  then
     $N \leftarrow token.word$ 
  end if
  if  $parent \neq null$  then
    Find  $n$  belongs to  $N : n = parent.word$ 
     $Edge(n, edge, token)$ 
  end if
   $i ++$ 
end while

```

Generated using Postgraduate Thesis L^AT_EX Template, Version 0.97. Department of
Computer Science and Engineering, Bangladesh University of Engineering and
Technology, Dhaka, Bangladesh.

This thesis was generated on August 21, 2019 at 5:59pm.